# ABSTRACT

Title of dissertation: MULTICASTING IN ALL-OPTICAL WDM NETWORKS

Anuj Rawat, Ph.D., 2008

Dissertation directed by: Professor Mark Shayman
Department of Electrical and Computer Engineering

In this dissertation, we study the problem of (i) routing and wavelength assignment, and (ii) traffic grooming for multicast traffic in Wavelength Division Multiplexing (WDM) based all-optical networks.

We focus on the *static* case where the set of multicast traffic requests is assumed to be known in advance. For the routing and wavelength assignment problem, we study the objective of minimizing the number of wavelengths required; and for the traffic grooming problem, we study the objectives of minimizing (i) the number of wavelengths required, and (ii) the number of electronic components required.

Both the problems are known to be hard for general fiber network topologies. Hence, it makes sense to study the problems under some restrictions on the network topology. We study the routing and wavelength assignment problem for bidirected trees, and the traffic grooming problem for unidirectional rings. The selected topologies are simple in the sense that the routing for any multicast traffic request is trivially determined, yet complex in the sense that the overall problems still remain hard. A motivation for selecting these topologies is that they are of practical interest since most of the deployed optical networks can be decomposed into these elemental topologies.

In the first part of the thesis, we study the the problem of multicast routing and wavelength assignment in all-optical bidirected trees with the objective of minimizing the number of wavelengths required in the network. We give a $\frac{5}{2}$-approximation algorithm for the case when the degree of the bidirected tree is at most 3. We give another algorithm with approximation ratio $\frac{10}{3}$, 3 and 2 for the case when the degree of the bidirected tree is equal to 4, 3 and 2, respectively. The time complexity analysis for both these algorithms is also presented. Next we prove that the problem is hard even for the two restricted cases when the bidirected tree has (i) depth 2, and (ii) degree 2. Finally, we present another hardness result for a related problem of finding the clique number for a class for intersection graphs.

In the second part of the thesis, we study the problem of multicast traffic grooming in all-optical unidirectional rings. For the case when the objective is to minimize the number of wavelengths required in the network, given an $\alpha$-approximation algorithm for the circular arc coloring problem, we give an algorithm having asymptotic approximation ratio $\alpha$ for the multicast traffic grooming problem. We develop an easy to calculate lower bound on the minimum number of electronic components required to support a given set of multicast traffic requests on a given unidirectional ring network. We use this lower bound to analyze the worst case performance of a pair of simple grooming schemes. We also study the case when no grooming is carried out in order to get an estimate on the maximum number of electronic components that can be saved by applying intelligent grooming. Finally, we present a new grooming scheme and compare its average performance against other grooming schemes via simulations. The time complexity analysis for all the grooming schemes is also presented.

# MULTICASTING IN ALL-OPTICAL WDM NETWORKS

by

Anuj Rawat

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

Advisory Committee:
Professor Mark Shayman, Chair/Advisor
Professor Richard La
Professor Steven Marcus
Professor Ankur Srivastava
Professor Samir Khuller, Dean's Representative

# Dedication

To my parents.

# Acknowledgments

First and foremost, I would like to thank my advisor Prof. Mark Shayman for his guidance and support through the course of my Ph.D. Without his constant encouragement and advice, this thesis would not have been possible. He not only provided ideas and directions for the various research problems, but also steered my graduate studies toward a fruitful conclusion.

I would like to thank Prof. Steve Marcus, Prof. Richard La, Prof. Ankur Srivastava and Prof. Samir Khuller for serving on the dissertation committee. I would like to thank Prof. Richard La and Prof. Steve Marcus for their patience in listening to and deciphering my jumbled thoughts and ideas during the research meetings, and providing invaluable comments. I would like to thank Prof. Ankur Srivastava for the useful discussion pertaining to the work in Chapter 7. I would like to thank Prof. Samir Khuller for the helpful discussions and advice regarding both my research as well as other professional endeavors.

I would also like to thank my colleagues Abhishek Kashyap, Mehdi Kalantari and Fangting Sun for their help and suggestions on various research problems that I worked on during my Ph.D.

I would also like to acknowledge all my friends, specifically Abhinav Gupta, Pavan Turaga, Srikanth Vishnubhotla, Ravi Tandon, Rahul Ratan, Manish Shukla, Archana Anibha, Swati Jarial, Vishal Khandelwal, Tarun Pruthi, Gaurav Agarwal, Amit Agrawal, Ashwin Swaminathan, Himanshu Tyagi for the various fruitful discussions about my research as well as for making my stay at UMD a pleasant and memorable experience.

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

The defining characteristic of optical networks is that the data transmission is carried out in optical domain, over directed fiber-optic links. Although the idea of employing fiber optic cables for transmitting data is not new in itself, the development of Wavelength Division Multiplexing (WDM) [1, p.208-210] technology proved decisive in the wide scale deployment of optical networks that we see today.

WDM is a technique which allows simultaneous transmission of multiple data streams over a single optical fiber by using a different wavelength of light for each individual signal. Hence, in a WDM based optical network, an optical fiber can be treated as a set of parallel optical channels, each operating at a different wavelength. The extremely high data transfer rate achievable by employing WDM, along with the low bit error rate and delay characteristics of the optical fiber has made WDM based optical networks the obvious contender for the next generation high speed data transport networks.

## 1.1 Optical Networks: Concepts

In this section, we present some basic concepts and definitions pertaining to optical networking. First we introduce the concept of all-optical networking, followed by the problems of routing and wavelength assignment, and traffic grooming in all-optical networks. Finally, we discuss the concept of multicasting and how to support multicast traffic in all-optical networks. For a more detailed review, the reader is referred to the excellent tutorial by Rouskas et al. [2].

### 1.1.1 All-Optical Networking

In WDM based optical networks, each node is equipped with an Optical Cross Connect (OXC) to switch the optical signals on the incoming fibers onto the outgoing fibers. OXC first demultiplexes the light on the incoming fibers into its constituent wavelengths, then carries out the required switching operation in either electronic or optical domain before multiplexing the switched wavelengths onto the outgoing fibers. In case of *electronic* (or *opaque*) switching, the optical signals are converted to electronic domain for switching and then back to optical domain. Thus, it involves Optical-Electronic-Optical (OEO) operations. The drawback of this approach is that with the current technology, it is difficult to perform electronic processing at the high data transfer rates supported by the optical fibers. On the other hand, in *optical* (or *transparent*) switching, the wavelength signals obtained after demultiplexing the incoming light, are switched using optical switch modules. Thus, there is no OEO operation and the switching is carried out entirely in the optical domain. The drawback of this approach is that the control over the switching is not as fine as with electronic switching. This is because in optical switching, the data traffic on each individual wavelength is preserved whereas in electronic switching even the sub-wavelength traffic can be switched independently.

The important special case when all the switching in the network is carried out in the optical domain is called *all-optical* networking. With the current technology trends, it seems likely that the mismatch between the data rates supported by optical and electronic components in the network shall continue to grow for some time. Therefore, in this work we restrict ourselves to the case of all-optical networks.

### 1.1.2 Routing and Wavelength Assignment

The optical switching capability of the network nodes allow us to setup *lightpaths* between any given pair of nodes. A lightpath between two network nodes is a *clear* optical channel from the start node to the end node, possibly spanning several fiber links. Here by clear we mean that there is no OEO operation on any of the intermediate nodes, i.e., the signal remains in the optical domain throughout the length of the path and the conversion of data between the optical and electronic domains takes place at the start and the end nodes only. Since optical switching is carried out at the granularity of wavelength channels, lightpaths are also setup at the same granularity. Setting up a lightpath requires determining its routing over the fiber links and assigning it a wavelength on each of the links in its path. The resulting problem is referred to as the *routing and wavelength assignment* problem. Clearly, wavelengths assigned to different lightpaths on a common fiber link must be distinct. Also, an implication of employing optical switching at the intermediate nodes is that the wavelength assigned to a lightpath must be the same on all its links. This is known as the *wavelength-continuity constraint*. This constraint can be relaxed if the network nodes are equipped with special optical devices called wavelength converters. But due to their prohibitive cost, in this work we assume that there are no such devices in the network.

The routing and wavelength assignment problem comes in two flavors: *static* and *dynamic*. In the static problem, it is assumed that the set of traffic requests, for which the routing and wavelength assignment problem needs to be solved, is known in advance. On the other hand the dynamic routing and wavelength assignment refers to the case in which traffic requests arrive in real time. In this work, we only study the static routing and wavelength assignment problem.

The most widely studied objective function for the problem of static routing and wavelength assignment in all-optical networks is to minimize the number of wavelengths required per fiber in order to support a given set of traffic requests. The justification for employing this as the objective function is that due to technological constraints, the number of wavelengths that can be placed on a single optical fiber is a limiting resource. Although, over the years other cost functions have been suggested and studied for the routing and wavelength assignment problem, in this work we try to find routing and wavelength assignments that minimize the number of wavelengths required.

The interested reader is referred to [3] for a review of the routing and wavelength assignment problem.

### 1.1.3 Traffic Grooming

As described before, in an all-optical network a traffic request must be routed on a single lightpath. One obvious strategy would be to setup a new lightpath for each individual traffic request. A problem with this approach is that since the maximum number of wavelengths of light that can be multiplexed over an optical fiber is limited by the current WDM technology, it might not be possible to set up a new lightpath for every traffic request. Hence, setting up dedicated lightpaths for each traffic request might not be a very attractive approach. Recent advances in the fiber-optic technology has pushed the transport capacities of individual wavelength channels on the optical fiber into gigabit range. Consequently, in many scenarios the bandwidth requirements of individual traffic requests are much lower than the transport capacity of individual wavelength channel. This observation suggests the possibility of packing several low rate traffic streams carrying data from individual requests, onto each available wavelength channel. This is referred to as *traffic grooming*. Usually the low rate data streams are groomed onto the wavelength channel using Time Division Multiplexing. Each individual wavelength channel is partitioned in time into fixed length *timeslots*. An individual low rate traffic stream can be assigned a set of timeslots on a wavelength channel. We refer to these timeslots on individual wavelengths as the *sub-wavelength channels*. The actual packing of low rate traffic on wavelength channels is carried out by using electronic devices known as Add-Drop Multiplexers (ADMs). An ADM is a device required to add (retrieve) sub-wavelength traffic onto (from) a particular wavelength channel. On receiving a wavelength channel, the ADM corresponding to that particular wavelength, can add/drop timeslots on the wavelength channel without disrupting the onward transmission of the other timeslots on the wavelength. Obviously for a lightpath routed on a particular wavelength, ADMs corresponding to that wavelength are required at both the start and the end nodes of the lightpath. Usually the network nodes do not have the ability to rearrange the timeslots on a wavelength channel. As a result, in case of optical switching, a *sub-wavelength-continuity constraint* (similar to the wavelength-continuity constraint desribed in Section 1.1.2) must be respected while implementing traffic grooming.

Analogous to the two flavors of the routing and wavelength assignment problem, the traffic grooming problem also has static and dynamic variants. As before, we only study the static traffic grooming problem, i.e., we assume that the set of traffic requests is known in advance.

For the problem of static traffic grooming in all-optical networks, usually the objective is to minimize the network cost which includes both the cost of optics and the cost of electronics. As stated before, the number of wavelengths that can be placed on a single optical fiber is a limiting resource. Hence, the number of wavelengths required per fiber is a fair measure of the cost of optics in the network. The cost of electronics is usually estimated by the number of Add-Drop Multiplexers (ADMs) required in the network.

The interested reader is referred to [4] and [5] for a review of the problem of grooming sub-wavelength traffic in WDM networks.

### 1.1.4 Multicasting in All-Optical WDM Networks

Most of the early work on both routing and wavelength assignment, and traffic grooming in WDM based all-optical networks, has concentrated on the scenario where the given traffic requests are unicast (single source-single destination) in nature. But multicasting (single source-multiple destinations) is an important technology which is tailor made for catering to several upcoming applications such as multimedia conferencing, video distribution, collaborative processing, etc. Therefore, studying both the problems of routing and wavelength assignment, and traffic grooming for multicast traffic in WDM based all-optical networks is of extreme importance.

Multicasting in WDM based all-optical networks involves setting up of *light-trees*, which are an obvious extension of the concept of lightpaths. A light tree can be viewed as a clear optical tree rooted at the source node and spanning the set of destinations. To support multicasting in all-optical networks via light-trees, the network nodes must be equipped with optical splitters and must have tap-and-continue capability [6]. As the name suggests, network nodes equipped with optical splitters can *split* any incoming wavelength channel onto multiple outgoing ports. This is required at nodes on the light-tree where there is a bifurcation. On the other hand, tap-and-continue capability at a network node means the ability to tap a small amount of light from a wavelength channel and use it for electronic processing while the rest of the light is switched in optical domain. This capability is required at the intermediate nodes of a light-tree which are also in the destination set of the multicast request being serviced by the light-tree. Obviously the number of times a particular wavelength channel can be tapped or split is upper bounded due to various phenomena such as power loss, distortion, etc., introduced by each such operation. These considerations require the placement of devices such as optical regenerators in the network that are used to boost the power of the optical signal. In this work, we ignore such restrictions.

The interested reader is referred to [7] for a review of the problem of multicasting in WDM networks.

## 1.2 Contributions

In this work, we develop and analyze algorithms for the problems of multicast routing and wavelength assignment, and multicast traffic grooming in all-optical WDM networks. We restrict our study to the static case, i.e., we assume that the set of multicast traffic requests is known in advance.

For the problem of routing and wavelength assignment of multicast traffic in all-optical WDM networks, we consider the objective of minimizing the total number of wavelengths required in the network. With this objective function, the problem is known to be hard in general topologies [8], even when the traffic requests are restricted to being unicast.

For the problem of multicast traffic grooming in all-optical WDM networks, we consider two different cost functions: (i) number of wavelengths required in the network (cost of optics), and (ii) number of ADMs required in the network (cost of electronics). We put an additional restriction that the bandwidth requirement of the individual multicast traffic requests are identical and are an integral fraction of the bandwidth available on individual wavelength channels. As a consequence of this assumption, without loss of any generality, we can further assume that the bandwidth requirement of the individual multicast traffic requests is equal to the capacity of the sub-wavelength channel. Observe that traffic grooming is a generalization of the routing and wavelength assignment problem. It is hardly surprising that for both the cost functions, even the unicast traffic grooming problem is hard in general topologies [5].

Since both the routing and wavelength assignment, and the traffic grooming problems are hard for general network topologies, in this work, we put restrictions on the network topologies. In particular, we study the multicast routing and wavelength assignment in all-optical bidirected trees, and the multicast traffic grooming problem in all-optical unidirectional rings. The motivation for selecting bidirected trees and unidirectional rings is that these simple structures are almost pervasive in the currently deployed fiber-optic networks. Moreover, there is a hope that analyzing the problems in these simple topologies could give insights into the problem for more general network topologies.

### 1.2.1 Multicast Wavelength Assignment in Bidirected Trees

A bidirected tree is the directed graph generated from a tree by replacing each edge of the tree by a pair of anti-parallel directed edges. We study the problem of routing and wavelength assignment for a given set of multicast traffic requests when the underlying fiber network is a bidirected tree and all-optical networking paradigm is employed. As stated before, we try to minimize the number of wavelengths required.

In Chapter 5, we prove that the problem is hard even for the cases when the bidirected tree is restricted to being (i) a bidirected path (i.e., the degree of the bidirected tree is restricted to being 2), and (ii) a bidirected star (i.e, the depth of the bidirected tree is restricted to being 2). Since the problem is hard even when the degree of the bidirected tree is 2, we restrict our study to the case when the degree of the bidirected tree is bounded. In particular, in Chapter 3, we present GREEDY-WA, an algorithm for the multicast routing and wavelength assignment problem in all-optical bidirected trees restricted to the case when the degree of the given bidirected tree is at most 3. We analyze the worst case performance guaranty and the time complexity for GREEDY-WA and prove that it is a $\frac{5}{2}$-approximation algorithm. In Chapter 4, we present SUBTREE-BASED-WA, an algorithm for the multicast routing and wavelength assignment problem in all-optical bidirected trees restricted to the case when the degree of the given bidirected tree is at most 4. Again, we analyze the worst case performance guaranty and the time complexity and prove that SUBTREE-BASED-WA is an approximation algorithm with approximation ratio $\frac{10}{3}$, 3 and 2 for the case when the degree of the given bidirected tree is equal to 4, 3 and 2, respectively. Finally, in Chapter 5, we prove that the problem of finding the clique number of conflict graphs of rooted subtrees of bidirected trees having degree 3 is hard. This result is interesting because the problem of determining the clique number of such conflict graphs arise when we try to establish a 'good' lower bound on the number of wavelengths required for various instances of the multicast routing and wavelength assignment problem of interest.

The work presented in Chapters 3 and 4 was published as [9], and parts of the work discussed in Chapter 5 was presented as [10].

### 1.2.2 Multicast Traffic Grooming in Unidirectional Rings

A unidirectional ring is the directed graph generated from a cycle by replacing each edge of the cycle by a directed edge such that the in-degree and the out-degree of every vertex is unity. We study the problem of grooming a given set of multicast traffic requests when the underlying fiber network is a unidirectional ring and all-optical networking paradigm is employed. As stated before, we study two objective functions: (i) minimizing the number of wavelengths required in the network, and (ii) minimizing the number of ADMs required in the network. In Chapter 7, we present three traffic grooming algorithms for this problem: ARC-COL-BASED-TG, RANDOM-TG, ITER-IMPROVE-TG. We prove that, given any $\alpha$-approximation algorithm for the problem of coloring circular arc graphs, ARC-COL-BASED-TG has an asymptotic approximation ratio of $\alpha$ for the grooming problem of interest with the objective of minimizing the number of wavelengths required in the network. We also develop an easy to calculate lower bound on the number of ADMs required by any traffic grooming solution for a given instance of the problem. We use this lower bound to analyze the worst case performance of both RANDOM-TG and ARC-COL-BASED-TG, as measured in terms of the number of ADMs required in the network. Finally, via extensive simulations, we compare the average performance of the three grooming schemes as well as the behavior of the developed lower bound. During the simulations, we also compare the average performance of CIRCLE-BASED-TG, which is a multicast extension of the unicast traffic grooming scheme developed in [11] for all-optical unidirectional ring networks. We analyze the time complexities of the various grooming schemes and discuss simulation results in detail.

The work presented in Chapter 7 was published as [12].

## 1.3 A Word on Notation

In this section we state the recurring notations, concepts and assumptions that are used in this work. This is not a comprehensive list and we introduce more notations in the text as and when required.

### 1.3.1 Basic Notation

We use ':=' to signify 'is defined to be equal to'. We denote the cardinality of a finite set $S$ by $|S|$. For real valued $x$, $[x]^+ := \max\{x, 0\}$. We denote the image of any mapping $f : D \longrightarrow R$, restricted to some set $S \subseteq D$, by $f(S)$, i.e., $f(S) := \{r \in R : r = f(s) \text{ for some } s \in S\}$.

Unless otherwise specified, all the graphs (both undirected and directed) are assumed to be simple. For a given graph $G$, we denote the edge set by $E_G$ and the vertex set by $V_G$. An edge between vertices $u, v \in V_G$ is denoted by the binary set $\{u, v\}$. Similarly, for a given directed graph $\vec{G}$, we denote the set of directed edges by $E_{\vec{G}}$ and the set of vertices by $V_{\vec{G}}$. For a pair of vertices $u, v \in V_{\vec{G}}$, a directed edge from $u$ to $v$ is denoted by the ordered pair $(u, v)$. We denote the degree of a vertex $v$ of graph $G$ by $\delta_G(v)$ and the degree of the graph by $\Delta_G := max_{v \in V_G} \delta_G(v)$. We denote the in-degree of a vertex $v$ of directed graph $\vec{G}$ by $\delta^{\mathrm{i}}_{\vec{G}}(v)$ and the in-degree of the directed graph by $\Delta^{\mathrm{i}}_{\vec{G}} := \max_{v \in V_{\vec{G}}} \delta^{\mathrm{i}}_{\vec{G}}(v)$. Similarly, we denote the out-degree of a vertex $v$ of directed graph $\vec{G}$ by $\delta^{\mathrm{o}}_{\vec{G}}(v)$, and the out-degree of the directed graph by $\Delta^{\mathrm{o}}_{\vec{G}} := \max_{v \in V_{\vec{G}}} \delta^{\mathrm{o}}_{\vec{G}}(v)$.

The undirected multigraph obtained by replacing all the directed edges of directed graph $\vec{G}$ by undirected edges is denoted by $\|\vec{G}\|$ and is referred to as the *skeleton* of $\vec{G}$. Hence, $V_{\|\vec{G}\|} := V_{\vec{G}}$, and corresponding to every directed edge $(u, v)$ in $E_{\vec{G}}$, there is an undirected edge $\{u, v\}$ in $E_{\|\vec{G}\|}$. Observe that in general the skeleton of a simple directed graph is a multigraph and not a simple graph, i.e., in general $E_{\|\vec{G}\|}$ is a multiset. This is because a simple directed graph $\vec{G}$ is allowed to have a pair of anti-parallel edges $(u, v)$ and $(v, u)$, but in that case its skeleton $\|\vec{G}\|$ has a pair of undirected edges between vertices $u$ and $v$.

We denote the complement of a graph $G$ by $\bar{G}$, i.e., $V_{\bar{G}} := V_G$ and $E_{\bar{G}} := \{\{u, v\} : u, v \in V_G \text{ and } \{u, v\} \notin E_G\}$. A subgraph of a graph $G$ is any graph $S$ with vertex set $V_S \subseteq V_G$ and edge set $E_S \subseteq \{\{u, v\} \in E_G : u, v \in V_S\}$. Similarly, a subgraph of a directed graph $\vec{G}$ is any directed graph $\vec{S}$ with vertex set $V_{\vec{S}} \subseteq V_{\vec{G}}$ and directed edge set $E_{\vec{S}} \subseteq \{(u, v) \in E_{\vec{G}} : u, v \in V_{\vec{S}}\}$. The subgraph of graph $G$ induced by vertex set $W \subseteq V_G$ is denoted by $G[W]$, and is defined as having vertex set $V_{G[W]} := W$ and edge set $E_{G[W]} := \{\{u, v\} \in E_G : u, v \in W\}$. Similarly, the subgraph of graph $G$ induced by edge set $F \subseteq E_G$ is denoted by $G[F]$, and is defined as having edge set $E_{G[F]} := F$ and vertex set $V_{G[F]} := \{v : \exists \{u, v\} \in F\}$.

A directed graph $\vec{R}$ is said to be a *rooted tree* if (i) its skeleton $\|\vec{R}\|$ is a tree, and (ii) there is a unique vertex $r \in V_{\vec{R}}$ with in-degree 0, and every other vertex has in-degree 1, i.e.,

$$\delta^{\mathrm{i}}_{\vec{R}}(v) = \begin{cases} 1 & \text{if } v \in V_{\vec{R}} \setminus \{r\}, \\ 0 & \text{if } v = r. \end{cases}$$

In this case, $r$ is said to be the *root* of $\vec{R}$, and $\vec{R}$ is said to be a tree rooted at $r$. Any vertex of the rooted tree $\vec{R}$ with out-degree 0 is said to be a *leaf* of $\vec{R}$. Given two directed graphs $\vec{R}$ and $\vec{G}$, $\vec{R}$ is said to be a *rooted subtree* of $\vec{G}$ with root $r$ (or equivalently, rooted at $r$) if it is a rooted tree with $r$ as the root, and it is a subgraph of $\vec{G}$.

Let $\mathcal{R}$ be a set of rooted subtrees of directed graph $\vec{G}$. We denote the set of all the rooted subtrees in $\mathcal{R}$ that contain directed edge $(u,v) \in E_{\vec{G}}$ by $\mathcal{R}[(u,v)]$, i.e., $\mathcal{R}[(u,v)] := \{\vec{R} \in \mathcal{R} : (u,v) \in E_{\vec{R}}\}$. If a rooted subtree $\vec{R}$ contains directed edge $(u,v)$, i.e., if $\vec{R} \in \mathcal{R}[(u,v)]$, we say that it is *present* on the directed edge $(u,v)$. Moreover, the set $\mathcal{R}$ of rooted subtrees of the directed graph $\vec{G}$ *collide* on some directed edge $(u,v) \in E_{\vec{G}}$, if for every rooted subtree $\vec{R} \in \mathcal{R}$, $(u,v) \in E_{\vec{R}}$. If the directed edge on which the collision occurs is not important for the subsequent discussion, we simply say that the set of rooted subtrees collide. If the set of leaves of a rooted subtree $\vec{R}$ of a directed tree $\vec{G}$, rooted at vertex $r \in V_{\vec{G}}$, is singleton $\{l\} \subseteq V_{\vec{G}} \setminus \{r\}$, $\vec{R}$ is called a *directed path* on $\vec{G}$ with *start vertex* $r$ and *end vertex* $l$. All the terminology described above for rooted subtrees, is extended in obvious manner for directed paths as well.

### 1.3.2 Independence, Cliques, Matching, Coloring

Given a graph $G$, a set of vertices $I \in V_G$ is said to be *independent* if there is no edge $\{u,v\}$ in the graph such that both the vertices $u,v \in I$, i.e., $E_{G[I]} = \emptyset$. In general, largest independent set of vertices in a graph is not unique. Any largest independent set of vertices of the graph $G$ is said to be a *maximum independent set* of $G$; and its size is said to be the *independence number* of $G$, and is denoted by $\alpha_G$. A set of vertices $I \in V_G$ is said to be a *clique* in the graph $G$, if for every pair of vertices $u, v \in C$, there is an edge $\{u, v\}$ in the graph, i.e., $G[C]$ is a complete graph. In general, largest clique in a graph is not unique. Any largest clique of the graph $G$ is said to be a *maximum clique* of $G$; and its size is said to the *clique number* of $G$, and is denoted by $\omega_G$. A set of edges $M \in E_G$ is said to be a *matching* in the graph $G$, if the degree of every vertex in the graph $G[M]$ is unity, i.e., $\delta_{G[M]}(v) = 1$ for every $v \in V_{G[M]}$. In general, largest matching in a graph is not unique. Any largest matching of the graph $G$ is called a *maximum matching* of $G$.

A *vertex coloring* of graph $G$ is a map $\psi : V_G \longrightarrow \mathbb{N} := \{1, 2, \ldots\}$ such that for any pair of vertices $u, v \in V_G$, if $\{u, v\} \in E_G$ then $\psi(u) \neq \psi(v)$. The *color* of vertex $v$ of graph $G$ according to the coloring $\psi$ is given by $\psi(v)$. According to the notation described above, the set of colors assigned to vertex set $W \subseteq V_G$ according to coloring $\psi$, is denoted by $\psi(W)$. Hence, the total number of colors used by vertex coloring $\psi$ is $|\psi(V_G)|$. We denote the set of all the vertex colorings of graph $G$ by $\Psi_G$. A *minimum vertex coloring* of graph $G$ is any vertex coloring that uses at most as few colors as any other vertex coloring of the graph. In general, minimum vertex coloring of a graph is not unique. The number of colors used in any minimum vertex coloring of the graph $G$ is called the *chromatic number* of $G$ and is denoted by $\chi_G$, i.e., $\chi_G := \min_{\psi \in \Psi_G} |\psi(V_G)|$.

### 1.3.3 Conflict Graphs

Consider a set of objects $\mathcal{J} = \{J_1, J_2, \ldots, J_{|\mathcal{J}|}\}$ such that associated with every object $J \in \mathcal{J}$, there is a set $S_J$. The *conflict graph* of the set $\mathcal{J}$ is formed by creating a vertex corresponding to each object in the set $\mathcal{J}$, and creating an edge between two vertices whenever the sets associated with the corresponding objects have a nonempty intersection. We denote the conflict graph of the set $\mathcal{J}$ by $G_{\mathcal{J}}$. For ease of exposition, we reuse the labels of the objects in the set $\mathcal{J}$ for the corresponding vertices of the conflict graph $G_{\mathcal{J}}$. In other words, $V_{G_{\mathcal{J}}} := \mathcal{J}$ and $E_{G_{\mathcal{J}}} := \{\{J_i, J_j\} : J_i, J_j \in \mathcal{J} \text{ and } S_{J_i} \cap S_{J_j} \neq \emptyset\}$. Observe that for any subset $\mathcal{I} \subseteq \mathcal{J}$ of the objects, the conflict graph $G_{\mathcal{I}}$ is the subgraph of the conflict graph $G_{\mathcal{J}}$ induced by the vertices corresponding to the objects in the set $\mathcal{I}$, i.e., $G_{\mathcal{I}} = G_{\mathcal{J}}[\mathcal{I}]$.

As an example, consider a graph $G$ and a set $\mathcal{H} = \{H_1, \ldots, H_{|\mathcal{H}|}\}$ of subgraphs of $G$. For any subgraph $H \in \mathcal{H}$, consider its edge set $E_H$ to be the set associated with $H$. In this case, the conflict graph $G_{\mathcal{H}}$ has edges between pairs of the subgraph in the set $\mathcal{H}$ that share some common edge in the graph $G$. Observe that the definition of the conflict graph depends on the associated sets that we select. For instance if instead of selecting the edge set, we had selected the vertex set of subgraphs as the associated sets, we would have ended up with a different conflict graph. The selection of the associated sets depends on exactly what we want to model with the conflict graph.

The concept of conflict graphs as defined here is borrowed from the well known concept of intersection graphs [13]. We use a different terminology to facilitate the idea that for most of the time we would be looking at sets of traffic requests, and the set associated with any traffic request would be the resources (such as the wavelength or the subwavelength channel) that it requires. In this sense, the edges of the generated conflict graph model the pairwise conflicts for common resources among the traffic requests.

## 1.4 Modeling Multicasting in All-Optical Networks

In this section, we first discuss the models that we use for modeling optical networks and multicast traffic requests. Next, we apply these models to describe the routing and wavelength assignment problem, and the traffic grooming problem in general network topologies.

### 1.4.1 Fiber Network and Multicast Traffic Requests

We represent an optical fiber network as a directed graph where the vertices model the network nodes and the directed edges model the fiber links. For the purpose of illustration, let the directed graph $\vec{G}$ represent a fiber network. A multicast traffic request on this network is modeled as a pair $\{s, D\}$, where $s$ is the source node and $D$ is the set of destination nodes corresponding to the traffic request. It is clear that $s \in V_{\vec{G}}$ and $D \subseteq V_{\vec{G}} \setminus \{s\}$.

### 1.4.2 Routing and Wavelength Assignment

As described in Section 1.1.4, under the all-optical networking paradigm, a multicast traffic request $\{s, D\}$ is supported on the fiber network $\vec{G}$ by constructing a light-tree with the source node $s$ as the root and the set of destination nodes $D$ as leaves. Technically, any possible routing solution for the light-tree corresponding to the multicast traffic request $\{s, D\}$ is nothing but a subgraph $\vec{S}$ of the fiber network $\vec{G}$ satisfying the property that it contains a directed path from the source node $s$ to every node in the destination set $D$. Observe that any such subgraph $\vec{S}$ must necessarily contain some rooted subtree $\vec{R}$ satisfying the following:

(i) It is rooted at the source node $s$, i.e., $\delta^{\mathrm{i}}_{\vec{R}}(s) = 0$.

(ii) It spans the set of destination nodes $D$, i.e., $D \subseteq V_{\vec{R}}$.

(iii) Every leaf vertex is a destination node, i.e., $\{v \in V_{\vec{R}} : \delta^{\mathrm{o}}_{\vec{R}}(v) = 0\} \subseteq D$.

Moreover, observe that the rooted subtree $\vec{R}$ of the fiber network $\vec{G}$ can also be viewed as a possible routing solution for the light-tree corresponding to the multicast traffic request $\{s, D\}$. Since $\vec{R}$ is a subgraph of $\vec{S}$, in terms of the resources (wavelength channels, fibers, etc.) required, using the rooted subtree $\vec{R}$ as the routing solution is no worse than using $\vec{S}$. Hence, for routing the light-tree corresponding to any multicast traffic request $\{s, D\}$, it is justified to only consider the rooted subtrees of the fiber network that satisfy the above listed properties. We denote this set of interesting rooted subtrees of the given directed graph $\vec{G}$ for a given multicast traffic request $\{s, D\}$ by $\mathcal{R}_{\{\vec{G},\{s,D\}\}}$. In light of this observation, the problem of routing and wavelength assignment for a given set of multicast traffic requests $\mathcal{M} = \{\{s_1, D_1\}, \{s_2, D_2\}, \ldots, \{s_{|\mathcal{M}|}, D_{|\mathcal{M}|}\}\}$ on the given fiber network $\vec{G}$ under all-optical networking paradigm, can be defined as follows.

**Problem 1.1** (MIN-MC-RWA)**.** *Given a pair $\{\vec{G}, \mathcal{M}\}$, where $\vec{G}$ is a directed graph and $\mathcal{M}$ is a set of multicast traffic requests on $\vec{G}$, determine a pair of mappings $\{\pi, \lambda\}$ as described next.*

- *(i) Mapping $\pi$ solves the routing problem in the sense that it maps each multicast traffic request $\{s, D\} \in \mathcal{M}$ to a rooted subtree $\pi(\{s, D\}) := \vec{R}_{\{s,D\}}$ of the directed tree $\vec{G}$ that determines the routing for the multicast traffic request $\{s, D\}$.*

- *(ii) Mapping $\lambda : \mathcal{M} \longrightarrow \mathbb{N}$ solves the wavelength assignment problem in the sense that it maps each multicast traffic request $\{s, D\} \in \mathcal{M}$ to a wavelength (described as a positive integer).*

- *(iii) Jointly the two mappings should satisfy the constraint that for every pair of multicast traffic requests $\{s_i, D_i\}, \{s_j, D_j\} \in \mathcal{M}$, if the rooted subtrees $\vec{R}_{\{s_i,D_i\}}$ and $\vec{R}_{\{s_j,D_j\}}$ collide, then $\lambda(\{s_i, D_i\}) \neq \lambda(\{s_j, D_j\})$.*

*The objective is to minimize $|\lambda(\mathcal{M})|$, the total number of wavelengths required.*

### 1.4.3 Traffic Grooming

As described in Section 1.2.2, in the traffic grooming problem we assume that the given traffic requests have sub-wavelength bandwidth requirements. Moreover, we assume that the traffic is homogeneous in the sense that the bandwidth requirements of all the traffic requests are the same, and are equal to an integral fraction of the bandwidth capacity of a single wavelength channel. In particular, we assume that at most $g$ of these low rate traffic requests can be simultaneously placed on a single wavelength channel. We refer to $g$ as the *grooming ratio.*

Based on the discussions in sections 1.1.3, 1.1.4 and 1.4.2, grooming a given set of multicast traffic requests can be modeled as follows.

**Definition 1.2** (MC-TG). *Given a triple $\{\vec{G}, \mathcal{M}, g\}$, where $\vec{G}$ is a directed graph, $\mathcal{M}$ is a set of multicast traffic requests on $\vec{G}$, and $g$ is a positive integer; a traffic grooming solution is a triple of mappings $\{\pi, \lambda, \omega\}$ as described next.*

*(i) Mapping $\pi$ solves the routing problem in the sense that it maps each multicast traffic request $\{s, D\} \in \mathcal{M}$ to a rooted subtree $\pi(\{s, D\}) := \vec{R}_{\{s,D\}}$ of the directed tree $\vec{G}$ that determines the routing of the multicast traffic request $\{s, D\}$.*

*(ii) Mapping $\lambda : \mathcal{M} \longrightarrow \mathbb{N}$ solves the wavelength assignment problem in the sense that it maps each multicast traffic request $\{s, D\} \in \mathcal{M}$ to a wavelength (described as a positive integer).*

*(iii) Mapping $\omega : \mathcal{M} \longrightarrow \mathbb{N}$ solves the sub-wavelength channel assignment problem in the sense that it maps each multicast traffic request $\{s, D\} \in \mathcal{M}$ to a sub-wavelength channel (described as a positive integer).*

*(iv) Jointly the three mappings should satisfy the constraints that for every pair of multicast traffic requests $\{s_i, D_i\}, \{s_j, D_j\} \in \mathcal{M}$, if the rooted subtrees $\vec{R}_{\{s_i,D_i\}}$*

*and* $\vec{R}_{\{s_j,D_j\}}$ *collide, then* $(\lambda(\{s_i, D_i\}), \omega(\{s_i, D_i\})) \neq (\lambda(\{s_j, D_j\}), \omega(\{s_j, D_j\}))$*; and the number of sub-wavelength channels in any wavelength must not exceed* $g$*, i.e.,* $max_{k \in \lambda(\mathcal{M})} |\omega(\{\{s, D\} \in \mathcal{M} : \lambda(\{s, D\}) = k\})| \leq g$*.*

In particular, we define the problem of grooming multicast traffic with the objective of minimizing the total number of wavelengths required as follows.

**Problem 1.3** (MIN-WAVE-MC-TG)**.** *Given a triple* $\{\vec{G}, \mathcal{M}, g\}$*, where* $\vec{G}$ *is a directed graph,* $\mathcal{M}$ *is a set of multicast traffic requests on* $\vec{G}$*, and* $g$ *is a positive integer; determine a traffic grooming solution* $\{\pi, \lambda, \omega\}$ *as defined in MC-TG, with the objective of minimizing* $|\lambda(\mathcal{M})|$*, the total number of wavelengths required.*

Similarly, we define the problem of grooming multicast traffic with the objective of minimizing the total number of ADMs required as follows.

**Problem 1.4** (MIN-ADM-MC-TG)**.** *Given a triple* $\{\vec{G}, \mathcal{M}, g\}$*, where* $\vec{G}$ *is a directed graph,* $\mathcal{M}$ *is a set of multicast traffic requests on* $\vec{G}$*, and* $g$ *is a positive integer; determine a traffic grooming solution* $\{\pi, \lambda, \omega\}$ *as defined in MC-TG, with the objective of minimizing the total number of ADMs required. The number of ADMs required by the traffic grooming solution* $\{\pi, \lambda, \omega\}$ *can be determined as* $\sum_{v \in V_{\vec{G}}} |\lambda(\mathcal{M}_v)|$*, where for any vertex* $v \in V_{\vec{G}}$*,* $\mathcal{M}_v$ *is defined to be the set of all the multicast traffic requests that have vertex* $v$ *as the source node or as one of the destination nodes, i.e.,* $\mathcal{M}_v := \{\{s, D\} \in \mathcal{M} : v \in \{s\} \cup D\}$*.*

In the MIN-ADM-MC-TG problem, we are calculating the total number of ADMs required at each network node and then summing this for all the nodes of the network. Since we assume all-optical networking paradigm, the number of ADMs required at any network node by a traffic grooming solution is simply the number of wavelengths required by the set of multicast traffic requests for which that node acts as either the source node or as one of the destination nodes.

## 1.5 Organization

The rest of the thesis is organized as follows. Chapters 2-5 are dedicated to the problem of routing and wavelength assignment for multicast traffic in all-optical bidirected trees. In Chapter 2, we define and model the exact routing and wavelength assignment problem that we study, and review the work that is most closely related to this problem. In Chapter 3, we present a greedy scheme for the problem. We analyze its worst case performance as well as its time complexity. In Chapter 4, we present another, simpler strategy for the problem. Again, we analyze worst case performance as well as the time complexity. In Chapter 5, we state and prove some NP completeness results for various restricted versions of the problem. Chapters 6 and 7 are dedicated to the problem of grooming multicast traffic in all-optical unidirectional rings. In Chapter 6, we define and model the exact problem, and review the related work. In Chapter 7, we present several schemes for the problem. We analyze the performance of these schemes, either analytically or by simulations. The time complexity of various schemes is also studied. Finally, in Chapter 8 we provide a short conclusion of this work and list some interesting directions for future research.

# Chapter 2

## Multicast Wavelength Assignment in Bidirected Trees

As stated in Section 1.2.1, we are interested in the problem of routing and wavelength assignment for multicast traffic in all-optical bidirected tree networks. In this chapter, we define the exact problem that we wish to study. We also present the work that is closely related to our problem of interest.

### 2.1 Model

As described in Section 1.2.1, a bidirected tree is a directed graph that is generated from some given tree $H$ by replacing all the edges of $H$ by pairs of anti-parallel directed edges. The bidirected tree thus generated is denoted by $\vec{T}_H$ and the tree $H$ is referred to as its *host tree*. The degree of the bidirected tree $\vec{T}_H$, denoted by $\Delta_{\vec{T}_H}$, is defined to be equal to the degree of the host tree $H$, i.e., $\Delta_{\vec{T}_H} := \Delta_H$. Let $\mathcal{R}$ be a set of rooted subtrees of bidirected tree $\vec{T}_H$. With a slight abuse of the notation introduced in Section 1.3, for any host tree edge $\{u, v\} \in E_{\vec{T}_H}$, we denote the set of all the rooted subtrees in $\mathcal{R}$ that contain directed edges $(u, v)$ or $(v, u)$ by $\mathcal{R}[\{u, v\}]$, i.e., $\mathcal{R}[\{u, v\}] := \{\vec{R} \in \mathcal{R} : \{u, v\} \in E_{\|\vec{R}\|}\}$. Observe that for any host tree edge $\{u, v\}$, sets $\mathcal{R}[(u, v)]$ and $\mathcal{R}[(v, u)]$ partition[1] the set $\mathcal{R}[\{u, v\}]$. Again extending the terminology introduced in Section 1.3, if a rooted subtree $\vec{R}$ contains directed edges $(u, v)$ or $(v, u)$, i.e., if $\vec{R} \in \mathcal{R}[\{u, v\}]$, we say that it is present on the host tree edge $\{u, v\}$.

Consider a restricted instance $\{\vec{T}_H, \mathcal{M}\}$ of the MIN-MC-RWA problem stated in Section 1.4, where the fiber network $\vec{T}_H$ is a bidirected tree, and $\mathcal{M}$ is a set of multicast traffic requests on $\vec{T}_H$. Observe that for a given multicast traffic request $\{s, D\}$ on the bidirected tree $\vec{T}_H$, there is a unique rooted subtree of the bidirected tree that satisfies the properties stated in Section 1.4.2, i.e, the set $\mathcal{R}_{\{\vec{T}_H, \{s,D\}\}}$ of interesting rooted subtrees contains exactly one rooted subtree. As a consequence, the routing of the light-tree corresponding to any give multicast traffic request in the set $\mathcal{M}$ is fixed, i.e., the mapping $\pi$ described in MIN-MC-RWA is trivially determined. Hence, the routing and wavelength assignment problem MIN-MC-RWA simply reduces to a problem of assigning wavelengths to the set of rooted subtrees of the given bidirected tree, corresponding to the given set of multicast traffic requests.

## 2.2 Problem Statement

As discussed in Section 2.1, the MIN-MC-RWA problem when restricted to a bidirected tree network, simply reduces to the problem of assigning wavelengths to a set of rooted subtrees of the bidirected tree. More precisely, we can define the exact problem as follows.

**Problem 2.1** (MIN-MC-WA-BT)**.** *Given a pair $\{\vec{T}_H, \mathcal{R}\}$, where $\vec{T}_H$ is a bidirected tree and $\mathcal{R}$ is a set of rooted subtrees on $\vec{T}_H$; consider a set of mappings $\Lambda_{\{\vec{T}_H, \mathcal{R}\}}$ from $\mathcal{R}$ to $\mathbb{N}$, such that for any mapping $\lambda \in \Lambda_{\{\vec{T}_H, \mathcal{R}\}}$, if a pair of rooted subtrees $\vec{R}_i, \vec{R}_j \in \mathcal{R}$ collide, then $\lambda(\vec{R}_i) \neq \lambda(\vec{R}_j)$.*

[1] Sets $A_0, \ldots, A_K$ are said to partition set $A$ if $\bigcup_{i=0}^{K} A_i = A$ and $A_i \cap A_j = \emptyset$ for every $i \neq j$ where $i, j \in \{0, \ldots, K\}$. In this case sets $A_0, \ldots, A_K$ are referred to as the partitions of set $A$.

*Determine a mapping* $\lambda^* \in \Lambda_{\{\vec{T}_H, \mathcal{R}\}}$ *that uses the minimum number of wavelengths, i.e.,* $\lambda^* \in \underset{\lambda \in \Lambda_{\{\vec{T}, \mathcal{R}\}}}{\operatorname{argmin}} |\lambda(\mathcal{R})|$.

Given an instance $\{\vec{T}_H, \mathcal{R}\}$ of the MIN-MC-WA-BT problem described above, consider the conflict graph $G_{\mathcal{R}}$ of the set $\mathcal{R}$ of rooted subtrees of the bidirected tree $\vec{T}_H$. The conflicts modeled in this graph correspond to all the pairwise collisions between the rooted subtrees in the set $\mathcal{R}$, i.e., for any pair of rooted subtrees $\vec{R}_i, \vec{R}_j \in \mathcal{R}$, there is an edge $\{\vec{R}_i, \vec{R}_j\} \in E_{G_{\mathcal{R}}}$ in the conflict graph if and only if they collide and therefore, cannot be assigned the same wavelength. It is straightforward to argue that assigning wavelengths to the set $\mathcal{R}$ of rooted subtrees of the bidirected tree $\vec{T}_H$ is equivalent to coloring the vertices of the conflict graph $G_{\mathcal{R}}$, where each color signifies a wavelength. In particular, solving the MIN-MC-WA-BT problem for the instance $\{\vec{T}_H, \mathcal{R}\}$ is equivalent to the problem of finding a minimum vertex coloring of the corresponding conflict graph $G_{\mathcal{R}}$.

In this work we shall look at the MIN-MC-WA-BT problem restricted to the case where the bidirected tree $\vec{T}_H$ has bounded degree, i.e., $\Delta_{\vec{T}_H} \leq d$ for some fixed value of $d$. In particular, we shall study the problem when $d \in \{2, 3, 4\}$. We shall see in Section 2.3 that the problem is known to be hard for all values of $d \geq 3$. Later, in Section 5.3, we prove that the problem is hard even for $d = 2$.

## 2.3 Related Work

As described in Section 2.2, any given instance of the problem MIN-MC-WA-BT can be recast as the problem of coloring the given set of rooted subtrees on the given bidirected tree.[2] The work that is most closely related to the problem of coloring a given set of rooted subtrees of a bidirected tree, and hence to the

[2]By coloring a set of rooted subtrees, we mean the vertex coloring of the corresponding conflict graph. Similar notation is used for coloring subtrees, paths, directed paths, arcs, etc.

MIN-MC-WA-BT problem, consists of the following:

(i) Coloring a given set of paths on a tree.

(ii) Coloring a given set of directed paths on a bidirected tree.

(iii) Coloring and characterization of a given set of subtrees of a tree.

Since, the MIN-MC-WA-BT problem is equivalent to the problem of coloring a given set of rooted subtrees of a bidirected tree, from an theoretical perspective, our contribution can be viewed as the next logical step in this series of works.

In [14], Golumbic et al. proved that coloring a given set of paths on a tree is NP complete in general. They showed that path coloring in stars is equivalent to edge coloring in multigraphs. Since edge coloring is NP complete [15], path coloring in stars is also NP complete. In fact, as observed in [16], this equivalence result has several important implications:

(i) Path coloring is solvable in polynomial time in bounded degree trees.

(ii) Path coloring is NP complete for trees of arbitrary degrees (even with diameter 2, i.e., even for stars).

(iii) Any approximation algorithm for edge coloring in multigraphs can be transformed into an approximation algorithm for path coloring in trees and vice versa with the same approximation ratio.

(iv) For path coloring in trees of arbitrary degree, there is no polynomial time algorithm with approximation ratio $\frac{4}{3} - \epsilon$ for any $\epsilon > 0$ unless P=NP.

In [17], Tarjan introduced a $\frac{3}{2}$-approximation algorithm for coloring a given set of paths in a tree. Later, this ratio was rediscovered by Raghavan and Upfal [18] in the context of optical networks. Mihail et al. [19] presented a coloring scheme with an asymptotic approximation ratio of $\frac{9}{8}$. Nishizeki et al. [20] presented an algorithm for edge coloring multigraphs with an asymptotic approximation ratio of 1.1 and an absolute approximation ratio of $\frac{4}{3}$. This improves the asymptotic and the absolute approximation ratio of path coloring in trees to 1.1 and $\frac{4}{3}$, respectively. Recently in [21], Sanders et al. have developed an algorithm that can achieve arbitrarily good asymptotic approximation ratios for the problem of edge coloring in multigraphs.

In [22], Erlebach et al. proved that coloring a given set of directed paths in bidirected trees is NP complete. The result holds even when we restrict instances to arbitrary bidirected trees and sets of directed paths of *load* 3 or to bidirected trees with arbitrary degree and depth 3 [23]. Here by load of a set of directed paths, we mean the maximum number of directed paths in the set that share a directed edge. For this problem, Mihail et al. [19] gave a $\frac{15}{8}$-approximation algorithm. This ratio was improved to $\frac{7}{4}$ in [24] and [25], and finally to $\frac{5}{3}$ in [26]. All these are greedy, deterministic algorithms and use the load of the given set of directed paths as the lower bound on the number of colors required. In [26], Kaklamanis et al. also proved that no greedy, deterministic algorithm can achieve a better approximation ratio than $\frac{5}{3}$. Later, in [16], Erlebach et al. proved that there is no polynomial time algorithm for directed path coloring with approximation ratio $\frac{4}{3} - \epsilon$ for any $\epsilon > 0$ unless P=NP.

Unlike its undirected counterpart, Erlebach et al. [27] proved by a reduction from circular arc coloring that the problem of coloring directed paths is NP complete even in binary bidirected trees. In [25], Kumar et al. gave a problem instance where the given set of directed paths on a binary bidirected tree of depth 3 having load $l$ requires at least $\frac{5}{4}l$ colors. Caragiannis et al. [28] and Jansen [29] gave simple algorithms for the directed path coloring problem in binary bidirected trees having approximation ratio $\frac{5}{3}$ (the same as the approximation ratio for problem on general bidirected trees). In [30], Auletta et al. presented a randomized greedy algorithm for coloring a given set of directed paths of maximum load $l$ in binary bidirected trees of depth $O(l^{\frac{1}{3}-\epsilon})$ that uses at most $\frac{7}{5}l+o(l)$ colors. They also proved that with high probability, randomized greedy algorithms cannot achieve an approximation ratio better than $\frac{3}{2}$ when applied for binary bidirected trees of depth $\Omega(l)$, and $1.293-o(1)$ when applied for binary bidirected trees of constant depth. Moreover, they proved that an existential upper bound of $\frac{7}{5}l + o(l)$ holds on any binary bidirected tree.

In [31], Jamison et al. proved that the conflict graphs of subtrees in a binary tree are chordal [32], and therefore easily colorable [33]. In [34], Golumbic et al. proved that the conflict graphs of paths on trees having degree at most 4 are weakly chordal [35], therefore coloring them is easy [36]. Later, in [37], they extended the result to the conflict graph of subtrees on trees having degree at most 4.

For an extensive compilation of complexity results on coloring paths in trees and directed paths in bidirected trees from the perspective of optical networks, the reader is referred to [23] and [16]. And for a survey of algorithmic results, the reader is referred to [38], [39] and [40].

We should mention that ours is the first work to study the problem of coloring rooted subtrees of a bidirected tree (which may be seen as the directed counterpart of the problem of coloring subtrees of a tree).

# Chapter 3

## Greedy Multicast Wavelength Assignment in Bidirected Trees

In this chapter, we present and analyze a greedy strategy for a restricted version of the MIN-MC-WA-BT problem described in Section 2.2. The additional restriction that we place on the problem is to limit the degree of the bidirected tree to be at most 3. In other words, the problem under consideration is the MIN-MC-WA-BT problem represented as a pair $\{\vec{T}_H, \mathcal{R}\}$, where $\vec{T}_H$ is a bidirected tree with degree $\Delta_{\vec{T}_H} \leq 3$ and $\mathcal{R}$ is a set of rooted subtrees on $\vec{T}_H$. We prove that the presented greedy scheme is a $\frac{5}{2}$-approximation algorithm.

### 3.1 Greedy Wavelength Assignment

The algorithm proceeds in *rounds*. In each round we select and *process* a host tree edge which has not been selected in any of the previous rounds. Processing a host tree edge means assigning wavelengths to all the unassigned rooted subtrees present on that edge, where by *unassigned* rooted subtrees we refer to the set of rooted subtrees that have not yet been assigned any wavelengths. The key steps are the order in which the host tree edges are traversed for processing and the policy used to assign wavelengths to the set of unassigned rooted subtrees present on the edge being processed.

The complete scheme is given as Algorithm 1 (GREEDY-WA). We denote the wavelength assignment generated by the scheme by $\lambda^{\text{GDY}}$.

**Algorithm 1** GREEDY-WA

**Require:** MIN-MC-WA-BT problem instance $\{\vec{T}_H, \mathcal{R}\}$, where $\Delta_{\vec{T}_H} \leq 3$.

**Ensure:** A wavelength assignment $\lambda^{\text{GDY}} \in \Lambda_{\{\vec{T}_H, \mathcal{R}\}}$.

Perform a BFS on host tree $H$ starting with an arbitrary vertex as the root and enumerate the tree edges in the order of their discovery. Let $\{e_1, \ldots, e_{|E_H|}\}$ be the ordered set of edges $E_H$.

$\mathcal{P}_0 \leftarrow \emptyset$

**for** $i = 1$ to $|E_H|$ **do**

  $\mathcal{Q}_i \leftarrow \mathcal{R}[e_i] \setminus \mathcal{P}_{i-1}$

  **if** edge $e_i = \{u, v\}$ is of type (iv) as defined in Lemma 3.3 **then**

    Let $\lambda_1, \lambda_2 \in \Lambda_{\{\vec{T}_H, \mathcal{Q}_i \cup \mathcal{P}_{i-1}\}}$

    $\lambda_1(\vec{R}), \lambda_2(\vec{R}) \leftarrow \lambda^{\text{GDY}}(\vec{R})$ for every $\vec{R}_j \in \mathcal{P}_{i-1}$ (unassigned otherwise).

    PROCESS-EDGE-1($\vec{T}_H, \{u, v\}, \mathcal{P}_{i-1}, \mathcal{Q}_i, \lambda_1$)

    PROCESS-EDGE-2($\vec{T}_H, \{\{u, v\}, \{u, w\}, \{u, x\}\}, \mathcal{P}_{i-1}, \mathcal{Q}_i, \lambda_2$)

    **if** $|\lambda_1(\mathcal{P}_{i-1} \cup \mathcal{Q}_i)| \leq |\lambda_2(\mathcal{P}_{i-1} \cup \mathcal{Q}_i)|$ **then**

      $\lambda^{\text{GDY}}(\vec{R}) \leftarrow \lambda_1(\vec{R})$ for every $\vec{R} \in \mathcal{Q}_i$

    **else**

      $\lambda^{\text{GDY}}(\vec{R}) \leftarrow \lambda_2(\vec{R})$ for every $\vec{R} \in \mathcal{Q}_i$

    **end if**

  **else**

    **while** $\exists$ some unassigned $\vec{R} \in \mathcal{Q}_i$ **do**

      $\lambda^{\text{GDY}}(\vec{R}) \leftarrow \min\{l \in \mathbb{N} : \nexists\, \vec{S} \in \mathcal{P}_{i-1} \cup \mathcal{Q}_i$ such that $\vec{R}, \vec{S}$ collide and $\lambda^{\text{GDY}}(\vec{S}) = l\}$

    **end while**

  **end if**

  $\mathcal{P}_i \leftarrow \mathcal{P}_{i-1} \cup \mathcal{Q}_i$

**end for**

### 3.1.1 Edge Order

We traverse the edges of the host tree in a breadth-first manner, i.e., starting with an arbitrary vertex $r \in V_H$ as root, we perform a Breadth First Search (BFS) on the host tree $H$ and rank its edges in the order of their discovery, and then process the edges in this order. Let us assume that the set of edges $E_H$ in the order of enumeration is $\{e_1, \ldots, e_{|E_H|}\}$. Note that this edge ordering is not unique, but the wavelength assignment scheme relies only on the fact that the ordering is obtained via some BFS. In the $i$-th round of GREEDY-WA, edge $e_i$ is processed, i.e., wavelengths are assigned to all the unassigned rooted subtrees present on $e_i$. Clearly, the algorithm involves exactly $|E_H|$ rounds of wavelength assignment.[1]

### 3.1.2 Wavelength Assignment Strategy

We denote the set of rooted subtrees that are assigned wavelengths in the first $i$ rounds in GREEDY-WA by $\mathcal{P}_i$. We define $\mathcal{P}_0 := \emptyset$. The set of rooted subtrees present on edge $e_i$ but not in the set $\mathcal{P}_i$ is denoted by $\mathcal{Q}_i$, i.e., $\mathcal{Q}_i := \mathcal{R}[e_i] \setminus \mathcal{P}_i$. Note that $\mathcal{Q}_i$ is the set of rooted subtrees that are assigned wavelengths in the $i$-th round of GREEDY-WA.

The basic idea is to be greedy in each round of wavelength assignment in the sense that we try to use as few new wavelengths as possible while processing each host tree edge, i.e., in $i$-th round we try to assign wavelengths to the rooted subtrees in the set $\mathcal{Q}_i$ using as few new wavelengths as possible. Note that the algorithm is constructive in the sense that once a wavelength has been assigned to any rooted subtree, it is never changed.

The actual wavelength assignment scheme followed in the $i$-th round of GREEDY-WA depends on the type of edge $e_i$ being processed. According to Lemma 3.3 below, tree edge $e_i$ encountered during the $i$-th round of GREEDY-WA can be classified

[1] It may happen that in some rounds no rooted subtrees are assigned wavelengths.

into one of the four types (defined in the lemma) based on the status (whether already processed or not) of its adjacent tree edges. If edge $e_i$ is of type (i), (ii) or (iii) as defined in Lemma 3.3, then unassigned rooted subtrees are randomly selected from the set $\mathcal{Q}_i$ one at a time and are assigned wavelengths greedily. In more detail, suppose rooted subtree $\vec{R}$ has been selected from the set $\mathcal{Q}_i$ for wavelength assignment. If there is a wavelength that has already been assigned to some rooted subtree(s) and can also be assigned to $\vec{R}$, then that wavelength is assigned to $\vec{R}$, otherwise a new wavelength (not assigned to any other rooted subtree previously) is assigned to $\vec{R}$. In case there are several such used wavelengths, any one of them can be assigned to $\vec{R}$, e.g., according to line 17 of GREEDY-WA. On the other hand, if edge $e_i$ is of type (iv) as defined in Lemma 3.3, then we assign wavelengths to the rooted subtrees in the set $\mathcal{Q}_i$ according to the better of the two different wavelength assignment schemes presented as Subroutine 2 (PROCESS-EDGE-1) and Subroutine 3 (PROCESS-EDGE-2).

As we shall see in Lemma 3.3, edge $e_i = \{u, v\}$ being a type (iv) edge means that none of the tree edges adjacent to vertex $v$ have yet been processed and there are two edges adjacent to vertex $u$ (besides edge $e_i = \{u, v\}$), namely $\{u, w\}$ and $\{u, x\}$, of which edge $\{u, w\}$ has already been processed and edge $\{u, x\}$ has not yet been processed. The two schemes employed for assigning wavelengths while processing a type (iv) edge $e_i = \{u, v\}$ differ in the way they go about reusing the wavelengths. In PROCESS-EDGE-1 we prefer to reuse wavelengths from the set $\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u, v\}])$ (set of wavelengths assigned to the rooted subtree(s) present on host tree edge $e_i = \{u, v\}$ that were assigned wavelengths in the first $i - 1$ rounds), whereas in PROCESS-EDGE-2 we prefer to reuse wavelengths from the set $\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u, x\}] \backslash \mathcal{P}_{i-1}[\{u, v\}])$ (set of wavelengths assigned to the rooted subtree(s) present on host tree edge $\{u, x\}$, but not on tree edge $e_i = \{u, v\}$, that were assigned wavelengths in the first $i - 1$ rounds). Note that the two sets of wavelengths are not

**Subroutine 2** PROCESS-EDGE-1

**Require:** $\{\vec{T}_H, \{u,v\} \in E_H, \mathcal{P}, \mathcal{Q}, \lambda\}$ such that the degree of the bidirected tree $\vec{T}_H$ is at most 3, $\mathcal{P}$ is the set of rooted subtrees of $\vec{T}_H$ that have already been assigned wavelengths according to the mapping $\lambda : \mathcal{P} \longrightarrow \mathbb{N}$ and $\mathcal{Q}$ is the set of all the unassigned rooted subtrees of $\vec{T}_H$ that are present on host tree edge $\{u,v\}$.

**Ensure:** Complete the given mapping $\lambda$ to $\lambda : \mathcal{P} \cup \mathcal{Q} \longrightarrow \mathbb{N}$ such that $\lambda \in \Lambda_{\{\vec{T}_H, \mathcal{P}\cup\mathcal{Q}\}}$.

$B_1 \leftarrow G_{\mathcal{P}[\{u,v\}]\cup\mathcal{Q}}$
**for all** pairs $\vec{R}, \vec{S} \in \mathcal{P}[\{u,v\}] \cup \mathcal{Q}$ such that $\vec{R}, \vec{S}$ do not collide **do**
    **if** any one of the following is true:

    (i) $\vec{R}, \vec{S} \in \mathcal{P}$ and $\lambda(\vec{R}) \neq \lambda(\vec{S})$.

    (ii) $\vec{R} \in \mathcal{Q}, \vec{S} \in \mathcal{P}$ and $\exists \vec{U} \in \mathcal{P}$ such that $\lambda(\vec{S}) = \lambda(\vec{U})$ and $\vec{R}, \vec{U}$ collide.

    **then**
        $E_{B_1} \leftarrow E_{B_1} \cup \{\{\vec{R}, \vec{S}\}\}$
    **end if**
**end for**
Determine a maximum matching $M_{\bar{B}_1} \subseteq E_{\bar{B}_1}$. $\{\bar{B}_1$ is bipartite.$\}$
**for all** matched edges $\{\vec{R}, \vec{S}\} \in M_{\bar{B}_1}$ such that $\vec{R} \in \mathcal{Q}$ and $\vec{S} \in \mathcal{P}$ **do**
    $\lambda(\vec{R}) \leftarrow \lambda(\vec{S})$
**end for**
**while** $\exists$ some unassigned $\vec{R} \in \mathcal{Q}$ **do**
    **if** $\exists$ matched edge $\{\vec{R}, \vec{S}\} \in M_{\bar{B}_1}$ **then**
        $\lambda(\vec{R}),\lambda(\vec{S}) \leftarrow \min\{m \in \mathbb{N} : \nexists \vec{U} \in \mathcal{P}\cup\mathcal{Q}$ such that $\vec{R},\vec{U}$ or $\vec{S},\vec{U}$ collide and $\lambda(\vec{U}) = m\}$
    **else**
        $\lambda(\vec{R}) \leftarrow \min\{m \in \mathbb{N} : \nexists \vec{U} \in \mathcal{P} \cup \mathcal{Q}$ such that $\vec{R}, \vec{U}$ collide and $\lambda(\vec{U}) = m\}$
    **end if**
**end while**

**Subroutine 3** PROCESS-EDGE-2

**Require:** $\{\vec{T}_H, \{\{u,v\},\{u,w\},\{u,x\}\} \subseteq E_H, \mathcal{P}, \mathcal{Q}, \lambda\}$ such that the degree of the bidirected tree $\vec{T}_H$ is 3, $\mathcal{P}$ is the set of rooted subtrees of $\vec{T}_H$ that have already been assigned wavelengths according to the mapping $\lambda : \mathcal{P} \longrightarrow \mathbb{N}$ and $\mathcal{Q}$ is the set of all the unassigned rooted subtrees of $\vec{T}_H$ that are present on host tree edge $\{v,u\}$.

**Ensure:** Complete the given mapping $\lambda$ to $\lambda : \mathcal{P} \cup \mathcal{Q} \longrightarrow \mathbb{N}$ such that $\lambda \in \Lambda_{\vec{T}_H, \mathcal{P}\cup\mathcal{Q}}$.

$B_2 \leftarrow G_{(\mathcal{P}[\{u,x\}]\setminus\mathcal{P}[\{u,v\}])\cup\mathcal{Q}[\{u,x\}]}$
**for all** pairs $\vec{R},\vec{S} \in (\mathcal{P}[\{u,x\}]\setminus\mathcal{P}[\{u,v\}])\cup\mathcal{Q}[\{u,x\}]$ such that $\vec{R},\vec{S}$ do not collide **do**
  **if** any one of the following is true:
   (i) $\vec{R},\vec{S} \in \mathcal{P}$ and $\lambda(\vec{R}) \neq \lambda(\vec{S})$.
   (ii) $\vec{R} \in \mathcal{Q}, \vec{S} \in \mathcal{P}$ and $\exists \vec{U} \in \mathcal{P}$ such that $\lambda(\vec{S}) = \lambda(\vec{U})$ and $\vec{R},\vec{U}$ collide.
  **then**
    $E_{B_2} \leftarrow E_{B_2} \cup \{\{\vec{R},\vec{S}\}\}$
  **end if**
**end for**
Determine a maximum matching $M_{\bar{B}_2} \subseteq E_{\bar{B}_2}$. $\{\bar{B}_2$ is bipartite.$\}$
**for all** matched edges $\{\vec{R},\vec{S}\} \in M_{\bar{B}_2}$ such that $\vec{R} \in \mathcal{Q}$ and $\vec{S} \in \mathcal{P}$ **do**
  $\lambda(\vec{R}) \leftarrow \lambda(\vec{S})$
**end for**
**while** $\exists$ some unassigned $\vec{R} \in \mathcal{Q}[\{u,x\}]$ **do**
  **if** $\exists$ matched edge $\{\vec{R},\vec{S}\} \in M_{\bar{B}_2}$ **then**
    $\lambda(\vec{R}),\lambda(\vec{S}) \leftarrow \min\{m \in \mathbb{N} : \nexists \vec{U} \in \mathcal{P}\cup\mathcal{Q}$ such that $\vec{R},\vec{U}$ or $\vec{S},\vec{U}$ collide and $\lambda(\vec{U}) = m\}$
  **else**
    $\lambda(\vec{R}) \leftarrow \min\{m \in \mathbb{N} : \nexists \vec{U} \in \mathcal{P}\cup\mathcal{Q}$ such that $\vec{R},\vec{U}$ collide and $\lambda(\vec{U}) = m\}$
  **end if**
**end while**
**while** $\exists$ some unassigned $\vec{R} \in \mathcal{Q}$ **do**
  $\lambda(\vec{R}) \leftarrow \min\{m \in \mathbb{N} : \nexists \vec{U} \in \mathcal{P}\cup\mathcal{Q}$ such that $\vec{R},\vec{U}$ collide and $\lambda(\vec{U}) = m\}$
**end while**

necessarily mutually exclusive. The two schemes also differ in the order in which unassigned rooted subtrees in the set $\mathcal{Q}_i$ are selected for wavelength assignment. More specifically, in PROCESS-EDGE-2, first wavelengths are assigned to all the rooted subtrees in the set $\mathcal{Q}_i[\{u, x\}]$ and then to the rest of the unassigned rooted subtrees.

In PROCESS-EDGE-1 (line 7), we determine the maximum number of mutually exclusive pairs of rooted subtrees such that in each matched pair (say $\vec{R}, \vec{S}$) at least one of the rooted subtrees (say $\vec{R}$) is an unassigned rooted subtree from the set $\mathcal{Q}_i$ (i.e., $\vec{R} \in \mathcal{Q}_i$) and the second rooted subtree ($\vec{S}$ in this case) may either be (i) another unassigned rooted subtree from the set $\mathcal{Q}_i$ (i.e., $\vec{S} \in \mathcal{Q}_i$) or (ii) a rooted subtree from the set $\mathcal{P}_{i-1}[e_i]$ such that the unassigned rooted subtree in the pair can be safely assigned its wavelength (i.e., $\vec{S} \in \mathcal{P}_{i-1}$ such that $\vec{R}$ does not collide with any rooted subtree that has already been assigned the same wavelength as $\vec{S}$). If the pair is of type (ii), then the unassigned rooted subtree is assigned the same wavelength as the other rooted subtree (line 9). If the pair is of type (i), then both the rooted subtrees of the pair are assigned the same wavelength (line 13). In this case, preference is given to the wavelengths that have already been assigned to some rooted subtree(s). If there is no such suitable wavelength, a new wavelength is used.

In PROCESS-EDGE-2 (line 7), we determine the maximum number of mutually exclusive pairs of rooted subtrees such that in each matched pair (say $\vec{R}, \vec{S}$) at least one of the rooted subtrees (say $\vec{R}$) is an unassigned rooted subtree from the set $\mathcal{Q}_i$ and is present on tree edge $\{u, x\}$ (i.e., $\vec{R} \in \mathcal{Q}_i[\{u, x\}]$) and the second rooted subtree ($\vec{S}$ in this case) may either be (i) another unassigned rooted subtree from the set $\mathcal{Q}_i$ present on edge $\{u, x\}$ (i.e., $\vec{S} \in \mathcal{Q}_i[\{u, x\}]$) or (ii) a rooted subtree from the set $\mathcal{P}_{i-1}[\{u, x\}] \setminus \mathcal{P}_{i-1}[\{u, v\}]$ such that the unassigned rooted subtree in the pair can be safely assigned its wavelength (i.e., $\vec{S} \in \mathcal{P}_{i-1}[\{u, x\}] \setminus \mathcal{P}_{i-1}[\{u, v\}]$ such that $\vec{R}$ does not collide with any rooted subtree that has already been assigned the same wavelength as $\vec{S}$). If the pair is of type (ii), then the unassigned rooted subtree is assigned the same wavelength as the other rooted subtree (line 9). If the pair is of type (i), then both the rooted subtrees of the pair are assigned the same wavelength (line 13). Again preference is given to the wavelengths that have already been assigned to some rooted subtree(s). If there is no such suitable wavelength, a new wavelength is used. After this all the remaining unassigned rooted subtrees (all the rooted subtree in the set $\mathcal{Q}_i \setminus \mathcal{Q}_i[\{u, x\}]$ and possibly some rooted subtrees still unassigned in the set $\mathcal{Q}_i[\{u, x\}]$) are assigned wavelengths one at a time (lines 15, 19). Again preference is given to the wavelengths that have already been assigned to some rooted subtree(s).

The exact steps of PROCESS-EDGE-1 and PROCESS-EDGE-2 are explained in detail in Lemmas 3.7 and 3.8, respectively.

## 3.2 Approximation Analysis

In this section, we shall prove that the number of wavelengths required by GREEDY-WA is within $\frac{5}{2}$ times the minimum number of wavelengths required to support the given set of rooted subtrees $\mathcal{R}$ on the given bidirected tree $\vec{T}_H$, i.e., we shall prove that

$$\frac{|\lambda^{\mathrm{GDY}}(\mathcal{R})|}{\min_{\lambda \in \Lambda_{\{\vec{T}_H, \mathcal{R}\}}} |\lambda(\mathcal{R})|} = \frac{|\lambda^{\mathrm{GDY}}(\mathcal{R})|}{\chi_{G_{\mathcal{R}}}} \leq \frac{5}{2}.$$

### 3.2.1 Some Local Properties

We start off by proving the following pair of useful results about the local structure of the problem at hand.

(i) In Lemma 3.1, we characterize the conflict graph of the rooted subtrees present on a single host tree edge as the complement of a bipartite graph [41, p.6]. This is because the rooted subtrees on a single host tree edge are actually the rooted subtrees present on the corresponding pair of anti-parallel directed edges of the bidirected tree. Therefore, they can be partitioned into two subsets based on the directed edge of the corresponding pair of anti-parallel directed edges on which they are present, and the conflict graph of each of these sets is a clique [41, p.112]. This result is important since most of the graphs that we encounter during the analysis of GREEDY-WA are of this type and therefore have nice properties (coloring etc.).

(ii) Lemma 3.2 allows us to study only those instances $\{\vec{T}_H, \mathcal{R}\}$ of the MIN-MC-WA-BT problem where the given set of rooted subtrees $\mathcal{R}$ is such that the number of rooted subtrees present on any directed edge of the given bidirected tree $\vec{T}_H$ is the same, i.e, for any pair of directed edges $(u, v), (w, x) \in E_{\vec{T}_H}$, $|\mathcal{R}[(u, v)]| = |\mathcal{R}[(w, x)]|$.

**Lemma 3.1.** *The complement of the conflict graph of any subset of rooted subtrees present on a single host tree edge is bipartite.*

*Proof.* Let $\mathcal{S} \subseteq \mathcal{R}[\{u, v\}]$, i.e., $\mathcal{S}$ is a subset of rooted subtrees present on host tree edge $\{u, v\} \in E_H$. We have to show that $\bar{G}_{\mathcal{S}}$, the complement of the conflict graph of rooted subtrees in the set $\mathcal{S}$, is bipartite. Observe that $\mathcal{S}$ can be partitioned into $\mathcal{S}[(u, v)]$ and $\mathcal{S}[(v, u)]$. Since all the rooted subtrees in partition $\mathcal{S}[(u, v)]$ collide on the directed edge $(u, v)$, there is no edge $\{\vec{R}_i, \vec{R}_j\}$ in $E_{\bar{G}_{\mathcal{S}}}$ such that the rooted subtrees $\vec{R}_i, \vec{R}_j \in \mathcal{S}[(u, v)]$. Hence, $\mathcal{S}[(u, v)]$ is an independent set in $\bar{G}_{\mathcal{S}}$. By similar reasoning, $\mathcal{S}[(v, u)]$ is also an independent set in $\bar{G}_{\mathcal{S}}$. This implies that $\bar{G}_{\mathcal{S}}$ is bipartite. □

The *load* of a set $\mathcal{R}$ of rooted subtrees on a bidirected tree $\vec{T}_H$ is defined to be $l_{\{\vec{T}_H, \mathcal{R}\}} := \max_{(u,v) \in E_{\vec{T}_H}} |\mathcal{R}[(u, v)]|$.

**Lemma 3.2.** *If the load of the set $\mathcal{R}$ of rooted subtrees on the bidirected tree $\vec{T}_H$ is $l_{\{\vec{T}_H, \mathcal{R}\}}$ and the chromatic number of the corresponding conflict graph is $\chi_{G_{\mathcal{R}}}$, then* *there exists a set $\mathcal{S} \supseteq \mathcal{R}$ of rooted subtrees on the bidirected tree $\vec{T}_H$ such that the following hold:*

*(i) The chromatic number of the new conflict graph is the same as that of the original conflict graph, i.e., $\chi_{G_{\mathcal{S}}} = \chi_{G_{\mathcal{R}}}$.*

*(ii) For every directed edge $(u, v) \in E_{\vec{T}_H}$, $|\mathcal{S}[(u, v)]| = l_{\{\vec{T}_H, \mathcal{S}\}} = l_{\{\vec{T}_H, \mathcal{R}\}}$.*

*Moreover, $\mathcal{S}$ can be constructed in polynomial time.*

*Proof.* Corresponding to the bidirected tree $\vec{T}_H$ and the set $\mathcal{R}$ of rooted subtrees on $\vec{T}_H$, we generate a set $\mathcal{S} \supseteq \mathcal{R}$ of rooted subtrees on $\vec{T}_H$ via Algorithm 4 (ADD-DUMMY-RS). Condition (ii) of the lemma is satisfied by construction of the set $\mathcal{S}$ in ADD-DUMMY-RS.

**Algorithm 4** ADD-DUMMY-RS

**Require:** MIN-MC-WA-BT problem instance $\{\vec{T}_H, \mathcal{R}\}$.

**Ensure:** MIN-MC-WA-BT problem instance $\{\vec{T}_H, \mathcal{S}\}$, where $\mathcal{S} \supseteq \mathcal{R}$ and for every directed edge $(u, v) \in E_{\vec{T}_H}$, $|\mathcal{S}[(u, v)]| = l^{\{\vec{T}_H, \mathcal{R}\}}$.

```
S ← R
for all edges (u, v) ∈ E_{T_H} do
    while |S[(u, v)]| < l^{T_H, R} do
        Let directed graph D be such that V_D = {u, v} and E_D = {(u, v)}.
        {D is a rooted subtree of bidirected tree T_H, having vertex u ∈ V_{T_H} as the root.}
        S ← S ∪ {D}
    end while
end for
```

Let $\psi^* \in \Psi_{G_{\mathcal{R}}}$ be a minimum vertex coloring of the conflict graph $G_{\mathcal{R}}$. Consider the vertex coloring $\psi \in \Psi_{G_{\mathcal{S}}}$ of the conflict graph $G_{\mathcal{S}}$ such that for each rooted subtree $\vec{R} \in \mathcal{R} \subseteq \mathcal{S}$, $\psi(\vec{R}) = \psi^*(\vec{R})$. For any host tree edge $\{u, v\} \in E_H$, the set of subtrees added by ADD-DUMMY-RS that are rooted at vertex $u$ is $\mathcal{S}[(u, v)] \setminus \mathcal{R}[(u, v)]$ and the set of subtrees added by ADD-DUMMY-RS that are rooted at vertex $v$ is $\mathcal{S}[(v,u)] \setminus \mathcal{R}[(v,u)]$. Note that $|\mathcal{S}[(u,v)] \setminus \mathcal{R}[(u,v)]| = l_{\{\vec{T}_H,\mathcal{R}\}} - |\mathcal{R}[(u,v)]|$ and $|\mathcal{S}[(v,u)] \setminus \mathcal{R}[(v,u)]| = l_{\{\vec{T}_H,\mathcal{R}\}} - |\mathcal{R}[(v,u)]|$. The number of colors used by all the rooted subtrees in the set $\mathcal{R}[\{u,v\}]$ in coloring $\psi$ is $|\psi(\mathcal{R}[\{u,v\}])| = |\psi^*(\mathcal{R}[\{u,v\}])|$. According to Lemma 3.1, $\bar{G}_{\mathcal{R}[\{u,v\}]}$ is bipartite. Therefore, in any vertex coloring of graph $G_{\mathcal{R}[\{u,v\}]}$, a rooted subtree can share its color with at most one other rooted subtree. Consequently, the number of rooted subtrees in the set $\mathcal{R}[\{u,v\}]$ that do not share their assigned colors with any other rooted subtree in the set $\mathcal{R}[\{u,v\}]$ is $2|\psi(\mathcal{R}[\{u,v\}])| - |\mathcal{R}[\{u,v\}]|$. Observe that a rooted subtree in the set $\mathcal{S}[(u,v)] \setminus \mathcal{R}[(u,v)]$ collides with every other rooted subtree in the set $\mathcal{S}[(u,v)]$ and does not collide with any other rooted subtree in the set $\mathcal{S}$. Similarly, a rooted subtree in the set $\mathcal{S}[(v,u)] \setminus \mathcal{R}[(v,u)]$ collides with every other rooted subtree in the set $\mathcal{S}[(v,u)]$ and does not collide with any other rooted subtree in the set $\mathcal{S}$. Therefore, we can color $\min\{2|\psi(\mathcal{R}[\{u,v\}])|, |\mathcal{S}[\{u,v\}]|\} - |\mathcal{R}[\{u,v\}]|$ rooted subtrees in the set $\mathcal{S}[\{u,v\}] \setminus \mathcal{R}[\{u,v\}]$ using the colors already assigned to some other rooted subtree in the set $\mathcal{R}[\{u,v\}]$. Hence, the number of remaining uncolored rooted subtrees in the set $\mathcal{S}[\{u,v\}] \setminus \mathcal{R}[\{u,v\}]$ is

$$\begin{aligned} &|\mathcal{S}[\{u,v\}] \setminus \mathcal{R}[\{u,v\}]| - (\min\{2|\psi(\mathcal{R}[\{u,v\}])|, |\mathcal{S}[\{u,v\}]|\} - |\mathcal{R}[\{u,v\}]|) \\ &\qquad = [|\mathcal{S}[\{u,v\}]| - 2|\psi(\mathcal{R}[\{u,v\}])|]^+ = 2\left[l_{\{\vec{T}_H,\mathcal{R}\}} - |\psi(\mathcal{R}[\{u,v\}])|\right]^+. \end{aligned}$$

Note that half of the remaining uncolored rooted subtrees are in the set $\mathcal{S}[(u,v)] \setminus \mathcal{R}[(u,v)]$ and the other half are in the set $\mathcal{S}[(v,u)] \setminus \mathcal{R}[(v,u)]$. Employing these insights for generating the coloring $\psi$, we need $\left[l_{\{\vec{T}_H,\mathcal{R}\}} - |\psi(\mathcal{R}[\{u,v\}])|\right]^+$ additional colors that have not been assigned to any rooted subtree in the set $\mathcal{R}[\{u,v\}]$ in order to color all the rooted subtrees in the set $\mathcal{S}[\{u,v\}]$. Thus, the total number of colors required by the mapping $\psi$ for coloring all the rooted subtrees in the set $\mathcal{S}[\{u,v\}]$ is

$$|\psi(\mathcal{R}[\{u,v\}])| + \left[l_{\{\vec{T}_H,\mathcal{R}\}} - |\psi(\mathcal{R}[\{u,v\}])|\right]^+ = \max\left\{l_{\{\vec{T}_H,\mathcal{R}\}}, |\psi(\mathcal{R}[\{u,v\}])|\right\}.$$

Therefore,

$$\begin{aligned}
\chi_{G_{\mathcal{S}}} &\leq |\psi(\mathcal{S})| = \max_{\{u,v\}\in E_H} \max\left\{ l_{\{\vec{T}_H,\mathcal{R}\}}, |\psi(\mathcal{R}[\{u,v\}])| \right\} \\
&= \max_{\{u,v\}\in E_H} |\psi(\mathcal{R}[\{u,v\}])| \leq \chi_{G_{\mathcal{R}}},
\end{aligned}$$

where the last equality is due to the fact that

$$\max_{\{u,v\}\in E_H} |\psi(\mathcal{R}[\{u,v\}])| \geq l_{\{\vec{T}_H,\mathcal{R}\}}.$$

Also since conflict graph $G_{\mathcal{R}}$ is a subgraph of the conflict graph $G_{\mathcal{S}}$, $\chi_{G_{\mathcal{R}}} \leq \chi_{G_{\mathcal{S}}}$. This gives us $\chi_{G_{\mathcal{R}}} = \chi_{G_{\mathcal{S}}}$, which proves condition (i) of the lemma. □

Recall that any given instance $\{\vec{T}_H, \mathcal{R}\}$ of the MIN-MC-WA-BT problem is equivalent to the minimum vertex coloring problem for the conflict graph $G_{\mathcal{R}}$. Using this equivalence along with Lemma 3.2 allows us to assume, without loss of generality, that the given instance $\{\vec{T}_H, \mathcal{R}\}$ of the MIN-MC-WA-BT problem is such that for every directed edge $(u,v) \in E_{\vec{T}_H}$, $|\mathcal{R}[(u,v)]| = l_{\{\vec{T}_H,\mathcal{R}\}}$. As a consequence, for any host tree $\{u,v\} \in E_H$, $|\mathcal{R}[\{u,v\}]| = 2l_{\{\vec{T}_H,\mathcal{R}\}}$. From now onwards we shall make the said assumption.

### 3.2.2 Roadmap

Next, we give a brief plan-of-action that we shall follow for the rest of this section for proving the approximation ratio of $\frac{5}{2}$ for GREEDY-WA. The analysis proceeds according to the following steps.

(i) First we characterize the types of host tree edges that we might encounter during any round of wavelength assignment in GREEDY-WA. This is done in Lemmas 3.3 and 3.4.

(ii) Next we prove that if the edge to be processed in $i$-th round of wavelength assignment is of type (i), (ii) or (iii) as defined in Lemma 3.3, then either no new wavelengths are required in the $i$-th round or the total number of wavelengths in use at the end of the $i$-th round is less than or equal to $2l_{\{\vec{T}_H,\mathcal{R}\}}$. This is proved in Lemma 3.5.

(iii) We prove a similar result for the case when the edge to be processed in the $i$-th round of wavelength assignment is of type (iv) as defined in Lemma 3.3. In this case we first show that either no new wavelength is required in the $i$-th round or $\lambda^{\text{GDY}}(\mathcal{Q}_i \cup \mathcal{P}_{i-1}[\{u,w\}]) = \lambda^{\text{GDY}}(\mathcal{P}_i)$. The set $\mathcal{Q}_i \cup \mathcal{P}_{i-1}[\{u,w\}]$ consists of all the rooted subtrees that are assigned wavelengths in the $i$-th round ($\mathcal{Q}_i$) and all the rooted subtrees that are present on host tree edge $\{u,w\}$ which is adjacent to the edge being processed in the $i$-th round and has already been processed ($\mathcal{P}_{i-1}[\{u,w\}]$). This is shown in Lemma 3.6. Next, we present bounds on the number of wavelengths required after the $i$-th round for assigning wavelengths to all the rooted subtrees in the set $\mathcal{Q}_i \cup \mathcal{P}_{i-1}[\{u,w\}]$ by subroutines PROCESS-EDGE-1 (Lemma 3.7) and PROCESS-EDGE-2 (Lemma 3.8). Note that in GREEDY-WA (line 10), of the two wavelength assignments generated by PROCESS-EDGE-1 and PROCESS-EDGE-2, the assignment requiring fewer wavelengths at the end of the $i$-th round is used.

(iv) Based on the previous lemmas, we determine the approximation ratio of GREEDY-WA in a parameterized form in Lemma 3.9. In Lemma 3.10, we determine the worst case (maximum) value of the parameterized fraction obtained in Lemma 3.9. This proves Theorem 3.11 that the approximation ratio of GREEDY-WA is $\frac{5}{2}$.

### 3.2.3 Host Tree Edge Types

Next, we start the actual analysis of our greedy wavelength assignment scheme. First, we note that as GREEDY-WA proceeds, the host tree edge that is processed in

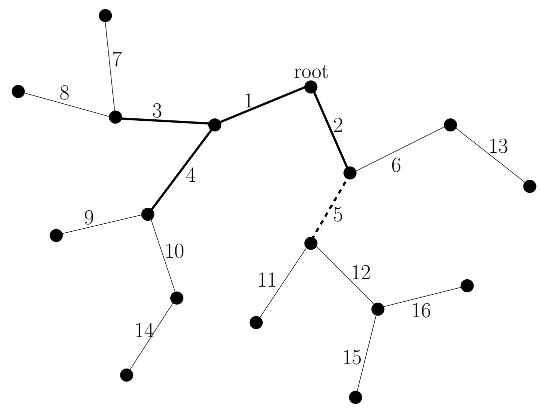



Figure 3.1: Status of host tree edges during the 5-th round of GREEDY-WA.

any round of wavelength assignment is from one of the four possible types defined in Lemma 3.3. The edge type is characterized by the status (whether already processed or not) of its adjacent edges. The scheme employed for assigning wavelengths to the unassigned rooted subtrees present on the edge being processed depends on the type of the edge. In Lemma 3.4, we characterize the set of rooted subtrees that have already been assigned wavelengths and can collide with the rooted subtrees that are being assigned wavelengths in the next round of GREEDY-WA.

Both these results mainly rely on the BFS ordering of the edges in GREEDY-WA and the fact that the bidirected tree $\vec{T}_H$ has degree $\Delta_{\vec{T}_H} \leq 3$ .

**Lemma 3.3.** *In GREEDY-WA, when a host tree edge* $\{u, v\} \in E_H$ *(where* $u$ *was discovered before* $v$ *in the BFS) is being processed, then all the edges adjacent to vertex* $v$ *are unprocessed, and for the edges adjacent to vertex* $u$ *exactly one of the following is satisfied:*

*(i) None of the edges adjacent to* $u$ *has been processed. In this case edge* $\{u, v\}$ *is the first edge to be processed among all host tree edges.*

*(ii) Host tree vertex* $u$ *has degree* $\delta_H(u) = 2$ *with adjacent edges* $\{u, v\}, \{u, w\}$ *of which edge* $\{u, w\}$ *has already been processed.*

*(iii) Host tree vertex $u$ has degree $\delta_H(u) = 3$ with adjacent edges $\{u, v\}, \{u, w\}, \{u, x\}$ of which edges $\{u, w\}, \{u, x\}$ have already been processed.*

*(iv) Host tree vertex $u$ has degree $\delta_H(u) = 3$ with adjacent edges $\{u, v\}, \{u, w\}, \{u, x\}$ of which edge $\{u, w\}$ has already been processed while edge $\{u, x\}$ has not yet been processed.*

*Proof.* To motivate the intuition behind this lemma, observe Figure 3.1. Consider the host tree and the BFS ordering of its edges as shown in the figure. In this case edge 1 is of type (i), edge 2 is of type (ii), edge 4 is of type (iii) and edge 3 is of type (iv). Similarly, note that all the host tree edges can be classified as being of one of the four types described in the lemma. Next, we present the actual proof.

GREEDY-WA selects an arbitrary host tree vertex $r \in V_H$ and ranks the edges of the host tree according to their order of discovery in a BFS with $r$ as the root. The edges are then processed according to this ordering. We denote the set of host tree edges that are processed in the first $i$ rounds of wavelength assignment by $E_H^{(i)}$. According to the notation defined in Section 3.1.1, $E_H^{(i)} := \{e_1, \ldots, e_i\}$. Observe that due to the BFS ordering, $H[E_H^{(i)}]$ is a connected subgraph of $H$. Moreover, since $H$ is a tree, $H[E_H^{(i)}]$ must be its subtree. Also note that the root of the BFS lies in this subtree, i.e., $r \in V_{H[E_H^{(i)}]}$ for every $i > 0$. This is because $r$ has to be an end vertex of $e_1$, the first processed edge.

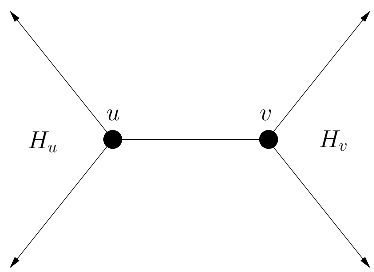
Let $e_k = \{u, v\} \in E_H$ be the host tree edge being processed in the $k$-th round of wavelength assignment. Observe that $H[E_H \setminus \{\{u, v\}\}]$, the subgraph of the host tree induced by all the edges of the host tree except edge $\{u, v\}$ is a forest [41, p.6] containing two trees. Let us denote the two trees as $H_u$ and $H_v$ such that $u \in V_{H_u}$ and $v \in V_{H_v}$. This is shown in Figure 3.2(a). Since the vertex $u$ was discovered before the vertex $v$ in the BFS, the path from root $r$ to $v$ should contain the edge $\{u, v\}$. This observation, along with the fact that $H$ is a tree, implies that $r \in V_{H_u}$. Hence, every edge in the set $E_{H_v}$ must have been discovered after the discovery of the edge $e_k = \{u, v\}$. Consequently, none of the edges in the set $E_{H_v}$ were processed in the first $k$ rounds of wavelength assignment. Since every edge adjacent to the vertex $v$ is in the set $E_{H_v} \cup \{\{u, v\}\}$, it must be unprocessed at the end of $k-1$ rounds of wavelength assignment.

Consider the edges adjacent to the vertex $u$. If none of the edges adjacent to $u$ are processed in the first $k-1$ rounds, then we claim that $k$ is equal to 1. This is because $H[E_H^{(k)}]$ is a tree (therefore connected) and none of the edges adjacent to $v$ were assigned wavelengths in the first $k-1$ rounds. Thus, the only scenario when $H[E_H^{(k)}]$ is connected is when $E_H^{(k-1)} = \emptyset$, which implies that $e_k = \{u, v\}$ is indeed the first edge being processed. This corresponds to case (i) of the lemma. Next, we consider the alternative scenario when there is an edge $\{u, w\} \in E_H^{(k-1)}$, i.e., there is an edge $\{u, w\}$ adjacent to the vertex $u$ that has already been processed in the first $k-1$ rounds. If the vertices $v$ and $w$ are the only neighbors of $u$, then $\delta_H(u) = 2$ and this corresponds to case (ii) of the lemma. On the other hand if $\delta_H(u) = 3$ then let the vertices $w$, $v$ and $x$ be the three neighbors of $u$ in the host tree. As previously discussed, the edge $\{u, w\}$ has already been processed in the first $k-1$ rounds and the edge $\{u, v\}$ is the current edge being processed in the $k$-th round. Depending on whether the edge $\{u, x\}$ has already been processed in the first $k-1$ rounds or not, we obtain cases (iv) and (iii) respectively of the lemma.

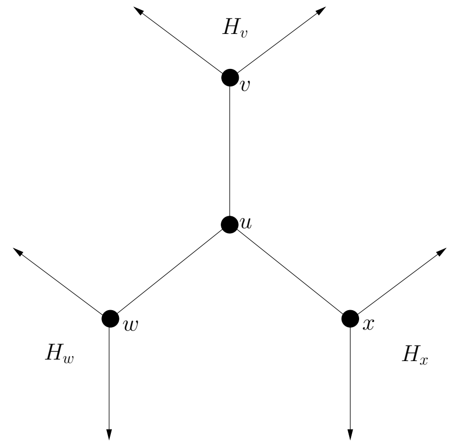
Since $\delta_H(u) \leq \Delta_H \leq 3$, there are no other possible cases. $\square$

**Lemma 3.4.** *In the $i$-th round of wavelength assignment in GREEDY-WA (while processing host tree edge $e_i = \{u, v\} \in E_H$), if a rooted subtree $\vec{P} \in \mathcal{P}_{i-1}$, that has already been assigned a wavelength, collides with any unassigned rooted subtree $\vec{Q} \in \mathcal{Q}_i$, then exactly one of the following is satisfied:*

*(i) Edge $e_i = \{u, v\}$ is of type (i), (ii) or (iii) defined in Lemma 3.3, and rooted subtree $\vec{P} \in \mathcal{P}_{i-1}[\{u, v\}]$.*




(a) Subgraph $H[E_H \setminus \{\{u,v\}\}]$ is a forest containing trees $H_u$ and $H_v$.




(b) Subgraph $H[V_H \setminus \{u\}]$ is a forest containing $\delta_H(u)$ trees. If $\delta_T(u) = 3$ and $w, v, x$ are the neighbors of $u$, the forest contains trees $H_w$, $H_v$, and $H_x$.

Figure 3.2: Graphs obtained by removing an edge ($\{u,v\}$) or a vertex ($u$) from the host tree $H$ are forests.

*(ii) Edge $e_i = \{u,v\}$ is of type (iv) defined in Lemma 3.3, and rooted subtree $\vec{P} \in \mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}]$ (where the vertex $x$ and the edge $\{u,x\}$ are as defined in Lemma 3.3).*

*Proof.* If the edge $e_i = \{u,v\} \in E_H$ being processed is of type (i) defined in Lemma 3.3, then it is the first edge being processed. Therefore, there are no rooted subtrees that have already been assigned some wavelength before processing edge $e_i = \{u,v\}$, i.e., $\mathcal{P}_{i-1} = \emptyset$. Consequently, the set of rooted subtrees that collide with any of the rooted subtrees in the set $\mathcal{Q}_i$ and have already been assigned wavelength before the processing of edge $e_i = \{u,v\}$, which is a subset of the set $\mathcal{P}_{i-1}$, is also empty. This is exactly what the lemma states for edges of type (i).

Next, we assume that the edge $e_i = \{u,v\} \in E_H$ being processed is of type (ii). As observed during the proof of Lemma 3.3, $H[E_H \setminus \{\{u,v\}\}]$ is a forest containing two trees $H_u$ and $H_v$ where $u \in V_{H_u}$ and $v \in V_{H_v}$. In this case the following hold:

(i) No edges in the set $E_{H_v}$ are processed in the first $i$ rounds of wavelength assignment.

(ii) No rooted subtree in the set $\mathcal{Q}_i$ is present on any host tree edge in the set $E_{H_u}$, i.e., for every rooted subtree $\vec{R} \in \mathcal{Q}_i$, $E_{\|\vec{R}\|} \cap E_{H_u} = \emptyset$.

We have already shown (i) in the proof of Lemma 3.3 and the reasoning for (ii) is as follows. Let there be a rooted subtree $\vec{R} \in \mathcal{Q}_i$ and an edge $\{a, b\} \in E_{H_u}$ such that $\{a, b\} \in E_{\|\vec{R}\|}$. First, note that since $\vec{R} \in \mathcal{Q}_i$, $e_i = \{u, v\} \in E_{\|\vec{R}\|}$. Next, observe that in this case edge $\{u, w\}$ (the only other edge adjacent to $u$ except $\{u, v\}$) has already been processed, so $\{u, w\} \notin E_{|\vec{R}\|}$. Also note that $\{u, w\}$ is the only edge adjacent to $u$ in the set $E_{H_u}$. Therefore, the facts that $\|\vec{R}\|$ is a subtree of $H$ and $\vec{R}$ is present on edges $\{a, b\} \in E_{H_u}$ and $\{u, v\}$ imply that it must be present on edge $\{u, w\}$. This is a contradiction, which proves (ii). Coming back to the proof of the lemma, let rooted subtree $\vec{S} \in \mathcal{P}_{i-1}$ collide with some rooted subtree in the set $\mathcal{Q}_i$. Since $\vec{S}$ has been assigned some wavelength in the first $i-1$ rounds of wavelength assignment, it must be present on some already processed edge. Therefore, by (i) it must be present on some edge in the set $E_{H_u}$. Also, since it collides with some rooted subtree from the set $\mathcal{Q}_i$, due to (ii) it must be present on some edge in the set $E_{H_v} \cup \{\{u, v\}\}$. The above two observations, combined with the fact that $\|\vec{S}\|$ is a subtree of the host tree, prove that $\|\vec{S}\|$ is present on the edge $e_i = \{u, v\}$. This is exactly what the lemma states for edges of type (ii).

The case when the edge being processed in the $i$-th round of wavelength assignment is of type (iii) is exactly analogous to the above case and the proof follows the same lines.

Next we assume that the edge $e_i = \{u, v\} \in E_T$ being processed is of type (iv). Hence, $\{u, w\}$, $\{u, v\}$ and $\{u, x\}$ are the three edges adjacent to $u$, and in the first $i-1$ rounds of wavelength assignment, $\{u, w\}$ has already been processed whereas $\{u, x\}$ has not been processed. In this case observe that $H[V_H \setminus \{u\}]$, the subgraph of the host tree induced by all the vertices of the host tree except the vertex $u$, is a forest containing three trees. Let us denote the three trees as $H_w$, $H_v$ and $H_x$ such that $w \in V_{H_w}$, $v \in V_{H_v}$ and $x \in V_{H_x}$. This is shown in Figure 3.2(b). We claim that in this case, the following hold:

(i) No edges in the set $E_{H_v} \cup E_{H_x} \cup \{\{u, x\}\}$ are processed in the first $i$ rounds of wavelength assignment.

(ii) No rooted subtree in the set $\mathcal{Q}_i$ is present on any host tree edge in the set $E_{H_w} \cup \{\{u, w\}\}$.

Note that we have already shown in the proof of Lemma 3.3 that no edges in the set $E_{H_v}$ are processed in the first $i$ rounds of wavelength assignment. Also note that in this case we assume that $\{u, x\}$ is unprocessed in the first $i$ rounds of wavelength assignment. Suppose there is an edge $\{a, b\} \in E_{H_x}$ which is processed in the first $i$ rounds of wavelength assignment. Since $\{u, v\}$ is a type (iv) edge, the edge $\{u, w\}$ has already been processed in the first $i$ rounds of wavelength assignment. Also, we have shown in the proof of Lemma 3.3 that $H[E_H^{(i)}]$, the subgraph of host tree $H$ induced by the set $E_H^{(i)}$ of edges processed during the first $i$ rounds of wavelength assignment, is a subtree of the host tree. Thus, the fact that edges $\{a, b\} \in E_{H_x}$ and $\{u, w\}$ both lie in the set $E_H^{(i)}$ requires that the edge $\{u, x\}$ also lie in the set $E_H^{(i)}$. This is a contradiction. Therefore, no edges in the set $E_{H_x}$ are processed in the first $i$ rounds of wavelength assignment. This proves (i). The reasoning for (ii) is as follows. Since edge $\{u, v\}$ is of type (iv), edge $\{u, w\}$ has already been processed in the first $i - 1$ rounds of wavelength assignment. Therefore, any rooted subtree that is unassigned after the first $i-1$ rounds of wavelength assignment cannot be present on the edge $\{u, w\}$. Let there be a rooted subtree $\vec{R} \in \mathcal{Q}_i$ and an edge $\{a, b\} \in E_{H_w}$ such that $\{a, b\} \in E_{\|\vec{R}\|}$. First note that since $\vec{R} \in \mathcal{Q}_i$, $e_i = \{u, v\} \in E_{\|\vec{R}\|}$. The facts that $\|\vec{R}\|$ is a subtree of the host tree $H$, and $\vec{R}$ is present on edges $\{a, b\} \in E_{H_w}$

and $\{u, v\}$ imply that it must be present on the edge $\{u, w\}$. Since we have already shown that this is not possible, we have a contradiction. This proves (ii). Coming back to the proof of the lemma, let rooted subtree $\vec{S} \in \mathcal{P}_{i-1}$ collide with some rooted subtree in the set $\mathcal{Q}_i$. Since $\vec{S}$ has been assigned some wavelength in the first $i-1$ rounds of wavelength assignment, it must be present on some already processed edge. Therefore, by (i) it must be present on some edge in the set $E_{H_w} \cup \{\{u, w\}\}$. Also, since it collides with some rooted subtree from the set $\mathcal{Q}_i$, due to (ii) it must be present on some edge in the set $E_{H_v} \cup E_{H_x} \cup \{\{u, v\}, \{u, x\}\}$. Let us suppose that $\vec{S}$ is present on some edge in the set $E_{H_v} \cup \{\{u, v\}\}$. This along with the facts that $\vec{S}$ must be present on some edge in the set $E_{H_w} \cup \{\{u, w\}\}$ and $\|\vec{S}\|$ is a subtree of the host tree $H$, imply that $\vec{S}$ is present on the edge $\{u, v\}$. Alternatively, if we let $\vec{S}$ to be present on some edge in the set $E_{H_x} \cup \{\{u, x\}\}$, then following similar reasoning we can show that it must be present on the edge $\{u, x\}$. Therefore, we conclude that $\vec{S}$ must be present on either edge $\{u, v\}$ or edge $\{u, x\}$ or both. This is exactly what the lemma states for edges of type (iv).

According to Lemma 3.3, these are the only possible types of edges that are encountered in GREEDY-WA. This observation completes the proof. $\square$

### 3.2.4 Type (i), (ii) and (iii) Edges

According to our notation, $\lambda^{\mathrm{GDY}}(\mathcal{P}_i)$ is the set of wavelengths used by GREEDY-WA for assigning wavelengths to all the rooted subtrees present on host tree edges that are processed in the first $i$ rounds of wavelength assignment. Hence, the number of wavelengths used by GREEDY-WA after $i$ rounds of wavelength assignment is given by $|\lambda^{\mathrm{GDY}}(\mathcal{P}_i)|$. By this convention $|\lambda^{\mathrm{GDY}}(\mathcal{P}_0)| = |\lambda^{\mathrm{GDY}}(\emptyset)| = 0$ and $|\lambda^{\mathrm{GDY}}(\mathcal{P}_{|E_H|})| = |\lambda^{\mathrm{GDY}}(\mathcal{R})|$.

First we study the case when the edge $e_i = \{u, v\}$ being processed during the $i$-th round of GREEDY-WA is of type (i), (ii) or (iii) defined in Lemma 3.3.

**Lemma 3.5.** *If edge $e_i = \{u, v\}$ being processed in the $i$-th round of GREEDY-WA is of type (i), (ii) or (iii) defined in Lemma 3.3, then*

$$|\lambda^{\text{GDY}}(\mathcal{P}_i)| \leq \max\left\{2l_{\{\vec{T}_H, \mathcal{R}\}}, |\lambda^{\text{GDY}}(\mathcal{P}_{i-1})|\right\}.$$

*Proof.* First note that the set $\mathcal{R}[\{u, v\}]$ of all the rooted subtrees present on host tree edge $e_i = \{u, v\}$, can be partitioned into sets $\mathcal{Q}_i$ and $\mathcal{P}_{i-1}[\{u, v\}]$. Therefore

$$|\mathcal{Q}_i| = |\mathcal{R}[\{u, v\}]| - |\mathcal{P}_{i-1}[\{u, v\}]| \leq 2l_{\{\vec{T}, \mathcal{R}\}} - |\lambda^{\text{GDY}}(\mathcal{P}_{i-1}[\{u, v\}])|, \tag{3.1}$$

where the last inequality is due to the fact that for any wavelength assignment, the number of wavelengths required in order to assign wavelengths to a set of rooted subtrees can never exceed the cardinality of the set of rooted subtrees.

Since the edge $e_i = \{u, v\}$ being processed in the $i$-th round of wavelength assignment is of type (i), (ii) or (iii) defined in Lemma 3.3, according to Lemma 3.4, if a rooted subtree $\vec{P} \in \mathcal{P}_{i-1}$ that has already been assigned some wavelength in the first $i - 1$ rounds of GREEDY-WA, collides with any rooted subtree $\vec{Q} \in \mathcal{Q}_i$ that is to be assigned wavelength in the $i$-th round, then $\vec{P} \in \mathcal{P}_{i-1}[\{u, v\}]$. Hence, any wavelength present in the set $\lambda^{\text{GDY}}(\mathcal{P}_{i-1})$ but absent in the set $\lambda^{\text{GDY}}(\mathcal{P}_{i-1}[\{u, v\}])$ can be safely assigned to any rooted subtree in the set $\mathcal{Q}_i$. There are $|\lambda^{\text{GDY}}(\mathcal{P}_{i-1})| - |\lambda^{\text{GDY}}(\mathcal{P}_{i-1}[\{u, v\}])|$ such wavelengths. GREEDY-WA tries to reuse these wavelengths first and if there are still unassigned rooted subtrees left in $\mathcal{Q}_i$, it starts to assign new wavelengths to those rooted subtrees. In the worst case we need $|\mathcal{Q}_i|$ wavelengths during the $i$-th round of wavelength assignment. Therefore, the number of new wavelengths required in the $i$-th round is given by

$$\begin{aligned}
|\lambda^{\text{GDY}}(\mathcal{P}_i)| - |\lambda^{\text{GDY}}(\mathcal{P}_{i-1})| &\leq \left[|\mathcal{Q}_i| - \left(|\lambda^{\text{GDY}}(\mathcal{P}_{i-1})| - |\lambda^{\text{GDY}}(\mathcal{P}_{i-1}[\{u, v\}])|\right)\right]^+ \\
&\leq \left[\left(2l_{\{\vec{T}_H, \mathcal{R}\}} - |\lambda^{\text{GDY}}(\mathcal{P}_{i-1}[\{u, v\}])|\right)\right. \\
&\qquad \left. - \left(|\lambda^{\text{GDY}}(\mathcal{P}_{i-1})| - |\lambda^{\text{GDY}}(\mathcal{P}_{i-1}[\{u, v\}])|\right)\right]^+ \\
&= \left[2l_{\{\vec{T}_H, \mathcal{R}\}} - |\lambda^{\text{GDY}}(\mathcal{P}_{i-1})|\right]^+,
\end{aligned} \tag{3.2}$$

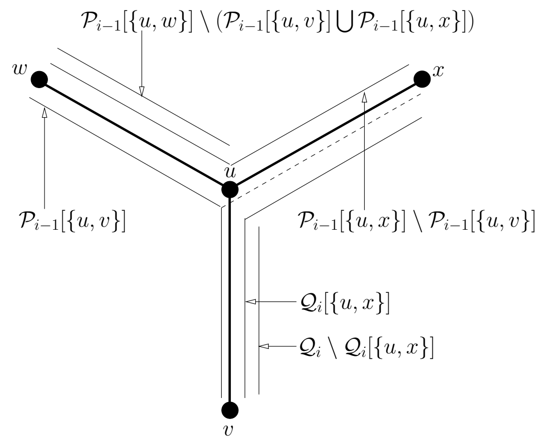



Figure 3.3: Sets of interesting rooted subtrees encountered while processing edge $\{u, v\}$ of type (iv) defined in Lemma 3.3

where the second inequality is by equation (3.1).

From equation (3.2), we obtain

$$\begin{aligned}|\lambda^{\mathrm{GDY}}(\mathcal{P}_i)| &\leq |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})| + \left[2l_{\{\vec{T}_H,\mathcal{R}\}} - |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})|\right]^+ \\ &= \max\left\{2l_{\{\vec{T}_H,\mathcal{R}\}}, |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})|\right\}.\end{aligned}$$

□

### 3.2.5 Type (iv) Edges

Next we consider the case when edge $e_i = \{u, v\}$ being processed during the $i$-th round of GREEDY-WA is of type (iv) defined in Lemma 3.3. As stated in Lemma 3.3, we assume that edge $e_i = \{u, v\}$ is such that (i) vertex $u$ was discovered before vertex $v$ in the BFS; (ii) all the edges adjacent to vertex $v$ are unprocessed after the first $i-1$ rounds of wavelength assignment; and (iii) vertex $u$ has degree 3 with adjacent edges $\{u, v\}$, $\{u, w\}$ and $\{u, x\}$ of which edge $\{u, w\}$ has already been processed while edge $\{u, x\}$ has not yet been processed.

As we shall discuss later in Lemma 3.6, in this case the set of relevant rooted subtrees consist of $\mathcal{P}_{i-1}[\{u, w\}]$, the set of rooted subtrees that have been assigned wavelengths in the first $i-1$ rounds of wavelength assignment and are present on the host tree edge $\{u, w\}$, and $\mathcal{Q}_i$, the set of rooted subtrees that are to be assigned wavelengths in the $i$-th round. These rooted subtrees are shown in more detail in Figure 3.3.

More specifically, we can partition the sets $\mathcal{P}_{i-1}[\{u, w\}]$ and $\mathcal{Q}_i$ of the relevant subtrees based on whether they are present or absent on the three host tree edges $\{u, v\}, \{u, w\}, \{u, x\}$. In Figure 3.3, we show representative rooted subtrees from the relevant partitions. The presence of a solid line in a representative rooted subtree on an edge implies that every rooted subtree of that set must be present on that edge. Similarly, the absence of a line in a representative rooted subtree on an edge implies that no rooted subtree of that set can be present on that edge. If some rooted subtrees of a set may be present on an edge, then the representative rooted subtree for that set has a dotted line on that edge in the figure.

As already stated, GREEDY-WA assigns wavelengths to the rooted subtrees in the set $\mathcal{Q}_i$ using two different schemes (PROCESS-EDGE-1 and PROCESS-EDGE-2) and then selects the better (the one using fewer new wavelengths) of the two. The basic difference between the two schemes is that of all the wavelengths in the set $\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u, w\}])$, PROCESS-EDGE-1 focuses on maximizing the reuse of wavelengths from the set $\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u, v\}])$, whereas PROCESS-EDGE-2 focuses on maximizing the reuse of wavelengths from the set $\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u, x\}] \setminus \mathcal{P}_{i-1}[\{u, v\}])$.

**Lemma 3.6.** *If edge $e_i = \{u, v\}$ being processed in the $i$-th round of GREEDY-WA is of type (iv) defined in Lemma 3.3, then*

$$|\lambda^{\mathrm{GDY}}(\mathcal{P}_i)| = \max\left\{\left|\lambda^{\mathrm{GDY}}(\mathcal{Q}_i \cup \mathcal{P}_{i-1}[\{u, w\}])\right|, |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})|\right\},$$

*where the edge $\{u, w\} \in E_H$ is as defined in Lemma 3.3.*

*Proof.* Since the edge $e_i = \{u, v\}$ being processed in the $i$-th round of wavelength assignment is of type (iv) defined in Lemma 3.3, according to Lemma 3.4, if a rooted subtree that has already been assigned some wavelength in the first $i-1$ rounds of GREEDY-WA, collides with any rooted subtree that is to be assigned wavelength in the $i$-th round, then it must belong to the set $\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}]$. Since $\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}] \subseteq \mathcal{P}_{i-1}[\{u,w\}]$, this implies that any rooted subtree in the set $\mathcal{P}_{i-1} \setminus \mathcal{P}_{i-1}[\{u,w\}]$ cannot collide with any rooted subtree in the set $\mathcal{Q}_i$. Therefore, any wavelength already assigned to some rooted subtree in the set $\mathcal{P}_{i-1} \setminus \mathcal{P}_{i-1}[\{u,w\}]$, but not to any rooted subtree in the set $\mathcal{P}_{i-1}[\{u,w\}]$, can be assigned to any rooted subtree in the set $\mathcal{Q}_i$. There are $|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})| - |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}])|$ such wavelengths. During the $i$-th round of wavelength assignment, let $\mathcal{N}_i \subseteq \mathcal{Q}_i$ be the set of rooted subtrees which do not share wavelengths with rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}]$, i.e, $\mathcal{Q}_i \setminus \mathcal{N}_i$ is the largest subset of the set $\mathcal{Q}_i$ such that $|\lambda^{\mathrm{GDY}}((\mathcal{Q}_i \setminus \mathcal{N}_i) \cup \mathcal{P}_{i-1}[\{u,w\}])| = |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}])|$. We need $|\lambda^{\mathrm{GDY}}(\mathcal{N}_i)|$ additional wavelengths for assigning wavelengths to all the rooted subtrees in the set $\mathcal{N}_i$ and there are $|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})| - |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}])|$ available wavelengths that can be used without adding any new wavelength in the $i$-th round of wavelength assignment. In GREEDY-WA, we always try to reuse these available wavelengths before adding any new wavelengths. Therefore, the total number of wavelengths required at the end of $i$-th round of wavelength assignment is

$$\begin{aligned}
|\lambda^{\mathrm{GDY}}(\mathcal{P}_i)| &= \Big[|\lambda^{\mathrm{GDY}}(\mathcal{N}_i)| - \Big(|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})| - |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}])|\Big)\Big]^+ \\
&\quad + |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})| \\
&= \Big[|\lambda^{\mathrm{GDY}}(\mathcal{N}_i)| + \big|\lambda^{\mathrm{GDY}}((\mathcal{Q}_i \setminus \mathcal{N}_i) \cup \mathcal{P}_{i-1}[\{u,w\}])\big| - |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})|\Big]^+ \\
&\quad + |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})| \\
&= \Big[\big|\lambda^{\mathrm{GDY}}(\mathcal{Q}_i \cup \mathcal{P}_{i-1}[\{u,w\}])\big| - |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})|\Big]^+ + |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})| \\
&= \max\left\{\big|\lambda^{\mathrm{GDY}}(\mathcal{Q}_i \cup \mathcal{P}_{i-1}[\{u,w\}])\big|, |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})|\right\},
\end{aligned}$$

where the third equality is due to the fact that the rooted subtrees in the set $\mathcal{N}_i$ do not share any wavelength with the rooted subtrees in the set $(\mathcal{Q}_i \setminus \mathcal{N}_i) \cup \mathcal{P}_{i-1}[\{u,w\}]$.

$\square$

In light of Lemma 3.6, we see that it makes sense to evaluate bounds for $|\lambda^{\text{GDY}}(\mathcal{Q}_i \cup \mathcal{P}_{i-1}[\{u,w\}])|$. Using the notation of the lemma, if $\mathcal{N}_i \subseteq \mathcal{Q}_i$ is the set of rooted subtrees that do not share wavelengths with any rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}]$, then

$$\begin{aligned}\left|\lambda^{\text{GDY}}(\mathcal{Q}_i \cup \mathcal{P}_{i-1}[\{u,w\}])\right| &= |\lambda^{\text{GDY}}(\mathcal{N}_i)| + \left|\lambda^{\text{GDY}}((\mathcal{Q}_i \setminus \mathcal{N}_i) \cup \mathcal{P}_{i-1}[\{u,w\}])\right| \\ &= |\lambda^{\text{GDY}}(\mathcal{N}_i)| + |\lambda^{\text{GDY}}(\mathcal{P}_{i-1}[\{u,w\}])|.\end{aligned}$$

Hence, in order to limit the use of new wavelengths in the $i$-th round of wavelength assignment, we try to minimize $|\lambda^{\text{GDY}}(\mathcal{N}_i)|$, the number of wavelengths used in the $i$-th round of wavelength assignment that are different from the wavelengths assigned to the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}]$.

For any set $\mathcal{S}$ of rooted subtrees on the given bidirected tree $\vec{T}_H$ such that the complement of their conflict graph is bipartite, i.e., $\bar{G}_{\mathcal{S}}$ is bipartite, we denote the size of maximum matching [41, p.67] in $\bar{G}_{\mathcal{S}}$ by $m_{\{\vec{T}_H,\mathcal{S}\}}$.

**Lemma 3.7.** *If the edge $e_i = \{u,v\}$ is of type (iv) defined in Lemma 3.3, and the wavelength assignment generated by PROCESS-EDGE-1 is used in the $i$-th round of GREEDY-WA, then*

$$\left|\lambda^{\text{GDY}}(\mathcal{Q}_i \cup \mathcal{P}_{i-1}[\{u,w\}])\right| \le 2l_{\{\vec{T}_H,\mathcal{R}\}} + |\mathcal{Q}_i| - m_{\{\vec{T}_H,\mathcal{R}[\{u,v\}]\}} + m_{\{\vec{T}_H,\mathcal{P}_{i-1}[\{u,v\}]\}}.$$

*Proof.* In order to limit $|\lambda^{\text{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i)| - |\lambda^{\text{GDY}}(\mathcal{P}_{i-1}[\{u,w\}])|$, PROCESS-EDGE-1 finds the maximum number of disjoint pairs $\vec{R}, \vec{S}$ of rooted subtrees such that one of the following is true:

(i) Both $\vec{R}, \vec{S} \in \mathcal{Q}_i$, and in this case they are assigned the same (possibly new) wavelength.

(ii) $\vec{R} \in \mathcal{Q}_i, \vec{S} \in \mathcal{P}_{i-1}[\{u,v\}]$, and in this case $\vec{R}$ is assigned the same wavelength as $\vec{S}$.

Note that some rooted subtrees in the set $\mathcal{Q}_i$ may remain unpaired.

PROCESS-EDGE-1 finds such pairs of rooted subtrees by studying graph $B_1$. First note that the sets $\mathcal{P}_{i-1}[\{u,v\}]$ and $\mathcal{Q}_i$ partition the set $\mathcal{R}[\{u,v\}]$, therefore by Lemma 3.1, graph $\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}$ is bipartite. This, along with the fact that $E_{G_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}} \subseteq E_{B_1}$, implies that $\bar{B}_1$ is also bipartite. Hence, it is easy to find a maximum matching in $\bar{B}_1$. Let $M \subseteq E_{\bar{B}_1}$ be any matching in $\bar{B}_1$. Observe that the edges are added to $B_1$ (lines 2-6) in such a way that if edge $\{\vec{R},\vec{S}\} \in M$, then one of the following holds:

(i) Both $\vec{R},\vec{S} \in \mathcal{Q}_i$.

(ii) $\vec{R} \in \mathcal{P}_{i-1}, \vec{S} \in \mathcal{Q}_i$, and there is no $\vec{U} \in \mathcal{P}_{i-1}$ such that $\vec{S},\vec{U}$ collide and $\lambda^{\mathrm{GDY}}(\vec{R}) = \lambda^{\mathrm{GDY}}(\vec{U})$.

(iii) Both $\vec{R},\vec{S} \in \mathcal{P}_{i-1}$ and $\lambda^{\mathrm{GDY}}(\vec{R}) = \lambda^{\mathrm{GDY}}(\vec{S})$.

This means that if edge $\{\vec{R},\vec{S}\} \in M$, then rooted subtrees $\vec{R}$ and $\vec{S}$ can be assigned the same wavelength. Note that the matched edges of type (i) and (ii) correspond to the rooted subtree pairs of type (i) and (ii), respectively. A matched edge of type (iii) does not provide any additional information; it simply lists all the pairs of rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,v\}]$ that have already been assigned the same wavelengths. Since the number of edges of type (iii) is already fixed, a maximum matching in $\bar{B}_1$ determines the maximum number of edges of types (i) and (ii), i.e., it determines the maximum number of rooted subtree pairs described above.

First assume that the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,v\}]$ do not share wavelengths with any of the rooted subtree in the set $\mathcal{P}_{i-1}[\{u,w\}]\setminus\mathcal{P}_{i-1}[\{u,v\}]$, although they may share wavelengths amongst themselves. As a consequence of Lemma 3.1, more than two rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,v\}]$ cannot have the same wavelength. Starting from any maximum matching $M_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}} \subseteq E_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}}$ in graph $\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}$, we can construct a matching $M \subseteq E_{\bar{B}_1}$ in graph $\bar{B}_1$ by first removing and then adding the edges described next. We remove every matched edge $\{\vec{R},\vec{S}\} \in M_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}}$ for which one of the following is true:

(i) Both $\vec{R},\vec{S} \in \mathcal{P}_{i-1}[\{u,v\}]$ such that $\lambda^{\mathrm{GDY}}(\vec{R}) \neq \lambda^{\mathrm{GDY}}(\vec{S})$, and there is no rooted subtree $\vec{U} \in \mathcal{P}_{i-1}[\{u,v\}]$ such that $\lambda^{\mathrm{GDY}}(\vec{U}) \notin \{\lambda^{\mathrm{GDY}}(\vec{R}), \lambda^{\mathrm{GDY}}(\vec{S})\}$.

(ii) Both $\vec{R},\vec{S} \in \mathcal{P}_{i-1}[\{u,v\}]$ such that $\lambda^{\mathrm{GDY}}(\vec{R}) \neq \lambda^{\mathrm{GDY}}(\vec{S})$, and there is a rooted subtree $\vec{U} \in \mathcal{P}_{i-1}[\{u,v\}]$ such that $\lambda^{\mathrm{GDY}}(\vec{U}) \in \{\lambda^{\mathrm{GDY}}(\vec{R}), \lambda^{\mathrm{GDY}}(\vec{S})\}$.

(iii) $\vec{R} \in \mathcal{Q}_i$, $\vec{S} \in \mathcal{P}_{i-1}[\{u,v\}]$, and there is a rooted subtree $\vec{U} \in \mathcal{P}_{i-1}$ such that $\lambda^{\mathrm{GDY}}(\vec{U}) = \lambda^{\mathrm{GDY}}(\vec{S})$.

Consider rooted subtrees $\vec{R},\vec{S} \in \mathcal{P}_{i-1}[\{u,v\}]$ with $\lambda^{\mathrm{GDY}}(\vec{R}) = \lambda^{\mathrm{GDY}}(\vec{S})$. Since $M_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}}$ is a maximum matching in $\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}$, either edge $\{\vec{R},\vec{S}\} \in M_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}}$, or at least one of the rooted subtrees $\vec{R},\vec{S}$ is matched to some other rooted subtree in $M_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}}$.[2] In the case when rooted subtrees $\vec{R},\vec{S}$ are not already matched to each other in $M_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}}$, the edge(s) adjacent to $\vec{R}$ or $\vec{S}$ (or both) in $M_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}}$ is (are) either of type (ii) or of type (iii) and is (are) therefore removed from the matching. Hence, we can safely add edge $\{\vec{R},\vec{S}\}$ to the matching. Let the set of removed edges of type (i), (ii) and (iii) be $E_{\mathrm{r(i)}}$, $E_{\mathrm{r(ii)}}$ and $E_{\mathrm{r(iii)}}$, respectively, and the set of added edges be $E_{\mathrm{a}}$. Observe that for every removed edge in the set $E_{\mathrm{r(ii)}} \cup E_{\mathrm{r(iii)}}$, there is a corresponding edge in the set $E_{\mathrm{a}}$ added to the matching such that for at most two removed edges in the set $E_{\mathrm{r(ii)}} \cup E_{\mathrm{r(iii)}}$, the corresponding added edge in the set $E_{\mathrm{a}}$ can be the same; therefore $|E_{\mathrm{a}}| \geq \frac{1}{2}(|E_{\mathrm{r(ii)}}| + |E_{\mathrm{r(iii)}}|)$. Hence, we can lower bound the size of maximum matching $M_{\bar{B}_1} \subseteq E_{\bar{B}_1}$ in graph $\bar{B}_1$ by the size of $M$, a valid matching in the graph. Note that $|M|$ is equal to $|M_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}}| = m_{\{\vec{T}_H,\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i\}}$ minus the number of edges removed plus the

[2]It may happen that both the rooted subtrees $\vec{R},\vec{S}$ are matched to different vertices in $M_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}}$

number of edges added. Thus

$$
\begin{aligned}
|M_{\bar{B}_1}| &\geq |M| = m_{\{\vec{T}_H, \mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i\}} - \big(|E_{\mathrm{r(i)}}| + |E_{\mathrm{r(ii)}}| + |E_{\mathrm{r(iii)}}| - |E_{\mathrm{a}}|\big) \\
&\geq m_{\{\vec{T}_H, \mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i\}} - \big(|E_{\mathrm{r(i)}}| + |E_{\mathrm{a}}|\big) \\
&\geq m_{\{\vec{T}_H, \mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i\}} - m_{\{\vec{T}_H, \mathcal{P}_{i-1}[\{u,v\}]\}}, \qquad (3.3)
\end{aligned}
$$

where we are using the fact that $E_{\mathrm{a}} \cup E_{\mathrm{r(i)}}$, the set of removed edges of type (i) and the set of added edges form a matching in the bipartite graph $\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]}$. To see this, note that $E_{\mathrm{a}} \cup E_{\mathrm{r(i)}} \subseteq E_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]}}$, and the end vertices of edges in the sets $E_{\mathrm{a}}, E_{\mathrm{r(i)}}$ are distinct.

Note that the vertex set $V_{\bar{B}_1}$ corresponds to all the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i$, and an edge in matching $M_{\bar{B}_1}$ determines two rooted subtrees which share their wavelength after this round of wavelength assignment. Therefore, using inequality (3.3) and the fact that the subsets $\mathcal{P}_{i-1}[\{u,v\}]$ and $\mathcal{Q}_i$ partition the set $\mathcal{R}[\{u,v\}]$,

$$
\begin{aligned}
\big|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i)\big| &\leq \big|\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i\big| - |M_{\bar{B}_1}| \\
&\leq |\mathcal{P}_{i-1}[\{u,v\}]| + |\mathcal{Q}_i| - m_{\{\vec{T}_H, \mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i\}} \\
&\quad + m_{\{\vec{T}_H, \mathcal{P}_{i-1}[\{u,v\}]\}}. \qquad (3.4)
\end{aligned}
$$

Thus, using inequality (3.4), the number of wavelengths required for assigning wavelengths to all the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i$ is

$$
\begin{aligned}
\big|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i)\big| &= \big|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i)\big| \\
&\quad - \big|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i)\big| \\
&\quad + \big|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i)\big| \\
&\leq |\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| + |\mathcal{P}_{i-1}[\{u,v\}]| + |\mathcal{Q}_i| \\
&\quad - m_{\{\vec{T}_H, \mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i\}} + m_{\{\vec{T}_H, \mathcal{P}_{i-1}[\{u,v\}]\}} \\
&\leq 2l_{\{\vec{T}_H, \mathcal{R}\}} + |\mathcal{Q}_i| - m_{\{\vec{T}_H, \mathcal{R}[\{u,v\}]\}} \\
&\quad + m_{\{\vec{T}_H, \mathcal{P}_{i-1}[\{u,v\}]\}}. \qquad (3.5)
\end{aligned}
$$

For the first inequality, we are using the fact that $|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i)| - |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i)|$ is the number of wavelengths used for assigning wavelengths to all the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ that are different from the wavelengths used for assigning the wavelengths to rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i$; therefore, this number is upper bounded by $|\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]|$. For the final inequality, we use the fact that the subsets $\mathcal{P}_{i-1}[\{u,v\}]$ and $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ partition the set $\mathcal{P}_{i-1}[\{u,w\}] = \mathcal{R}[\{u,w\}]$.

Next, suppose some rooted subtree $\vec{R} \in \mathcal{P}_{i-1}[\{u,v\}]$ shares its wavelength with another rooted subtree $\vec{S} \in \mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$. In this case, the worst that can happen is that some rooted subtrees in the set $\mathcal{Q}_i$, that could have shared wavelength with rooted subtree $\vec{R}$, can no longer do so since they collide with rooted subtree $\vec{S}$. Hence the size of maximum matching $M_{\bar{B}_1}$ reduces by 1. The unit reduction is independent of the number of affected rooted subtrees in the set $\mathcal{Q}_i$, since in $M_{\bar{B}_1}$ rooted subtree $\vec{R}$ can be potentially matched to only one of them. On the other hand, the rooted subtrees $\vec{R} \in \mathcal{P}_{i-1}[\{u,v\}], \vec{S} \in \mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ sharing wavelength means that $|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i)| - |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i)|$, the number of wavelengths used for assigning wavelengths to all the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ that are different from the wavelengths used for assigning wavelengths to the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i$, also reduces by 1. Applying both the observations, we note that the final inequality in (3.5) still holds. $\square$

**Lemma 3.8.** *If the edge $e_i = \{u,v\}$ is of type (iv) defined in Lemma 3.3, and the wavelength assignment generated by PROCESS-EDGE-2 is used in the $i$-th round of GREEDY-WA, then*

$$\left|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i)\right| \leq 2l_{\{\vec{T}_H,\mathcal{R}\}} + [g-h]^+ ,$$

*where*

$$
\begin{aligned}
g &= |\mathcal{Q}_i[\{u,x\}]| + |\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| - |\mathcal{Q}_i|, \\
h &= \left[ |\mathcal{Q}_i[\{u,x\}]| + \frac{|\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]|}{2} + m_{\{\vec{T}_H, \mathcal{R}[\{u,x\}]\}} - 2l_{\{\vec{T}_H, \mathcal{R}\}} \right]^+ .
\end{aligned}
$$

*Proof.* Since $\mathcal{R}[\{u,w\}] = \mathcal{P}_{i-1}[\{u,w\}]$ can be partitioned into $\mathcal{P}_{i-1}[\{u,v\}]$ and $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$,

$$
\begin{aligned}
|\mathcal{P}_{i-1}[\{u,v\}]| + |\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| &= |\mathcal{P}_{i-1}[\{u,w\}]| \\
&= |\mathcal{R}[\{u,w\}]| = 2l_{\{\vec{T}_H, \mathcal{R}\}}.
\end{aligned}
$$

Also, $\mathcal{R}[\{u,v\}]$ can be partitioned into $\mathcal{P}_{i-1}[\{u,v\}]$ and $\mathcal{Q}_i$, therefore

$$
|\mathcal{P}_{i-1}[\{u,v\}]| + |\mathcal{Q}_i| = |\mathcal{R}[\{u,v\}]| = 2l_{\{\vec{T}_H, \mathcal{R}\}}.
$$

From the above two equations, it follows that

$$
|\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| = |\mathcal{Q}_i|. \tag{3.6}
$$

Since $\mathcal{Q}_i$ can be partitioned into $\mathcal{Q}_i[\{u,x\}]$ and $\mathcal{Q}_i \setminus \mathcal{Q}_i[\{u,x\}]$, and $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ can be partitioned into $\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ and $\mathcal{P}_{i-1}[\{u,w\}] \setminus (\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}])$; from equation (3.6), it follows that

$$
\begin{aligned}
&\big|\mathcal{P}_{i-1}[\{u,w\}] \setminus (\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}])\big| + |\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| \\
&\quad = |\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| = |\mathcal{Q}_i \setminus \mathcal{Q}_i[\{u,x\}]| + |\mathcal{Q}_i[\{u,x\}]| = |\mathcal{Q}_i|. \qquad (3.7)
\end{aligned}
$$

In PROCESS-EDGE-2, first we find the maximum number of disjoint pairs $\vec{R}, \vec{S}$ of rooted subtrees such that one of the following is true:

(i) Both $\vec{R}, \vec{S} \in \mathcal{Q}_i[\{u,x\}]$. In this case, both $\vec{R}$ and $\vec{S}$ are assigned the same wavelength (we shall specify exactly which wavelength is assigned in a moment).

(ii) $\vec{R} \in \mathcal{Q}_i[\{u,x\}]$ and $\vec{S} \in \mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ such that $\vec{R}$ can be assigned the same wavelength as $\vec{S}$. In this case $\vec{R}$ is indeed assigned the same wavelength as $\vec{S}$.

We find such pairs of rooted subtrees by studying the graph $B_2$. First note that the sets $\mathcal{Q}_i[\{u,x\}]$ and $\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ are disjoint subsets of the set $\mathcal{R}[\{u,x\}]$; therefore by Lemma 3.1, the graph $\bar{G}_{(\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]) \cup \mathcal{Q}_i[\{u,x\}]}$ is bipartite. This, along with the fact that $E_{G_{(\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]) \cup \mathcal{Q}_i[\{u,x\}]}} \subseteq E_{B_2}$, implies that $\bar{B}_2$ is also bipartite. Hence, it is easy to find a maximum matching in $\bar{B}_2$. Let $M \subseteq E_{\bar{B}_2}$ be any matching in $\bar{B}_2$. Observe that the edges are added to $B_2$ in such a way that if edge $\{\vec{R}, \vec{S}\} \in M$, then one of the following holds:

(i) Both $\vec{R}, \vec{S} \in \mathcal{Q}_i[\{u,x\}]$.

(ii) $\vec{R} \in \mathcal{Q}_i[\{u,x\}], \vec{S} \in \mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$, and there is no $\vec{U} \in \mathcal{P}_{i-1}$ such that $\vec{R}, \vec{U}$ collide and $\lambda^{\mathrm{GDY}}(\vec{S}) = \lambda^{\mathrm{GDY}}(\vec{U})$.

(iii) Both $\vec{R}, \vec{S} \in \mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ and $\lambda^{\mathrm{GDY}}(\vec{R}) = \lambda^{\mathrm{GDY}}(\vec{S})$.

This means that if edge $\{\vec{R}, \vec{S}\} \in M$, then the rooted subtrees $\vec{R}, \vec{S}$ can be assigned the same wavelength. Note that the matched edges of type (i) and (ii) correspond to the rooted subtree pairs of type (i) and (ii), respectively. A matched edge of type (iii) does not provide any additional information; it simply lists all the pairs of rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ that have already been assigned the same wavelengths. Since the number of edges of type (iii) is already fixed, a maximum matching in $\bar{B}_2$ determines the maximum number of edges of types (i) and (*ii*), i.e., it determines the maximum number of rooted subtree pairs described above.

First, we assume that the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ do not share wavelengths with any rooted subtree in the set $\mathcal{P}_{i-1}[\{u,v\}]$, although they may share wavelengths amongst themselves. Let $M_{\bar{B}_2} \subseteq E_{\bar{B}_2}$ be a maximum matching in $\bar{B}_2$. Let the number of type (i), (ii) and (iii) edges in the matching be $t_1, t_2, t_3$, respectively. In this case the size of the maximum matching in $\bar{B}_2$ is lower bounded as

$$\begin{aligned}
|M_{\bar{B}_2}| = t_1 + t_2 + t_3 \;\; &\geq \;\; m_{\{\vec{T}_H, \mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])\}} \\
&\quad - m_{\{\vec{T}_H, \mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]\}} \\
&\geq \;\; \Bigg[ m_{\{\vec{T}_H, \mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])\}} \\
&\quad - \frac{|\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]|}{2} \Bigg]^{+}, \qquad (3.8)
\end{aligned}$$

where $m_{\{\vec{T}_H, \mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])\}}$ and $m_{\{\vec{T}_H, \mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]\}}$ are the sizes of maximum matchings in the bipartite graphs $\bar{G}_{\mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])}$ and $\bar{G}_{\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]}$, respectively. The reasoning for the initial inequality follows exactly as the reasoning for inequality (3.3) presented in the proof of Lemma 3.7. For the final inequality, we use the facts that the size of any matching in the bipartite graph $\bar{G}_{\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]}$ must be smaller than half of the size of its vertex set, and the size of a matching cannot be negative. Note that $\bar{G}_{\mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])}$ is a subgraph of $\bar{G}_{\mathcal{R}[\{u,x\}]}$ induced by the vertex set corresponding to the rooted subtrees in the set $\mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])$. If the size of a maximum matching in $\bar{G}_{\mathcal{R}[\{u,x\}]}$ is $m_{\{\vec{T}_H, \mathcal{R}[\{u,x\}]\}}$, then the size of a maximum matching in $\bar{G}_{\mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])}$ is bounded as

$$\begin{aligned}
m_{\{\vec{T}_H, \mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])\}} \geq &\Bigg[ m_{\{\vec{T}_H, \mathcal{R}[\{u,x\}]\}} - |\mathcal{R}[\{u,x\}]| \\
&\quad + \left|\mathcal{Q}_i[\{u,x\}] \cup \Big(\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]\Big)\right| \Bigg]^{+} \\
= &\Bigg[ |\mathcal{Q}_i[\{u,x\}]| + |\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| \\
&\quad + m_{\{\vec{T}_H, \mathcal{R}[\{u,x\}]\}} - 2 l_{\{\vec{T}_H, \mathcal{R}\}} \Bigg]^{+}. \qquad (3.9)
\end{aligned}$$

This is because if we consider a maximum matching $M_{\bar{G}_{\mathcal{R}[\{u,x\}]}} \subseteq E_{\bar{G}_{\mathcal{R}[\{u,x\}]}}$ in the graph $\bar{G}_{\mathcal{R}[\{u,x\}]}$, any edge $\{\vec{R},\vec{S}\} \in M_{\bar{G}_{\mathcal{R}[\{u,x\}]}}$ can be classified into one of the following three types:

(i) Both $\vec{R},\vec{S} \in \mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])$.

(ii) Rooted subtree $\vec{R} \in \mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])$ whereas rooted subtree $\vec{S} \in \mathcal{R}[\{u,x\}] \setminus (\mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]))$.

(iii) Both $\vec{R},\vec{S} \in \mathcal{R}[\{u,x\}] \setminus (\mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]))$.

Let the set of edges of type (i), (ii) and (iii) be $E_{(\mathrm{i})}, E_{(\mathrm{ii})}, E_{(\mathrm{iii})}$, respectively. Clearly, $E_{(\mathrm{i})}$ is a valid matching in the graph $\bar{G}_{\mathcal{Q}_i[\{u,x\}]\cup(\mathcal{P}_{i-1}[\{u,x\}]\setminus\mathcal{P}_{i-1}[\{u,v\}])}$, therefore a lower bound for $|E_{(\mathrm{i})}|$ can be treated as a lower bound for $m_{\{\vec{T}_H,\mathcal{Q}_i[\{u,x\}]\cup(\mathcal{P}_{i-1}[\{u,x\}]\setminus\mathcal{P}_{i-1}[\{u,v\}])\}}$. Also, since maximum matching $M_{\bar{G}_{\mathcal{R}[\{u,x\}]}}$ can be partitioned into sets $E_{(\mathrm{i})}, E_{(\mathrm{ii})}, E_{(\mathrm{iii})}$, we get

$$\begin{aligned} m_{\{\vec{T}_H,\mathcal{Q}_i[\{u,x\}]\cup(\mathcal{P}_{i-1}[\{u,x\}]\setminus\mathcal{P}_{i-1}[\{u,v\}])\}} &\geq |E_{(\mathrm{i})}| \\ &\geq m_{\{\vec{T}_H,\mathcal{R}[\{u,x\}]\}} - |E_{(\mathrm{ii})}| - |E_{(\mathrm{iii})}|. \quad (3.10) \end{aligned}$$

Since an edge in the set $E_{(\mathrm{ii})}$ requires one of the rooted subtree from the set $\mathcal{R}[\{u,x\}] \setminus (\mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]))$ and an edge in the set $E_{(\mathrm{iii})}$ requires both of the rooted subtrees from the same set, we have

$$\begin{aligned} |E_{(\mathrm{ii})}| + 2|E_{(\mathrm{iii})}| &\leq \left|\mathcal{R}[\{u,x\}] \setminus (\mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]))\right| \\ &= |\mathcal{R}[\{u,x\}]| - \left|\mathcal{Q}_i[\{u,x\}] \cup (\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])\right|. \quad (3.11) \end{aligned}$$

From inequalities (3.10), (3.11) and the fact that the size of a matching cannot be negative, we obtain the required inequality (3.9).

From equations (3.8) and (3.9),

$$\begin{aligned} |M_{\bar{B}_2}| &= t_1 + t_2 + t_3 \\ &\geq \Bigg[\Big[|\mathcal{Q}_i[\{u,x\}]| + |\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| + m_{\{\vec{T}_H, \mathcal{R}[\{u,x\}]\}} - 2l_{\{\vec{T}_H, \mathcal{R}\}}\Big]^+ \\ &\quad - \frac{|\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]|}{2}\Bigg]^+ \\ &= \Bigg[|\mathcal{Q}_i[\{u,x\}]| + \frac{|\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]|}{2} + m_{\{\vec{T}_H, \mathcal{R}[\{u,x\}]\}} - 2l_{\{\vec{T}_H, \mathcal{R}\}}\Bigg]^+ \\ &= h. \end{aligned} \tag{3.12}$$

Note that each of these $h$ edges is of type (i), (ii) or (iii) described before.

Observe that PROCESS-EDGE-2 assigns wavelengths to the unassigned rooted subtrees in the set $\mathcal{Q}_i$ in the following order:

(i) First, all those matched pairs of rooted subtree are considered in which one of the rooted subtree is in the set $\mathcal{Q}_i[\{u,x\}]$ and the other is in the set $\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$. For every such matched pair, the unassigned rooted subtree is assigned the same wavelength that has already been assigned to its matched partner during the first $i-1$ rounds of GREEDY-WA. The number of such rooted subtrees in the matching $M_{\bar{B}_2}$ is equal to $t_2$.

(ii) Next, the remaining rooted subtrees from the set $\mathcal{Q}_i[\{u,x\}]$ are randomly selected one-at-a-time for wavelength assignment. If the selected rooted subtree $\vec{R}$ was not matched, and if there is a wavelength that has already been used previously that can be safely assigned to $\vec{R}$, then that wavelength is used; otherwise, a new wavelength is used. On the other hand, if the selected rooted subtree $\vec{R}$ was matched to another rooted subtree $\vec{S}$, then clearly $\vec{S}$ is also unassigned. In this case both $\vec{R}$ and $\vec{S}$ are assigned the same wavelength. Again, preference is given to the wavelengths that are already in use over the use of new wavelengths. According to Lemma 3.4, rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus (\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}])$ can never collide with any rooted subtree in the set $\mathcal{Q}_i$. Therefore, any wavelength used for rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus (\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}])$, that is not used by any other rooted subtree in the set $\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$, can be assigned to any of the rooted subtrees in the set $\mathcal{Q}_i$. Let $z_1$ be the number wavelengths assigned to the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus (\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}])$ that are reused for rooted subtrees in the set $\mathcal{Q}_i[\{u,x\}]$ during this step of the subroutine. We can bound $z_1$ as

$$\begin{aligned} z_1 \;\; \geq \;\; \min \Big\{ & |\mathcal{Q}_i[\{u,x\}]| - t_1 - t_2, \\ & |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])| \\ & - |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])| \Big\}. \end{aligned} \tag{3.13}$$

Here the first term in min is the maximum number of wavelengths required for assigning wavelengths to all the rooted subtrees in the set $\mathcal{Q}_i[\{u,x\}]$ that remain unassigned after step (i) of the subroutine described above. The second term is the number of wavelengths used for assigning wavelengths to the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus (\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}])$ that are not used for any rooted subtree in the set $\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$.

(iii) Next, the remaining unassigned rooted subtrees (all the rooted subtrees in the set $\mathcal{Q}_i \setminus \mathcal{Q}_i[\{u,x\}]$) are assigned wavelengths one-at-a-time. Again preference is given to the wavelengths that are already in use over the use of new wavelengths. Since the rooted subtrees in the set $\mathcal{Q}_i \setminus \mathcal{Q}_i[\{u,x\}]$ can never collide with any rooted subtree in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$, any wavelength used for rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ that has not yet been reused for any rooted subtree in the set $\mathcal{Q}_i[\{u,x\}]$, can be assigned to any of the rooted subtrees in the set $\mathcal{Q}_i \setminus \mathcal{Q}_i[\{u,x\}]$. Let $z_2$ be the number of wavelengths assigned to the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ that are reused for rooted subtrees in the set $\mathcal{Q}_i \setminus \mathcal{Q}_i[\{u,x\}]$ during this step of the subroutine. We can bound $z_2$ as

$$z_2 \geq \min\Big\{|\mathcal{Q}_i \setminus \mathcal{Q}_i[\{u,x\}]|, \\ |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])| - t_2 - z_1\Big\}. \quad (3.14)$$

Here the first term in min is the maximum number of wavelengths required for assigning wavelengths all the rooted subtrees in the set $\mathcal{Q}_i \setminus \mathcal{Q}_i[\{u,x\}]$ and the second term is the number of wavelengths used for assigning wavelengths to the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ that have not yet been reused in the first two steps of the subroutine.

Let $z_3$ be the number of wavelengths used for assigning wavelengths to pairs of rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus (\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}])$, or to pairs of rooted subtrees where one of the rooted subtree belongs to the set $\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ and the other belongs to the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus (\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}])$. We can determine $z_3$ by subtracting the total number of wavelengths used for assigning wavelengths to all the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ from the sum of the total number of wavelengths used for assigning wavelengths to all the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ and the total number of rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \setminus (\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}])$. Hence, using equation (3.7),

$$\begin{aligned} z_3 &= \big|\mathcal{P}_{i-1}[\{u,w\}] \setminus (\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{P}_{i-1}[\{u,x\}])\big| + |\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| \\ &\quad - t_3 - |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])| \\ &= |\mathcal{Q}_i| - t_3 - |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])|. \end{aligned} \quad (3.15)$$

We note that the total number of wavelengths required for assigning wavelengths to all the rooted subtrees in the set $\mathcal{Q}_i \cup (\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])$ can be bounded as

$$\begin{aligned}
& \left|\lambda^{\mathrm{GDY}}(\mathcal{Q}_i \cup (\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]))\right| \\
= \;& |\mathcal{Q}_i \cup (\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])\,| - |M_{\bar{B}_2}| - z_1 - z_2 - z_3 \\
\leq \;& \left|\mathcal{Q}_i \cup (\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])\,\right| - |\mathcal{Q}_i| \\
& + \max\Big\{|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])| - |\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]|, \\
& \qquad |\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])| - |\mathcal{Q}_i \setminus \mathcal{Q}_i[\{u,x\}]| - t_1 - t_2, -t_1\Big\} \\
\leq \;& |\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| \\
& + \Big[|\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| - |\mathcal{Q}_i \setminus \mathcal{Q}_i[\{u,x\}]| - t_1 - t_2 - t_3\Big]^+ \\
\leq \;& |\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| + [g-h]^+ . \qquad (3.16)
\end{aligned}$$

To get the first inequality we need to perform some algebra (that we have omitted here) using equations (3.7), (3.13), (3.14), (3.15) and the fact that $|M_{\bar{B}_2}| = t_1+t_2+t_3$. For getting the second inequality we again use equation (3.7) along with the fact that the sets $\mathcal{Q}_i$ and $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ are mutually exclusive. For this step we also use the observation that the first and the third terms in max are always less than or equal to zero and in the second term $|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])| = |\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| - t_3$. Final inequality uses equations (3.7) and (3.12).

Using inequality (3.16), the number of wavelengths required for assigning wavelengths to all the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i$ is

$$\begin{aligned}
\left|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i)\right| \;=\; & \left|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i)\right| \\
& - \left|\lambda^{\mathrm{GDY}}(\mathcal{Q}_i \cup (\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]))\right| \\
& + \left|\lambda^{\mathrm{GDY}}(\mathcal{Q}_i \cup (\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]))\right| \\
\leq \;& |\mathcal{P}_{i-1}[\{u,v\}]| + |\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| + [g-h]^+ \\
= \;& 2l_{\{\vec{T}_H,\mathcal{R}\}} + [g-h]^+ . \qquad (3.17)
\end{aligned}$$

The inequality uses the fact that since the number of wavelengths used for assigning wavelengths to all the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,v\}]$ that are different from the wavelengths used for assigning wavelengths to the rooted subtrees in the set $\mathcal{Q}_i \cup (\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])$ is equal to $|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i)| - |\lambda^{\mathrm{GDY}}(\mathcal{Q}_i \cup (\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]))|$, it is upper bounded by $|\mathcal{P}_{i-1}[\{u,v\}]|$. For the final equality, we use the fact that the subsets $\mathcal{P}_{i-1}[\{u,v\}]$ and $\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ partition the set $\mathcal{P}_{i-1}[\{u,w\}] = \mathcal{R}[\{u,w\}]$.

Suppose some rooted subtree $\vec{R} \in \mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ shares its wavelength with another rooted subtree $\vec{S} \in \mathcal{P}_{i-1}[\{u,v\}]$. In this case, the worst that can happen is that we may have to add a single new wavelength for assigning wavelengths to all the rooted subtrees in the set $\mathcal{Q}_i$. On the other hand, rooted subtrees $\vec{R} \in \mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}], \vec{S} \in \mathcal{P}_{i-1}[\{u,v\}]$ sharing a wavelength means that $|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1}[\{u,w\}] \cup \mathcal{Q}_i)| - |\lambda^{\mathrm{GDY}}(\mathcal{Q}_i \cup (\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]))|$, the number of wavelengths used for assigning wavelengths to all the rooted subtrees in the set $\mathcal{P}_{i-1}[\{u,v\}]$ that are different from the wavelengths used for assigning wavelengths to the rooted subtrees in the set $\mathcal{Q}_i \cup (\mathcal{P}_{i-1}[\{u,w\}] \setminus \mathcal{P}_{i-1}[\{u,v\}])$, also reduces by 1. Applying both the observations, we note that the inequality in (3.17) still holds. $\square$

### 3.2.6 Approximation Ratio

Using the bounds developed in Lemmas 3.5, 3.6, 3.7 and 3.8, we prove the required approximation ratio for GREEDY-WA. We develop the approximation ratio in the form of a parameterized inequality in Lemma 3.9 and then in Lemma 3.10, using the ranges of the parameters, we show that the ratio is bounded by $\frac{5}{2}$.

**Lemma 3.9.** *Given an instance $\{\vec{T}_H, \mathcal{R}\}$ of the MIN-MC-WA-BT problem, where $\vec{T}_H$ is a bidirected tree of degree $\Delta_{\vec{T}_H} \leq 3$ and $\mathcal{R}$ is a set of rooted subtrees on $\vec{T}_H$; the ratio of the number of wavelengths used by the mapping $\lambda^{\mathrm{GDY}}$ generated by GREEDY-WA and the minimum number of wavelengths required for assigning* *wavelengths to all the rooted subtrees in the set $\mathcal{R}$, satisfies*

$$\frac{|\lambda^{\mathrm{GDY}}(\mathcal{R})|}{\min_{\lambda\in\Lambda_{\{\vec{T}_H,\mathcal{R}\}}}|\lambda(\mathcal{R})|} \le \max_{\alpha,\beta,\gamma,\delta,\epsilon} \frac{2+\min\left\{f_1,[f_2-f_3]^+\right\}}{2-\min\{\beta,\gamma\}},$$

*where*

$$f_1 = \alpha - \left[\beta + \frac{\alpha}{2} - 1\right]^+, \qquad f_2 = \delta + \epsilon - \alpha, \qquad f_3 = \left[\delta + \frac{\epsilon}{2} + \gamma - 2\right]^+,$$

*and the maximum is over $\alpha, \beta, \gamma, \delta, \epsilon$ satisfying*

$$0 \le \beta, \gamma \le 1, \qquad 0 \le \delta, \epsilon \le \alpha \le 2, \qquad \delta + \epsilon \le 2.$$

*Proof.* If in the $i$-th round of wavelength assignment, the host tree edge $e_i = \{u, v\} \in E_H$ being processed is of type (i), (ii) or (iii) defined in Lemma 3.3, then according to Lemma 3.5

$$|\lambda^{\mathrm{GDY}}(\mathcal{P}_i)| \le \max\left\{|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})|, 2l_{\{\vec{T}_H,\mathcal{R}\}}\right\}. \tag{3.18}$$

On the other hand, if the edge $e_i = \{u, v\} \in E_H$ being processed in the $i$-th round of wavelength assignment is of type (iv) defined in Lemma 3.3, then according to Lemmas 3.6, 3.7 and 3.8

$$|\lambda^{\mathrm{GDY}}(\mathcal{P}_i)| \le \max\left\{|\lambda^{\mathrm{GDY}}(\mathcal{P}_{i-1})|, 2l_{\{\vec{T}_H,\mathcal{R}\}} + \min\left\{a_i, [g_i - h_i]^+\right\}\right\}, \tag{3.19}$$

where

$$a_i = |\mathcal{Q}_i| - \left(m_{\{\vec{T}_H,\mathcal{R}[\{u,v\}]\}} - m_{\{\vec{T}_H,\mathcal{P}_{i-1}[\{u,v\}]\}}\right), \tag{3.20}$$

and as defined in Lemma 3.8,

$$\begin{aligned} g_i &= |\mathcal{Q}_i[\{u,x\}]| + |\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| - |\mathcal{Q}_i|, \\ h_i &= \left[|\mathcal{Q}_i[\{u,x\}]| + \frac{|\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]|}{2} + m_{\{\vec{T}_H,\mathcal{R}[\{u,x\}]\}} - 2l_{\{\vec{T}_H,\mathcal{R}\}}\right]^+. \end{aligned} \tag{3.21}$$

Here we follow the naming convention of Lemma 3.3, i.e., edge $e_i = \{u, v\}$ is the edge being processed in the $i$-th round of wavelength assignment and edges $\{u, w\}, \{u, x\}$ have the corresponding meanings as defined in Lemma 3.3 whenever $e_i = \{u, v\}$ is of type (iv).

We claim that the number of wavelengths required by GREEDY-WA satisfies

$$|\lambda^{\mathrm{GDY}}(\mathcal{R})| \leq 2l_{\{\vec{T}_H,\mathcal{R}\}} + \max_{e_i \in E_H^{(\mathrm{iv})}} \min\left\{a_i, [g_i - h_i]^+\right\}, \tag{3.22}$$

where $E_H^{(\mathrm{iv})} \subseteq E_H$ is the set of all the host tree edges of type (iv) as defined in Lemma 3.3, encountered in GREEDY-WA. The proof follows from equations (3.18) and (3.19), and a straightforward induction argument.

Also, the minimum number of wavelengths required for assigning wavelengths to all the rooted subtrees in the set $\mathcal{R}$ can be lower bounded as

$$\begin{aligned} \min_{\lambda \in \Lambda_{\{\vec{T}_H,\mathcal{R}\}}} |\lambda(\mathcal{R})| &\geq \max_{\{a,b\} \in E_H} \min_{\lambda \in \Lambda_{\{\vec{T}_H,\mathcal{R}[\{a,b\}]\}}} |\lambda(\mathcal{R}[\{a,b\}])| \\ &= \max_{\{a,b\} \in E_H} \chi_{G_{\mathcal{R}[\{a,b\}]}} \\ &= 2l_{\{\vec{T}_H,\mathcal{R}\}} - \min_{\{a,b\} \in E_H} m_{\{\vec{T}_H,\mathcal{R}[\{a,b\}]\}}. \end{aligned} \tag{3.23}$$

The first inequality simply says that for every host tree edge $\{a, b\} \in E_H$, the minimum number of wavelengths required for assigning wavelengths to all the rooted subtrees in the set $\mathcal{R}[\{a, b\}]$ is less than or equal to the minimum number of wavelengths required for assigning wavelengths to all the rooted subtrees in the set $\mathcal{R}$. This is because for every host tree edge $\{a, b\} \in E_H$, $\mathcal{R}[\{a, b\}]$ is a subset of $\mathcal{R}$. The first equality uses the equivalence of the MIN-MC-WA-BT problem $\{\vec{T}_H, \mathcal{R}[\{a, b\}]\}$ and the problem of minimum vertex coloring of the corresponding conflict graph $G_{\mathcal{R}[\{a,b\}]}$. The final equality is due to the fact that $\bar{G}_{\mathcal{R}[\{a,b\}]}$, the complement of the conflict graph of rooted subtrees on host tree edge $\{a, b\}$, is bipartite with the size of maximum matching being $m_{\{\vec{T}_H,\mathcal{R}[\{a,b\}]\}}$ and the size of the vertex set being $|V_{\bar{G}_{\mathcal{R}[\{a,b\}]}}| = |V_{G_{\mathcal{R}[\{a,b\}]}}| = |\mathcal{R}[\{a, b\}]| = 2l_{\{\vec{T}_H,\mathcal{R}\}}$. Therefore, from equations (3.22) and (3.23) we have

$$\begin{aligned}
\frac{|\lambda^{\mathrm{GDY}}(\mathcal{R})|}{\min_{\lambda\in\Lambda_{\{\vec{T}_H,\mathcal{R}\}}}|\lambda(\mathcal{R})|} &\le \frac{2l_{\{\vec{T}_H,\mathcal{R}\}}+\max_{e_i\in E_H^{(\mathrm{iv})}}\min\left\{a_i,[g_i-h_i]^+\right\}}{2l_{\{\vec{T}_H,\mathcal{R}\}}-\min_{\{a,b\}\in E_H} m_{\{\vec{T}_H,\mathcal{R}[\{a,b\}]\}}}\\
&= \max_{e_i\in E_H^{(\mathrm{iv})}}\left\{\frac{2l_{\{\vec{T}_H,\mathcal{R}\}}+\min\left\{a_i,[g_i-h_i]^+\right\}}{2l_{\{\vec{T}_H,\mathcal{R}\}}-\min_{\{a,b\}\in E_H} m_{\{\vec{T}_H,\mathcal{R}[\{a,b\}]\}}}\right\}\\
&\le \max_{e_i\in E_H^{(\mathrm{iv})}}\left\{\frac{2l_{\{\vec{T}_H,\mathcal{R}\}}+\min\left\{a_i,[g_i-h_i]^+\right\}}{2l_{\{\vec{T}_H,\mathcal{R}\}}-\min\left\{m_{\{\vec{T}_H,\mathcal{R}[\{u,v\}]\}},m_{\{\vec{T}_H,\mathcal{R}[\{u,x\}]\}}\right\}}\right\}.
\end{aligned} \tag{3.24}$$

Observe that for any host tree edge $e_i = \{u,v\}$ of type (iv) as defined in Lemma 3.3, we have the following.

(i) Since $\mathcal{Q}_i \subseteq \mathcal{R}[\{u,v\}]$,

$$|\mathcal{Q}_i| \le |\mathcal{R}[\{u,v\}]| = 2l_{\{\vec{T}_H,\mathcal{R}\}}.$$

Let $|\mathcal{Q}_i| = \alpha_i l_{\{\vec{T}_H,\mathcal{R}\}}$, where $\alpha_i$ is a constant from the set $[0,2]$.

(ii) Since $m_{\{\vec{T}_H,\mathcal{R}[\{u,v\}]\}}$ is the size of maximum matching in graph $\bar{G}_{\mathcal{R}[\{u,v\}]}$,

$$m_{\{\vec{T}_H,\mathcal{R}[\{u,v\}]\}} \le \frac{|V_{\bar{G}_{\mathcal{R}[\{u,v\}]}}|}{2} = \frac{|\mathcal{R}[\{u,v\}]|}{2} = l_{\{\vec{T}_H,\mathcal{R}\}}.$$

Let $m_{\{\vec{T}_H,\mathcal{R}[\{u,v\}]\}} = \beta_i l_{\{\vec{T}_H,\mathcal{R}\}}$, where $\beta_i$ is a constant from the set $[0,1]$.

(iii) $\mathcal{R}[\{u,v\}]$, the set of rooted subtrees present on the edge $\{u,v\}$, can be partitioned into subsets $\mathcal{Q}_i$ and $\mathcal{P}_{i-1}[\{u,v\}]$; therefore

$$|\mathcal{P}_{i-1}[\{u,v\}]| = |\mathcal{R}[\{u,v\}]| - |\mathcal{Q}_i| = (2-\alpha_i)\, l_{\{\vec{T}_H,\mathcal{R}\}}.$$

Since $m_{\{\vec{T}_H,\mathcal{P}_{i-1}[\{u,v\}]\}}$ is the size of maximum matching in graph $\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]}$, we have

$$m_{\{\vec{T}_H,\mathcal{P}_{i-1}[\{u,v\}]\}} \le \frac{|V_{\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]}}|}{2} = \frac{|\mathcal{P}_{i-1}[\{u,v\}]|}{2} = \left(1-\frac{\alpha_i}{2}\right) l_{\{\vec{T}_H,\mathcal{R}\}}.$$

Also, since $\bar{G}_{\mathcal{P}_{i-1}[\{u,v\}]}$ is a subgraph of $\bar{G}_{\mathcal{R}[\{u,v\}]}$, we have

$$m_{\{\vec{T}_H,\mathcal{P}_{i-1}[\{u,v\}]\}} \leq m_{\{\vec{T}_H,\mathcal{R}[\{u,v\}]\}}.$$

The above two inequalities imply that

$$m_{\{\vec{T}_H,\mathcal{R}[\{u,v\}]\}} - m_{\{\vec{T}_H,\mathcal{P}_{i-1}[\{u,v\}]\}} \geq \left[\beta_i + \frac{\alpha_i}{2} - 1\right]^+ l_{\{\vec{T}_H,\mathcal{R}\}}.$$

(iv) Since $\mathcal{Q}_i[\{u,x\}] \subseteq \mathcal{Q}_i$,

$$|\mathcal{Q}_i[\{u,x\}]| \leq |\mathcal{Q}_i| = \alpha_i l_{\{\vec{T}_H,\mathcal{R}\}}.$$

Let $|\mathcal{Q}_i[\{u,x\}]| = \delta_i l_{\{\vec{T}_H,\mathcal{R}\}}$, where $\delta_i$ is a constant from the set $[0,\alpha_i]$.

(v) Note that $\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]$ and $\mathcal{P}_{i-1}[\{u,v\}]$ are non-overlapping subsets of $\mathcal{P}_{i-1}[\{u,w\}] = \mathcal{R}[\{u,w\}]$. Also, the set $\mathcal{R}[\{u,v\}]$ can be partitioned into $\mathcal{Q}_i$ and $\mathcal{P}_{i-1}[\{u,v\}]$. Therefore,

$$\begin{aligned}|\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| &\leq |\mathcal{R}[\{u,w\}]| - |\mathcal{P}_{i-1}[\{u,v\}]| \\ &= |\mathcal{R}[\{u,v\}]| - |\mathcal{P}_{i-1}[\{u,v\}]| \\ &= |\mathcal{Q}_i| = \alpha_i l_{\{\vec{T}_H,\mathcal{R}\}}.\end{aligned}$$

Let $|\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| = \epsilon_i l_{\{\vec{T}_H,\mathcal{R}\}}$, where $\epsilon_i$ is a constant from the set $[0,\alpha_i]$.

(vi) Note that the sets $\mathcal{Q}_i[\{u,x\}]$ and $\mathcal{P}_{i-1}[\{u,x\}]\setminus\mathcal{P}_{i-1}[\{u,v\}]$ are non-overlapping subsets of $\mathcal{R}[\{u,x\}]$. Therefore,

$$|\mathcal{Q}_i[\{u,x\}]| + |\mathcal{P}_{i-1}[\{u,x\}] \setminus \mathcal{P}_{i-1}[\{u,v\}]| \leq |\mathcal{R}[\{u,x\}]|.$$

This implies that $\delta_i + \epsilon_i \leq 2$.

(vii) Since $m_{\{\vec{T}_H,\mathcal{R}[\{u,x\}]\}}$ is the size of maximum matching in graph $\bar{G}_{\mathcal{R}[\{u,x\}]}$,

$$m_{\{\vec{T}_H,\mathcal{R}[\{u,x\}]\}} \leq \frac{|V_{\bar{G}_{\mathcal{R}[\{u,x\}]}}|}{2} = \frac{|\mathcal{R}[\{u,x\}]|}{2} = l_{\{\vec{T}_H,\mathcal{R}\}}.$$

Let $m_{\{\vec{T}_H,\mathcal{R}[\{u,x\}]\}} = \gamma_i l_{\{\vec{T}_H,\mathcal{R}\}}$, where $\gamma_i$ is a constant from the set $[0,1]$.

From (i), (ii) and (iii),

$$a_i \leq \left(\alpha_i - \left[\beta_i + \frac{\alpha_i}{2} - 1\right]^+\right) l_{\{\vec{T}_H, \mathcal{R}\}}. \tag{3.25}$$

From (i), (iv), (v) and (vi),

$$g_i = \left(\delta_i + \frac{\epsilon_i}{2} - \alpha_i\right) l_{\{\vec{T}_H, \mathcal{R}\}}. \tag{3.26}$$

And, from (iv), (v), (vi) and (vii),

$$h_i = \left[\delta_i + \frac{\epsilon_i}{2} + \gamma_i - 2\right]^+ l_{\{\vec{T}_H, \mathcal{R}\}}, \tag{3.27}$$

where $\alpha_i, \beta_i, \gamma_i, \delta_i, \epsilon_i$ are known constants satisfying the following inequalities.

$$0 \leq \beta_i, \gamma_i \leq 1, \qquad 0 \leq \delta_i, \epsilon_i \leq \alpha_i \leq 2, \qquad \delta_i + \epsilon_i \leq 2 \tag{3.28}$$

From equations (3.24), (3.25), (3.26) and (3.27) we obtain

$$\frac{|\lambda^{\mathrm{GDY}}(\mathcal{R})|}{\min_{\lambda \in \Lambda_{\{\vec{T}_H, \mathcal{R}\}}} |\lambda(\mathcal{R})|} \quad \leq \quad \max_{e_i \in E_H^{(\mathrm{iv})}} \left\{ \frac{2 + \min\left\{f_{1_i}, [f_{2_i} - f_{3_i}]^+\right\}}{2 - \min\left\{\beta_i, \gamma_i\right\}} \right\}, \tag{3.29}$$

where

$$f_{1_i} = \alpha_i - \left[\beta_i + \frac{\alpha_i}{2} - 1\right]^+, \qquad f_{2_i} = \delta_i + \epsilon_i - \alpha_i, \qquad f_{3_i} = \left[\delta_i + \frac{\epsilon_i}{2} + \gamma_i - 2\right]^+,$$

and $\alpha_i, \beta_i, \gamma_i, \delta_i, \epsilon_i$ are constants satisfying the inequalities (3.28).

The lemma follows from equation (3.29). □

**Lemma 3.10.** *For any real $\alpha, \beta, \gamma, \delta$ and $\epsilon$ satisfying*

$$0 \leq \beta, \gamma \leq 1, \qquad 0 \leq \delta, \epsilon \leq \alpha \leq 2, \qquad \delta + \epsilon \leq 2,$$

*and functions $f_1, f_2, f_3$ given by*

$$f_1 = \alpha - \left[\beta + \frac{\alpha}{2} - 1\right]^+, \qquad f_2 = \delta + \epsilon - \alpha, \qquad f_3 = \left[\delta + \frac{\epsilon}{2} + \gamma - 2\right]^+,$$

*the following holds*

$$\max_{\alpha,\beta,\gamma,\delta,\epsilon} \frac{2 + \min\left\{f_1, [f_2 - f_3]^+\right\}}{2 - \min\left\{\beta, \gamma\right\}} \leq \frac{5}{2}.$$

*Proof.* Note that for all permissible values of $\alpha, \beta, \gamma, \delta$ and $\epsilon$ we have the following.

$$\begin{aligned}\frac{2+\min\left\{f_1, [f_2-f_3]^+\right\}}{2-\min\{\beta,\gamma\}} &= \min\left\{\frac{2+f_1}{2-\min\{\beta,\gamma\}}, \frac{2+[f_2-f_3]^+}{2-\min\{\beta,\gamma\}}\right\} \\ &\leq \min\left\{\frac{2+f_1}{2-\beta}, \frac{2+[f_2-f_3]^+}{2-\gamma}\right\} \qquad (3.30)\end{aligned}$$

Next we shall prove that, for $0 \leq \alpha \leq 1$,

$$\frac{2+f_1}{2-\beta} \leq \frac{5}{2}, \qquad (3.31)$$

and, for $1 \leq \alpha \leq 2$,

$$\frac{2+[f_2-f_3]^+}{2-\gamma} \leq \frac{5}{2}. \qquad (3.32)$$

From equations (3.30), (3.31), and (3.32) we get the required result.

For equation (3.31), observe that

$$\begin{aligned}\frac{2+f_1}{2-\beta} &= \frac{2+\alpha-\left[\beta+\frac{1}{2}\alpha-1\right]^+}{2-\beta} = \frac{2+\alpha-\max\left\{\beta+\frac{1}{2}\alpha-1, 0\right\}}{2-\beta} \\ &= \frac{\min\left\{3+\frac{1}{2}\alpha-\beta, 2+\alpha\right\}}{2-\beta} \leq \frac{3+\frac{1}{2}\alpha-\beta}{2-\beta} \leq \frac{5}{2},\end{aligned}$$

where the final inequality follows from the assumption that $0 \leq \alpha, \beta \leq 1$.

Next, we prove equation (3.32). Note that if $f_2 \leq f_3$, we have

$$\frac{2+[f_2-f_3]^+}{2-\gamma} = \frac{2}{2-\gamma} \leq 2,$$

where the inequality follows from the assumption that $0 \leq \gamma \leq 1$. Thus, the case of interest is when $f_2 > f_3$. Also, since $f_3 \geq 0$, $f_2 = \delta+\epsilon-\alpha > 0$. Hence, in this case we have

$$\begin{aligned}\frac{2+[f_2-f_3]^+}{2-\gamma} &= \frac{2+\delta+\epsilon-\alpha-\left[\delta+\frac{1}{2}\epsilon+\gamma-2\right]^+}{2-\gamma} \\ &= \frac{2+\delta+\epsilon-\alpha-\max\left\{\delta+\frac{1}{2}\epsilon+\gamma-2, 0\right\}}{2-\gamma} \\ &= \frac{\min\left\{2+\frac{1}{2}\epsilon-\alpha+2-\gamma, 2+\frac{1}{2}\epsilon-\alpha+\delta+\frac{1}{2}\epsilon\right\}}{2-\gamma} \\ &= \frac{2+\frac{1}{2}\epsilon-\alpha}{2-\gamma} + \min\left\{1, \frac{\delta+\frac{1}{2}\epsilon}{2-\gamma}\right\} \\ &\leq \frac{2-\frac{1}{2}\alpha}{2-\gamma} + 1 \leq \frac{5}{2}, \qquad (3.33)\end{aligned}$$

where the first inequality follows from the assumption that $\epsilon \leq \alpha$ and the second inequality follows from the assumptions that $0 \leq \gamma \leq 1$ and $1 \leq \alpha \leq 2$. $\square$

**Theorem 3.11.** *For the restricted MIN-MC-WA-BT problem where the degree of the bidirected tree is at most* 3*, GREEDY-WA is an approximation algorithm with approximation ratio* $\frac{5}{2}$.

*Proof.* The theorem follows from Lemmas 3.9 and 3.10. $\square$

## 3.3 Complexity Analysis

In this section, we prove that the wavelength assignment scheme GREEDY-WA, presented in Section 3.1, has a polynomial running time. In particular, we claim the following result.

**Proposition 3.12.** *For the given instance* $\{\vec{T}_H, \mathcal{R}\}$ *of the MIN-MC-WA-BT problem restricted to the case when the degree of the bidirected tree is at most* 3*, the running time complexity of GREEDY-WA is*

$$O\Big(|E_H|\big(l_{\{\vec{T}_H,\mathcal{R}\}}\big)^{2.5} + |\mathcal{R}|l_{\{\vec{T}_H,\mathcal{R}\}} + |E_H||\mathcal{R}|^2\Big).$$

*Proof.* GREEDY-WA starts off with a BFS of host tree $H$ from some arbitrary root vertex. Complexity of BFS in graph $G$ is $O\big(|V_G| + |E_G|\big)$ [42, p.531-539]. Therefore, for tree $H$, BFS is linear in $|E_H|$. For constructing the conflict graph $G_{\mathcal{R}}$ of the given set of rooted subtrees, we need to decide for every pair of rooted subtrees in the set $\mathcal{R}$, whether the rooted subtrees in that pair collide or not. For each pair we have to check for collision on a maximum of $|E_{\vec{T}_H}| = 2|E_H|$ directed edges. Therefore, the conflict graph can be constructed in $O\big(|E_H||\mathcal{R}|^2\big)$ time.

First consider the case when the host tree edge $e_i = \{u, v\} \in E_H$ being processed in the $i$-th round of wavelength assignment is of type (i), (ii) or (iii) as defined in Lemma 3.3. In order to assign wavelength to rooted subtree $\vec{R} \in \mathcal{Q}_i$, we first determine the set of unavailable wavelengths for $\vec{R}$. This is the set of wavelengths that have already been assigned to (either in the first $i-1$ rounds or in the $i$-th round itself) any rooted subtree that collides with $\vec{R}$. Using Lemma 3.4, we can upper bound the size of this set by $|\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i| = |\mathcal{R}[\{u,v\}]| = 2l_{\{\vec{T}_H,\mathcal{R}\}}$. Rooted subtree $\vec{R}$ is greedily assigned the first wavelength that is not in this set of unavailable wavelengths. This shows that $\vec{R}$ is assigned wavelength in $O\big(l_{\{\vec{T}_H,\mathcal{R}\}}\big)$ time.

Next consider the case when the host tree edge $e_i = \{u,v\} \in E_H$ being processed in the $i$-th round of wavelength assignment is of type (iv) as defined in Lemma 3.3. In this case GREEDY-WA calls subroutines PROCESS-EDGE-1 and PROCESS-EDGE-2. Note that since $|\mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i| = 2l_{\{\vec{T}_H,\mathcal{R}\}}$, in PROCESS-EDGE-1 initializing $B_1$ as the complementary bipartite graph $G_{\mathcal{P}_{i-1}[\{u,v\}]\cup\mathcal{Q}_i}$ takes $O\Big(\big(l_{\{\vec{T}_H,\mathcal{R}\}}\big)^2\Big)$ time. Between every pair of independent vertices in $B_1$, we decide whether to introduce an edge or not. Let rooted subtrees $\vec{R}, \vec{S} \in \mathcal{P}_{i-1}[\{u,v\}] \cup \mathcal{Q}_i$ be a pair of independent vertices in $B_1$. If $\vec{R}, \vec{S}$ are both unassigned or assigned with the same wavelength, then no edge is added. On the other hand, if $\vec{R}, \vec{S}$ are assigned with different wavelengths, then the new edge $\{\vec{R}, \vec{S}\}$ is added in $B_1$. Clearly these are constant time checks. The interesting case is when $\vec{R}$ is unassigned whereas some wavelength has already been assigned to $\vec{S}$. In this case we check if there is some rooted subtree $\vec{U}$ that has already been assigned a wavelength which it shares with $\vec{S}$, and it collides with $\vec{R}$. If there is such a rooted subtree, then we add the new edge $\{\vec{R}, \vec{S}\}$ in $B_1$. To perform this check in constant time, for each processed host tree edge we track the pairs of rooted subtrees that share wavelengths. Note that due to Lemma 3.1, more than two rooted subtrees present on a host tree edge can not share wavelengths. Also, from Lemma 3.4 we can infer that if there is a rooted subtree $\vec{U}$ which shares wavelength with $\vec{S}$ and collides with $\vec{R}$, then it must be present on edge $\{u,w\}$ (as defined in Lemma 3.3, $\{u,w\}$ is the host tree edge adjacent to $u$ that has already been processed). Since $\vec{S}, \vec{U}$ form a pair of rooted subtrees present on host tree edge $\{u, w\}$ that share wavelength, the pair is tracked. So we can simply check (in constant time) if $\vec{R}$ collides with the rooted subtree (if present) that shares its wavelength with $\vec{S}$ on the host tree edge $\{u, w\}$. This determines whether we have to add the new edge $\{\vec{R}, \vec{S}\}$ in $B_1$ or not. Since the number of pairs of independent vertices in $B_1$ is upper bounded by $\left(l_{\{\vec{T}_H, \mathcal{R}\}}\right)^2$, graph $B_1$ is updated in $O\Big(\big(l_{\{\vec{T}_H, \mathcal{R}\}}\big)^2\Big)$ time. After this, $\bar{B}_1$ can also be obtained from $B_1$ in $O\Big(\big(l_{\{\vec{T}_H, \mathcal{R}\}}\big)^2\Big)$ time. Complexity of determining a maximum matching in bipartite graph $B$ is $O\big(\sqrt{|V_B|}|E_B|\big)$ [42, p.696-697]. Therefore, in bipartite graph $\bar{B}_1$ having $2l_{\{\vec{T}_H, \mathcal{R}\}}$ vertices, determining a maximum matching requires $O\Big(\big(l_{\{\vec{T}_H, \mathcal{R}\}}\big)^{2.5}\Big)$ time. If an unassigned rooted subtree is matched to a rooted subtree that has already been assigned some wavelength, the wavelength assignment for that unassigned rooted subtree is a constant time operation. On the other hand, for unmatched unassigned rooted subtrees and matched pairs of unassigned rooted subtrees, as explained in the previous paragraph, wavelength assignment is carried out in $O\big(l_{\{\vec{T}_H, \mathcal{R}\}}\big)$ time. Similar time complexities hold for various steps of PROCESS-EDGE-2. Determining the better of the two subroutines and assigning wavelength to an unassigned rooted subtree $\vec{R} \in \mathcal{Q}_i$ is a constant time operation.

To summarize, the running time complexity of GREEDY-WA depends on the following steps.

(i) Constructing the conflict graph $G_{\mathcal{R}}$ requires $O\big(|E_H||\mathcal{R}|^2\big)$ time.

(ii) Determining a maximum matching in bipartite graphs $\bar{B}_1$ and $\bar{B}_2$ requires $O\Big(\big(l_{\{\vec{T}_H, \mathcal{R}\}}\big)^{2.5}\Big)$ time. This is done for all host tree edges of type (iv). Since there are $O\big(|E_H|\big)$ such edges, the total time required for determining maximum matchings is $O\Big(|E_H|\big(l_{\{\vec{T}_H, \mathcal{R}\}}\big)^{2.5}\Big)$.

(iii) Assigning wavelengths to rooted subtrees is either a constant time or a $O\big(l_{\{\vec{T}_H, \mathcal{R}\}}\big)$

operation. Since there are $|\mathcal{R}|$ rooted subtrees, total time required for assigning wavelengths is $O\big(|\mathcal{R}| l_{\{\vec{T}_H, \mathcal{R}\}}\big)$.

This gives us the required time complexity for GREEDY-WA. $\square$

# Chapter 4

## Subtree Based Multicast Wavelength Assignment in Bidirected Trees

In this chapter, we present another, simpler scheme for a restricted version of the MIN-MC-WA-BT problem described in Section 2.2. The additional restriction that we place on the problem is to limit the degree of the bidirected tree to be at most 4. In other words, the problem under consideration is the MIN-MC-WA-BT problem represented as a pair $\{\vec{T}_H, \mathcal{R}\}$, where $\vec{T}_H$ is a bidirected tree with degree $\Delta_{\vec{T}_H} \leq 4$ and $\mathcal{R}$ is a set of rooted subtrees on $\vec{T}_H$. We prove that the presented scheme is a $\frac{10}{3}$-approximation algorithm when $\Delta_{\vec{T}_H} = 4$, a 3-approximation algorithm when $\Delta_{\vec{T}_H} = 3$ and a 2-approximation algorithm when $\Delta_{\vec{T}_H} = 2$.

### 4.1 Subtree Based Wavelength Assignment

Let $\mathcal{U}_\mathcal{R}$ denote the set of subtrees of host tree $H$ obtained by taking the skeleton graphs of all the rooted subtrees in the set $\mathcal{R}$, i.e., if $\mathcal{R} = \{\vec{R}_1, \vec{R}_2, \ldots, \vec{R}_{|\mathcal{R}|}\}$, then $\mathcal{U}_\mathcal{R} = \{U_1, U_2 \ldots, U_{|\mathcal{R}|}\}$, where $U_i := \|\vec{R}_i\|$ for every $\vec{R}_i \in \mathcal{R}$. Consider the conflict graph $G_{\mathcal{U}_\mathcal{R}}$ corresponding to the set of subtrees $\mathcal{U}_\mathcal{R}$, defined to be the intersection graph of the family of the sets of edges of the subtrees. In other words, for any pair of subtrees $U_i, U_j \in \mathcal{U}_\mathcal{R}$, there is an edge $\{U_i, U_j\} \in E_{G_{\mathcal{U}_\mathcal{R}}}$ in the conflict graph if and only if they share some common host tree edge, i.e., $E_{U_i} \cap E_{U_j} \neq \emptyset$.

The basic idea is that instead of solving the MIN-MC-WA-BT problem instance $\{\vec{T}_H, \mathcal{R}\}$, which is hard, we color the conflict graph $G_{\mathcal{U}_\mathcal{R}}$ and then use this coloring to generate a wavelength assignment for the original set of rooted subtrees.

**Lemma 4.1.** *$G_\mathcal{R}$ is a spanning subgraph of $G_{\mathcal{U}_\mathcal{R}}$.*

*Proof.* By definition of the set $\mathcal{U}_\mathcal{R}$ there is an obvious bijection between $V_{G_{\mathcal{U}_\mathcal{R}}}$ and $V_{G_\mathcal{R}}$. Also, for every edge $\{\vec{R}_i, \vec{R}_j\} \in E_{G_\mathcal{R}}$ of the conflict graph $G_\mathcal{R}$, there is a corresponding edge $\{U_i, U_j\} \in E_{G_{\mathcal{U}_\mathcal{R}}}$ in the conflict graph $G_{\mathcal{U}_\mathcal{R}}$. This is because the presence of edge $\{\vec{R}_i, \vec{R}_j\} \in E_{G_\mathcal{R}}$ implies that the rooted subtrees $\vec{R}_i$ and $\vec{R}_j$ share some common directed edge $(u, v) \in E_{\vec{T}_H}$ of the bidirected tree $\vec{T}_H$, i.e, $(u, v) \in E_{\vec{R}_i} \cap E_{\vec{R}_j}$. In that case, the host tree edge $\{u, v\} \in E_H$ is shared by the corresponding subtrees $U_i$ and $U_j$, i.e., $\{u, v\} \in E_{U_i} \cap E_{U_j}$. □

Lemma 4.1 results in the following corollary.

**Corollary 4.2.** *Any vertex coloring for the conflict graph $G_{\mathcal{U}_\mathcal{R}}$ determines a vertex coloring for the conflict graph $G_\mathcal{R}$. Consequently it determines a wavelength assignment for the corresponding MIN-MC-WA-BT problem instance $\{\vec{T}_H, \mathcal{R}\}$.*

Corollary 4.2 suggests that we can simply color the conflict graph $G_{\mathcal{U}_\mathcal{R}}$ of the skeleton subtrees of the rooted subtrees in the set $\mathcal{R}$, and then assign each rooted subtree $\vec{R}_i \in \mathcal{R}$ the wavelength corresponding to the color determined for its skeleton subtree $U_i$. This is essentially the scheme that we follow.

Observe that if the host tree degree $\Delta_H = 2$, then the graph $G_{\mathcal{U}_\mathcal{R}}$ is simply an interval graph [41, p.175]. Moreover as stated in Section 2.3, if the host tree degree $\Delta_H = 3$, then the graph $G_{\mathcal{U}_\mathcal{R}}$ is chordal, and if the host tree degree $\Delta_H = 4$, then the graph $G_{\mathcal{U}_\mathcal{R}}$ is weakly chordal. In all three cases, the graph is easily colorable. Since the degree of a bidirected tree is defined to be equal to the degree of its host tree, the characterization of the conflict graph $G_{\mathcal{U}_\mathcal{R}}$ based on the degree of the bidirected tree $\vec{T}_H$ is exactly the same as the characterization based on the degree of the host tree $H$.

The complete scheme is given as Algorithm 5 (SUBTREE-BASED-WA). We denote the wavelength assignment generated by the scheme by $\lambda^{\text{SUB}}$.

**Subroutine 5** SUBTREE-BASED-WA

**Require:** MIN-MC-WA-BT problem instance $\{\vec{T}_H, \mathcal{R}\}$, where $\Delta_{\vec{T}_H} \leq 4$.

**Ensure:** A wavelength assignment $\lambda^{\text{SUB}} \in \Lambda_{\{\vec{T}_H, \mathcal{R}\}}$.

1: Determine $\mathcal{U}_{\mathcal{R}} = \{U_1, U_2, \ldots, U_{|\mathcal{R}|}\}$ where $U_i := \|\vec{R}_i\|$ for every $\vec{R}_i \in \mathcal{R}$.

2: Determine the conflict graph $G_{\mathcal{U}_{\mathcal{R}}}$.

3: Determine a minimum vertex coloring $\psi^*$ for the conflict graph $G_{\mathcal{U}_{\mathcal{R}}}$.

{This is easy since the conflict graph is an interval graph, chordal graph or weakly chordal graph depending on whether the degree of the bidirected tree is 2, 3 or 4.}

4: $\lambda(\vec{R}_i) \leftarrow \psi^*(U_i)$ for every $\vec{R}_i \in \mathcal{R}$

## 4.2 Approximation Analysis

In this section, we shall prove that SUBTREE-BASED-WA is an approximation algorithm for the problem. We shall first discuss the case when the degree of the bidirected tree, and hence the host tree, is equal to 4, i.e., $\Delta_H = 4$. The other two cases when the degree of the bidirected tree, and hence the host tree, is equal to $2, 3$, i.e., $\Delta_H = 2, 3$, are similar.

We start our analysis by proving a pair of useful results that characterize the subtrees in the set $\mathcal{U}_{\mathcal{R}}$ based on the structure of the conflict graph $G_{\mathcal{U}_{\mathcal{R}}}$. Both of these results are independent of the degree of the host tree $H$. In Lemma 4.3 we prove that in the conflict graph $G_{\mathcal{U}_{\mathcal{R}}}$, all the subtrees forming a clique must have at least one host tree vertex in common. And in Lemma 4.4 we prove that if two subtrees of a tree contain a common edge, then they must contain at least one common edge adjacent to their every common vertex.

We shall see that Lemma 4.3 allows us to determine the size of maximum clique in the conflict graph $G_{\mathcal{U}_\mathcal{R}}$ by studying the sets of subtrees containing a common host tree vertex one at a time, rather than studying the set of all the subtrees at once. For each such set of subtrees, Lemma 4.4 allows us to concentrate only on the conflicts on the host tree edges adjacent to the common host tree vertex among the subtrees in the set, and ignore the presence or absence of the subtrees on all the other host tree edges. We require the size of maximum clique in conflict graph $G_{\mathcal{U}_\mathcal{R}}$ to determine the chromatic number of the graph, which in turn is needed to determine the approximation ratio for SUBTREE-BASED-WA.

**Lemma 4.3.** *If subtrees $U_{i_1}, \ldots, U_{i_k} \in \mathcal{U}_\mathcal{R}$ form a clique of size $k$ in the conflict graph $G_{\mathcal{U}_\mathcal{R}}$, then there is a host tree vertex $v \in V_H$ common to all these subtrees, i.e., $v \in \bigcap_{j=1}^{k} V_{U_{i_j}}$.*

*Proof.* We prove by induction.

For the case when $k = 2$, the lemma effectively states that if there is an edge $\{U_{i_1}, U_{i_2}\} \in E_{G_{\mathcal{U}_\mathcal{R}}}$, then for the corresponding subtrees $V_{U_{i_1}} \cap V_{U_{i_2}} \neq \emptyset$. By the definition of conflict graph, the existence of edge $\{U_{i_1}, U_{i_2}\} \in E_{G_{\mathcal{U}_\mathcal{R}}}$ implies that there is at least one common edge in the corresponding subtrees, i.e., $E_{U_{i_1}} \cap E_{U_{i_2}} \neq \emptyset$, which in turn implies that $V_{U_{i_1}} \cap V_{U_{i_2}} \neq \emptyset$. Hence, the statement holds for $k = 2$.

Let it hold for $k = m$, i.e., if subtrees $U_{i_1}, \ldots, U_{i_m} \in \mathcal{U}_\mathcal{R}$ form a clique of size $m$ in the conflict graph $G_{\mathcal{U}_\mathcal{R}}$, then there is a host tree vertex $v \in V_H$ common to all these subtrees, i.e., $v \in \bigcap_{j=1}^{m} V_{U_{i_j}}$.

Next we consider the case when $k = m + 1$. Let the set of subtrees $\mathcal{C} = \{U_{i_1}, \ldots, U_{i_{m+1}}\} \subseteq \mathcal{U}_\mathcal{R}$ form a clique of size $m + 1$ in the conflict graph $G_{\mathcal{U}_\mathcal{R}}$. Let $\mathcal{C}_j := \mathcal{C} \setminus \{U_{i_j}\}$ for $j \in \{1, \ldots, m+1\}$. For every $j$, $\mathcal{C}_j$ forms a clique of size $m$ in the conflict graph $G_{\mathcal{U}_\mathcal{R}}$. By inductive assumption, there is a host tree vertex common to all the subtrees in the clique $\mathcal{C}_j$. Let $v_j \in V_H$ be a host tree vertex that is common to all the subtrees in the clique $\mathcal{C}_j$. Note that if $v_l \in V_{U_{i_l}}$ for some $l \in \{1, \ldots, m + 1\}$, then this means $v_l \in \bigcap_{j=1}^{m+1} V_{U_{i_j}}$ and hence the statement of the lemma holds for

$k = m + 1$. Let us assume the alternative case, i.e., for every $j$, $v_j \notin V_{U_{i_j}}$. Consider the host tree vertices $v_1, v_l, v_{m+1}$ where $1 < l < m + 1$. Since $U_{i_l}$ lies in the cliques $\mathcal{C}_1$ and $\mathcal{C}_{m+1}$; $v_1, v_{m+1} \in V_{U_{i_l}}$. Also, by assumption, $v_l \notin V_{U_{i_l}}$. Therefore, there is a path in the host tree $H$ (using edges from the set $E_{U_{i_l}}$) between vertices $v_1, v_{m+1}$ that does not contain vertex $v_l$. Using similar arguments we can find a path between vertices $v_1, v_l$ not containing vertex $v_{m+1}$ and a path between vertices $v_l, v_{m+1}$ not containing vertex $v_1$. This shows the presence of a cycle in the host tree $H$, which is a contradiction. Hence, the statement of the lemma holds for $k = m + 1$. □

For any bidirected tree vertex $v \in V_{\vec{T}_H}$, let us define $\mathcal{R}[v]$ to be the set of rooted subtrees that contain $v$, i.e., $\mathcal{R}[v] := \{\vec{R} \in \mathcal{R} : v \in V_{\vec{R}}\}$. Hence, for any host tree vertex $v \in V_H$, $\mathcal{U}_{\mathcal{R}[v]} := \{U \in \mathcal{U} : v \in V_U\}$. An immediate implication of Lemma 4.3 is that the size of maximum clique in the conflict graph $G_{\mathcal{U}_\mathcal{R}}$ is equal to the largest of the size of maximum cliques in the conflict graphs of subtrees containing various host tree vertices, i.e.,

$$\omega_{G_{\mathcal{U}_\mathcal{R}}} = \max_{v \in V_H} \omega_{G_{\mathcal{U}_{\mathcal{R}[v]}}}, \tag{4.1}$$

where $\omega_{G_{\mathcal{U}_\mathcal{R}}}$ denotes the clique number of the graph $G_{\mathcal{U}_\mathcal{R}}$, and $\mathcal{U}_{\mathcal{R}[v]}$ denotes the set of subtrees that contain host tree vertex $v \in V_H$.

**Lemma 4.4.** *If subtrees $U_i, U_j \in \mathcal{U}_{\mathcal{R}[v]}$ share some host tree edge, then they must share at least one host tree edge adjacent to the host tree vertex $v \in V_H$.*

*Proof.* Subtrees $U_i, U_j \in \mathcal{U}_{\mathcal{R}[v]}$ imply that host tree vertex $v \in V_H$ lies in both the vertex sets $V_{U_i}$ and $V_{U_j}$. Let subtrees $U_i, U_j$ share some host tree edge that is not adjacent to $v$. Let one of its end vertices be $w$. Therefore, host tree vertex $w$ lies in both the vertex sets $V_{U_i}$ and $V_{U_j}$. Since vertices $v, w \in V_{U_i}$ and $U_i$ is a subtree of the host tree $H$, all the host tree edges on the path between vertices $v, w$ are in the set $E_{U_i}$. Let $\{u, v\} \in E_H$ be the first edge on the path starting from vertex $v$.

Therefore, host tree edge $\{u, v\} \in E_{U_i}$. Following similar arguments we can show that host tree $\{u, v\} \in E_{U_j}$ as well. $\square$

One of the implications of Lemma 4.4 is that if two subtrees $U_i, U_j \in \mathcal{U}_{\mathcal{R}[v]}$ do not share any host tree edge adjacent to vertex $v$, then there is no edge between the two subtrees in the conflict graph $G_{\mathcal{U}_\mathcal{R}}$, i.e., $\{U_i, U_j\} \notin E_{G_{\mathcal{U}_\mathcal{R}}}$.

After having established Lemmas 4.3 and 4.4, we try to study the sets of subtrees containing a common host tree vertex in more detail. Consider a host tree vertex $v \in V_H$. Two subtrees $U_i, U_j \in \mathcal{U}_{\mathcal{R}[v]}$ are said to be *equivalent* (with respect to $v$) if there is no host tree edge adjacent to $v$ such that $U_i$ is present on the edge but $U_j$ is not, and vice versa. For any host tree vertex $v \in V_H$, we can partition $\mathcal{U}_{\mathcal{R}[v]}$, the set of subtrees that contain $v$, into equivalence classes based on their presence or absence on the tree edges adjacent to vertex $v$. In the case when the degree of the host tree is $\Delta_H = 4$, for any host tree vertex $v \in V_H$, there are 15 such equivalence classes. Let these be $\mathcal{U}^1_{\mathcal{R}[v]}, \ldots, \mathcal{U}^{15}_{\mathcal{R}[v]}$. Figure 4.1 shows a sample subtree from each of these classes in the neighborhood of vertex $v$. In the figure, vertex $v$ is depicted as black dot. Note that there are host tree vertices for which some of the equivalence classes may be empty, e.g. for a vertex $v \in V_H$ having degree $\delta_H(v) < 4$.

Next, in Lemmas 4.5 and 4.6, we shall determine an upper bound on the size of maximum clique in the conflict graph. Lemma 4.5 is another useful result pertaining to the cliques in conflict graph $G_{\mathcal{U}_\mathcal{R}}$, and is independent of the degree of host tree $H$. Finally, in Lemma 4.6 we specifically look at the maximal cliques in the conflict graphs of subtrees of host tree $H$ of degree $\Delta_H = 4$.

**Lemma 4.5.** *For some host tree vertex $v \in V_H$, let the set of rooted subtrees $\mathcal{C} \subseteq \mathcal{U}_{\mathcal{R}[v]}$ form a clique of size $k$ in the conflict graph $G_{\mathcal{U}_\mathcal{R}}$. If there are two equivalent subtrees $U_i, U_j \in \mathcal{U}_{\mathcal{R}[v]}$ such that $U_i \in \mathcal{C}$ but $U_j \notin \mathcal{C}$, then the vertex set $\mathcal{C} \cup \{U_j\}$ forms a clique of size $k+1$ in the conflict graph $G_{\mathcal{U}_\mathcal{R}}$.*

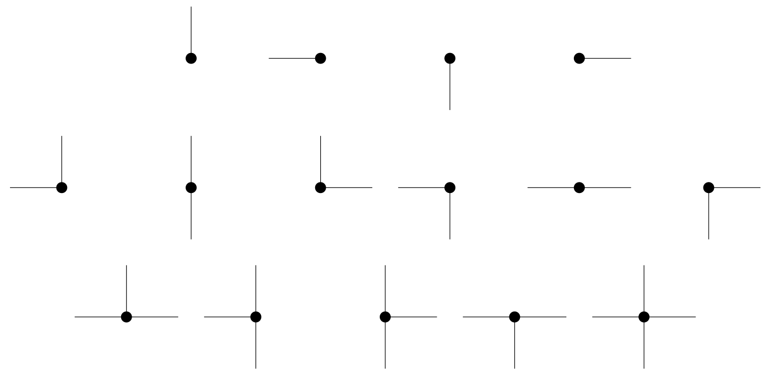


Figure 4.1: Partition of $\mathcal{U}_{\mathcal{R}[v]}$, the set of subtrees of host tree $H$ with degree $\Delta_H = 4$ containing vertex $v \in V_H$, into 15 equivalence classes: $\mathcal{U}^1_{\mathcal{R}[v]}, \ldots, \mathcal{U}^{15}_{\mathcal{R}[v]}$.

*Proof.* Note that if subtree $U_i \in \mathcal{U}_{\mathcal{R}[v]}$ then it must be present on at least one of the host tree edges adjacent to $v$. This is simply because we assume that there are at least two vertices in $U_i$, i.e., $|V_{U_i}| \geq 2$. The reason for this assumption is that if the subtree $U_i$ is singleton, then the corresponding rooted subtree $\vec{R}_i$ is also singleton, which is not possible since $\vec{R}_i$ models some multicast traffic request with a source and at least one destination node. Since $U_i$ is a subtree and therefore connected, the host tree edges on the paths from $v$ to every other vertex in the set $V_{U_i}$ must belong to the set $E_{U_i}$. At least one of these paths must necessarily contain some host tree edge adjacent to $v$. With this observation in mind, we begin the proof of the lemma.

As explained above, since subtrees $U_i, U_j \in \mathcal{U}_{\mathcal{R}[v]}$ are equivalent, they share at least one host tree edge (adjacent to $v$). Therefore, there is an edge in the conflict graph between subtrees $U_i, U_j$, i.e., $\{U_i, U_j\} \in E_{G_{\mathcal{U}_\mathcal{R}}}$. For every subtree $U_l \in \mathcal{C} \setminus \{U_i\}$, since the edge $\{U_i, U_l\} \in E_{G_{\mathcal{U}_\mathcal{R}}}$, by Lemma 4.4, subtrees $U_i, U_l$ share some host tree edge adjacent to vertex $v$. Also, since subtrees $U_i, U_j$ are equivalent (w.r.t. $v$), every host tree edge adjacent to vertex $v$ is either in both the sets $E_{U_i}, E_{U_j}$, or is in neither of the two. Therefore, for every subtree $U_l \in \mathcal{C}$, the edge $\{U_j, U_l\}$ exists in the conflict graph $G_{\mathcal{U}_\mathcal{R}}$. This proves that the subtree set $\mathcal{C} \cup \{U_j\}$ forms a clique of

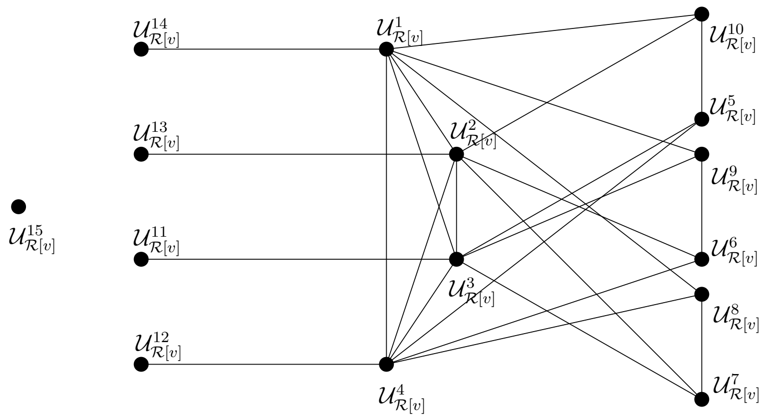



Figure 4.2: Structure of the complementary conflict graph $\bar{G}_{\mathcal{U}_{\mathcal{R}[v]}}$ in the case when the degree of the host tree is $\Delta_H = 4$.

size $k+1$ in the conflict graph. $\square$

An immediate implication of Lemma 4.5 is that if the subtree set $\mathcal{C} \subseteq \mathcal{U}_{\mathcal{R}[v]}$ forms a maximal clique in $G_{\mathcal{U}_{\mathcal{R}[v]}}$, then for every equivalence class $\mathcal{U}^l_{\mathcal{R}[v]}$ of the subtree set $\mathcal{U}_{\mathcal{R}[v]}$, exactly one of the following holds:

(i) Every subtree in the equivalence class is in the maximal clique, i.e, $\mathcal{U}^l_{\mathcal{R}[v]} \subseteq \mathcal{C}$.

(ii) None of the subtrees in the equivalence class is in the maximal clique, i.e, $\mathcal{U}^l_{\mathcal{R}[v]} \cap \mathcal{C} = \emptyset$.

Using this observation we determine an upper bound on the size of maximum clique in the conflict graph $G_{\mathcal{U}_{\mathcal{R}[v]}}$.

**Lemma 4.6.** *Given a bidirected tree $\vec{T}_H$ having degree $\Delta_{\vec{T}_H} = 4$, and a set $\mathcal{R}$ of rooted subtrees of $\vec{T}_H$. The size of maximum clique in the conflict graph $G_{\mathcal{U}_{\mathcal{R}[v]}}$ is bounded as $\omega_{G_{\mathcal{U}_{\mathcal{R}}}} \leq \frac{10}{3} l_{\{\vec{T}_H, \mathcal{R}\}}$, where $\mathcal{U}_{\mathcal{R}}$ is the set of skeletons of the rooted trees in the set $\mathcal{R}$ as defined in Section 4.1, and $l_{\{\vec{T}_H, \mathcal{R}\}}$ is the load of the set $\mathcal{R}$ of rooted subtrees on the bidirected tree $\vec{T}_H$ as defined in Section 3.2.*

*Proof.* Using Lemmas 4.4 and 4.5, we can determine the maximal cliques in the conflict graph $G_{\mathcal{U}_{\mathcal{R}[v]}}$. It turns out that it is much easier to observe the maximal independent sets in the complementary conflict graph $\bar{G}_{\mathcal{U}_{\mathcal{R}[v]}}$. These are exactly the same as the maximal cliques in conflict graph $G_{\mathcal{U}_{\mathcal{R}[v]}}$. Figure 4.2 depicts the structure of complementary conflict graph $\bar{G}_{\mathcal{U}_{\mathcal{R}[v]}}$. Each vertex in the figure represents a set of independent subtrees in $\bar{G}_{\mathcal{U}_{\mathcal{R}[v]}}$. And, an edge in the figure represents an edge between every subtree in one set and every subtree in the other set.

We observe that the only possible maximal cliques in the conflict graph $G_{\mathcal{U}_{\mathcal{R}[v]}}$ comprise of the subtrees in the following equivalence classes.

(i) $\mathcal{U}^1_{\mathcal{R}[v]}, \mathcal{U}^5_{\mathcal{R}[v]}, \mathcal{U}^6_{\mathcal{R}[v]}, \mathcal{U}^7_{\mathcal{R}[v]}, \mathcal{U}^{11}_{\mathcal{R}[v]}, \mathcal{U}^{12}_{\mathcal{R}[v]}, \mathcal{U}^{13}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

(ii) $\mathcal{U}^2_{\mathcal{R}[v]}, \mathcal{U}^5_{\mathcal{R}[v]}, \mathcal{U}^8_{\mathcal{R}[v]}, \mathcal{U}^9_{\mathcal{R}[v]}, \mathcal{U}^{11}_{\mathcal{R}[v]}, \mathcal{U}^{12}_{\mathcal{R}[v]}, \mathcal{U}^{14}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

(iii) $\mathcal{U}^3_{\mathcal{R}[v]}, \mathcal{U}^6_{\mathcal{R}[v]}, \mathcal{U}^8_{\mathcal{R}[v]}, \mathcal{U}^{10}_{\mathcal{R}[v]}, \mathcal{U}^{12}_{\mathcal{R}[v]}, \mathcal{U}^{13}_{\mathcal{R}[v]}, \mathcal{U}^{14}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

(iv) $\mathcal{U}^4_{\mathcal{R}[v]}, \mathcal{U}^7_{\mathcal{R}[v]}, \mathcal{U}^9_{\mathcal{R}[v]}, \mathcal{U}^{10}_{\mathcal{R}[v]}, \mathcal{U}^{11}_{\mathcal{R}[v]}, \mathcal{U}^{13}_{\mathcal{R}[v]}, \mathcal{U}^{14}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

(v) $\mathcal{U}^5_{\mathcal{R}[v]}, \mathcal{U}^6_{\mathcal{R}[v]}, \mathcal{U}^7_{\mathcal{R}[v]}, \mathcal{U}^{11}_{\mathcal{R}[v]}, \mathcal{U}^{12}_{\mathcal{R}[v]}, \mathcal{U}^{13}_{\mathcal{R}[v]}, \mathcal{U}^{14}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

(vi) $\mathcal{U}^5_{\mathcal{R}[v]}, \mathcal{U}^8_{\mathcal{R}[v]}, \mathcal{U}^9_{\mathcal{R}[v]}, \mathcal{U}^{11}_{\mathcal{R}[v]}, \mathcal{U}^{12}_{\mathcal{R}[v]}, \mathcal{U}^{13}_{\mathcal{R}[v]}, \mathcal{U}^{14}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

(vii) $\mathcal{U}^6_{\mathcal{R}[v]}, \mathcal{U}^8_{\mathcal{R}[v]}, \mathcal{U}^{10}_{\mathcal{R}[v]}, \mathcal{U}^{11}_{\mathcal{R}[v]}, \mathcal{U}^{12}_{\mathcal{R}[v]}, \mathcal{U}^{13}_{\mathcal{R}[v]}, \mathcal{U}^{14}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

(viii) $\mathcal{U}^7_{\mathcal{R}[v]}, \mathcal{U}^9_{\mathcal{R}[v]}, \mathcal{U}^{10}_{\mathcal{R}[v]}, \mathcal{U}^{11}_{\mathcal{R}[v]}, \mathcal{U}^{12}_{\mathcal{R}[v]}, \mathcal{U}^{13}_{\mathcal{R}[v]}, \mathcal{U}^{14}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

(ix) $\mathcal{U}^5_{\mathcal{R}[v]}, \mathcal{U}^7_{\mathcal{R}[v]}, \mathcal{U}^9_{\mathcal{R}[v]}, \mathcal{U}^{11}_{\mathcal{R}[v]}, \mathcal{U}^{12}_{\mathcal{R}[v]}, \mathcal{U}^{13}_{\mathcal{R}[v]}, \mathcal{U}^{14}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

(x) $\mathcal{U}^5_{\mathcal{R}[v]}, \mathcal{U}^6_{\mathcal{R}[v]}, \mathcal{U}^8_{\mathcal{R}[v]}, \mathcal{U}^{11}_{\mathcal{R}[v]}, \mathcal{U}^{12}_{\mathcal{R}[v]}, \mathcal{U}^{13}_{\mathcal{R}[v]}, \mathcal{U}^{14}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

(xi) $\mathcal{U}^6_{\mathcal{R}[v]}, \mathcal{U}^7_{\mathcal{R}[v]}, \mathcal{U}^{10}_{\mathcal{R}[v]}, \mathcal{U}^{11}_{\mathcal{R}[v]}, \mathcal{U}^{12}_{\mathcal{R}[v]}, \mathcal{U}^{13}_{\mathcal{R}[v]}, \mathcal{U}^{14}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

(xii) $\mathcal{U}^8_{\mathcal{R}[v]}, \mathcal{U}^9_{\mathcal{R}[v]}, \mathcal{U}^{10}_{\mathcal{R}[v]}, \mathcal{U}^{11}_{\mathcal{R}[v]}, \mathcal{U}^{12}_{\mathcal{R}[v]}, \mathcal{U}^{13}_{\mathcal{R}[v]}, \mathcal{U}^{14}_{\mathcal{R}[v]}, \mathcal{U}^{15}_{\mathcal{R}[v]}$

According to our notation, the load of the set $\mathcal{R}$ of rooted subtrees on the bidirected tree $\vec{T}_H$ is $l_{\{\vec{T}_H, \mathcal{R}\}}$. Therefore, the number of subtrees present on any host tree edge is upper bounded by $2l_{\{\vec{T}_H,\mathcal{R}\}}$. For any host tree vertex $v \in V_H$, this leads to the following inequalities:

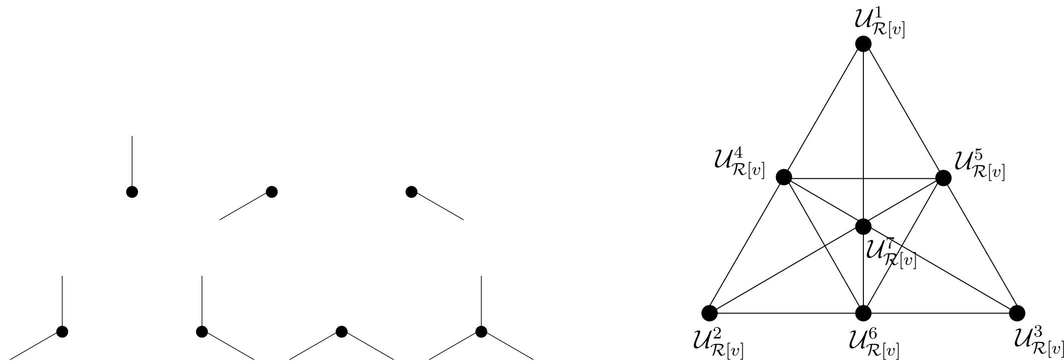
$$
\begin{aligned}
&|\mathcal{U}^1_{\mathcal{R}[v]}| + |\mathcal{U}^5_{\mathcal{R}[v]}| + |\mathcal{U}^6_{\mathcal{R}[v]}| + |\mathcal{U}^7_{\mathcal{R}[v]}| \\
&+ |\mathcal{U}^{11}_{\mathcal{R}[v]}| + |\mathcal{U}^{12}_{\mathcal{R}[v]}| + |\mathcal{U}^{13}_{\mathcal{R}[v]}| + |\mathcal{U}^{15}_{\mathcal{R}[v]}| \leq 2l_{\{\vec{T}_H,\mathcal{R}\}} \qquad (4.2)\\
&|\mathcal{U}^2_{\mathcal{R}[v]}| + |\mathcal{U}^5_{\mathcal{R}[v]}| + |\mathcal{U}^8_{\mathcal{R}[v]}| + |\mathcal{U}^9_{\mathcal{R}[v]}| \\
&+ |\mathcal{U}^{11}_{\mathcal{R}[v]}| + |\mathcal{U}^{12}_{\mathcal{R}[v]}| + |\mathcal{U}^{14}_{\mathcal{R}[v]}| + |\mathcal{U}^{15}_{\mathcal{R}[v]}| \leq 2l_{\{\vec{T}_H,\mathcal{R}\}} \qquad (4.3)\\
&|\mathcal{U}^3_{\mathcal{R}[v]}| + |\mathcal{U}^6_{\mathcal{R}[v]}| + |\mathcal{U}^8_{\mathcal{R}[v]}| + |\mathcal{U}^{10}_{\mathcal{R}[v]}| \\
&+ |\mathcal{U}^{12}_{\mathcal{R}[v]}| + |\mathcal{U}^{13}_{\mathcal{R}[v]}| + |\mathcal{U}^{14}_{\mathcal{R}[v]}| + |\mathcal{U}^{15}_{\mathcal{R}[v]}| \leq 2l_{\{\vec{T}_H,\mathcal{R}\}} \qquad (4.4)\\
&|\mathcal{U}^4_{\mathcal{R}[v]}| + |\mathcal{U}^7_{\mathcal{R}[v]}| + |\mathcal{U}^9_{\mathcal{R}[v]}| + |\mathcal{U}^{10}_{\mathcal{R}[v]}| \\
&+ |\mathcal{U}^{11}_{\mathcal{R}[v]}| + |\mathcal{U}^{13}_{\mathcal{R}[v]}| + |\mathcal{U}^{14}_{\mathcal{R}[v]}| + |\mathcal{U}^{15}_{\mathcal{R}[v]}| \leq 2l_{\{\vec{T}_H,\mathcal{R}\}} \qquad (4.5)
\end{aligned}
$$

Note that inequalities (4.2), (4.3), (4.4) and (4.5) actually bound the size of maximal cliques listed as (i), (ii), (iii) and (iv), respectively, by $2l_{\{\vec{T}_H,\mathcal{R}\}}$.

Adding inequalities (4.3), (4.4), (4.5) and 2×(4.2), we get

$$
\begin{aligned}
&2|\mathcal{U}^1_{\mathcal{R}[v]}| + |\mathcal{U}^2_{\mathcal{R}[v]}| + |\mathcal{U}^3_{\mathcal{R}[v]}| + |\mathcal{U}^4_{\mathcal{R}[v]}| + 3|\mathcal{U}^5_{\mathcal{R}[v]}| \\
&+ 3|\mathcal{U}^6_{\mathcal{R}[v]}| + 3|\mathcal{U}^7_{\mathcal{R}[v]}| + 2|\mathcal{U}^8_{\mathcal{R}[v]}| + 2|\mathcal{U}^9_{\mathcal{R}[v]}| + 2|\mathcal{U}^{10}_{\mathcal{R}[v]}| \\
&+ 4|\mathcal{U}^{11}_{\mathcal{R}[v]}| + 4|\mathcal{U}^{12}_{\mathcal{R}[v]}| + 4|\mathcal{U}^{13}_{\mathcal{R}[v]}| + 3|\mathcal{U}^{14}_{\mathcal{R}[v]}| + 5|\mathcal{U}^{15}_{\mathcal{R}[v]}| \leq 10l_{\{\vec{T}_H,\mathcal{R}\}} \\
&\Rightarrow |\mathcal{U}^5_{\mathcal{R}[v]}| + |\mathcal{U}^6_{\mathcal{R}[v]}| + |\mathcal{U}^7_{\mathcal{R}[v]}| + |\mathcal{U}^{11}_{\mathcal{R}[v]}| \\
&+ |\mathcal{U}^{12}_{\mathcal{R}[v]}| + |\mathcal{U}^{13}_{\mathcal{R}[v]}| + |\mathcal{U}^{14}_{\mathcal{R}[v]}| + |\mathcal{U}^{15}_{\mathcal{R}[v]}| \leq \frac{10}{3}l_{\{\vec{T}_H,\mathcal{R}\}}. \qquad (4.6)
\end{aligned}
$$

Inequality (4.6) bounds the size of maximal clique listed as (v) above. We can similarly show that the size of maximal cliques listed as (vi), (vii) and (viii) are also bounded by $\frac{10}{3}l_{\{\vec{T}_H,\mathcal{R}\}}$.




(a) Partition of the set $\mathcal{U}_{\mathcal{R}[v]}$ of subtrees into 7 equivalence classes: $\mathcal{U}^1_{\mathcal{R}[v]}, \ldots, \mathcal{U}^7_{\mathcal{R}[v]}$.

(b) Structure of the conflict graph $G_{\mathcal{U}_{\mathcal{R}[v]}}$.

Figure 4.3: Equivalence classes and structure of the conflict graph of subtrees $\mathcal{U}_{\mathcal{R}[v]}$ in the case when the degree of the host tree is 3.

Adding inequalities (4.2), (4.3) and (4.5), we get

$$\begin{aligned}
|\mathcal{U}^1_{\mathcal{R}[v]}| + |\mathcal{U}^2_{\mathcal{R}[v]}| + |\mathcal{U}^4_{\mathcal{R}[v]}| + 2|\mathcal{U}^5_{\mathcal{R}[v]}| + |\mathcal{U}^6_{\mathcal{R}[v]}| & \\
+ 2|\mathcal{U}^7_{\mathcal{R}[v]}| + |\mathcal{U}^8_{\mathcal{R}[v]}| + 2|\mathcal{U}^9_{\mathcal{R}[v]}| + |\mathcal{U}^{10}_{\mathcal{R}[v]}| + 3|\mathcal{U}^{11}_{\mathcal{R}[v]}| & \\
+ 3|\mathcal{U}^{12}_{\mathcal{R}[v]}| + 3|\mathcal{U}^{13}_{\mathcal{R}[v]}| + 3|\mathcal{U}^{14}_{\mathcal{R}[v]}| + 3|\mathcal{U}^{15}_{\mathcal{R}[v]}| & \leq 6l_{\{\vec{T}_H,\mathcal{R}\}} \\
\Rightarrow |\mathcal{U}^5_{\mathcal{R}[v]}| + |\mathcal{U}^7_{\mathcal{R}[v]}| + |\mathcal{U}^9_{\mathcal{R}[v]}| + |\mathcal{U}^{11}_{\mathcal{R}[v]}| & \\
+ |\mathcal{U}^{12}_{\mathcal{R}[v]}| + |\mathcal{U}^{13}_{\mathcal{R}[v]}| + |\mathcal{U}^{14}_{\mathcal{R}[v]}| + |\mathcal{U}^{15}_{\mathcal{R}[v]}| & \leq 3l_{\{\vec{T}_H,\mathcal{R}\}}
\end{aligned} \tag{4.7}$$

Inequality (4.7) bounds the size of maximal clique listed as (ix) above. We can similarly show that the size of maximal cliques listed as (x), (xi) and (xii) are also bounded by $3l_{\{\vec{T}_H,\mathcal{R}\}}$.

Hence, for any host tree vertex $v \in V_T$, the size of maximum clique in conflict graph $G_{\mathcal{U}_{\mathcal{R}[v]}}$ is upper bounded by $\frac{10}{3}l_{\{\vec{T}_H,\mathcal{R}\}}$, i.e., $\omega_{G_{\mathcal{U}_{\mathcal{R}[v]}}} \leq \frac{10}{3}l_{\{\vec{T}_H,\mathcal{R}\}}$. Therefore, from equation (4.1), the size of maximum clique in the conflict graph $G_{\mathcal{U}_{\mathcal{R}}}$ is upper bounded as $\omega_{G_{\mathcal{U}_{\mathcal{R}}}} \leq \frac{10}{3}l_{\{\vec{T}_H,\mathcal{R}\}}$. $\square$

Next we prove the main theorem of this section.

**Theorem 4.7.** *SUBTREE-BASED-WA is a $\frac{10}{3}$-approximation algorithm for the restricted MIN-MC-WA-BT problem where the degree of the bidirected tree is* 4.

*Proof.* As stated before, SUBTREE-BASED-WA assigns wavelengths to the rooted subtrees in the set $\mathcal{R}$ as determined by vertex coloring of $G_{\mathcal{U}_\mathcal{R}}$, the conflict graph of their skeleton subtrees. When the degree of the bidirected tree is $\Delta_{\vec{T}_H} = 4$, the conflict graph $G_{\mathcal{U}_\mathcal{R}}$ is weakly chordal, and the following hold.

(i) Coloring $G_{\mathcal{U}_\mathcal{R}}$ is easy. Therefore, the total number of wavelengths required by SUBTREE-BASED-WA is equal to $\chi_{G_{\mathcal{U}_\mathcal{R}}}$.

(ii) The conflict graph $G_{\mathcal{U}_\mathcal{R}}$ is perfect [41, p.146]. Therefore, its chromatic number is equal to its clique number, i.e., $\chi_{G_{\mathcal{U}_\mathcal{R}}} = \omega_{G_{\mathcal{U}_\mathcal{R}}}$.

Hence, by Lemma 4.6 we get the upper bound on the number of wavelengths required by the algorithm as

$$|\lambda^{\text{SUB}}(\mathcal{R})| = \chi_{G_{\mathcal{U}_\mathcal{R}}} = \omega_{G_{\mathcal{U}_\mathcal{R}}} \leq \frac{10}{3} l_{\{\vec{T}_H,\mathcal{R}\}}. \tag{4.8}$$

Note that the minimum number of wavelengths required for assigning wavelengths to the set $\mathcal{R}$ of rooted subtrees on the bidirected tree $\vec{T}_H$ is lower bounded by $l_{\{\vec{T}_H,\mathcal{R}\}}$, i.e.,

$$\min_{\lambda \in \Lambda_{\{\vec{T}_H,\mathcal{R}\}}} |\lambda(\mathcal{R})| \geq l_{\{\vec{T}_H,\mathcal{R}\}}. \tag{4.9}$$

From equations (4.8) and (4.9), we obtain

$$\frac{|\lambda^{\text{SUB}}(\mathcal{R})|}{\min_{\lambda \in \Lambda_{\{\vec{T}_H,\mathcal{R}\}}} |\lambda(\mathcal{R})|} \leq \frac{10}{3},$$

which gives the required approximation ratio for SUBTREE-BASED-WA. □

As already stated, Lemmas 4.3, 4.4 and 4.5 are independent of the degree of the host tree $H$. In particular, they hold for $\Delta_H = 2, 3$ as well. It is much easier to determine the upper bound on the size of maximum clique in the conflict graph


(a) Partition of the set $\mathcal{U}_{\mathcal{R}[v]}$ of subtrees into 3 equivalence classes: $\mathcal{U}^1_{\mathcal{R}[v]}, \mathcal{U}^2_{\mathcal{R}[v]}, \mathcal{U}^3_{\mathcal{R}[v]}$.

(b) Structure of the conflict graph $G_{\mathcal{U}_{\mathcal{R}[v]}}$.

Figure 4.4: Equivalence classes and structure of the conflict graph of subtrees $\mathcal{U}_{\mathcal{R}[v]}$ in the case when the degree of the host tree is 2.

$G_{\mathcal{U}_{\mathcal{R}}}$ for the case when $\Delta_H = 2, 3$ compared to the case when $\Delta_H = 4$ (Lemma 4.6). These bounds are $2l_{\{\vec{T}_H,\mathcal{R}\}}$ and $3l_{\{\vec{T}_H,\mathcal{R}\}}$ for the case when the degree of the host tree is 2 and 3, respectively. For the case when $\Delta_H = 3$, Figure 4.3(a) shows a sample subtree from each of the equivalence classes (as defined before) in the set $\mathcal{U}_{\mathcal{R}[v]}$ in the neighborhood of host tree vertex $v \in V_H$. Figure 4.3(b) depicts the structure of the conflict graph $G_{\mathcal{U}_{\mathcal{R}[v]}}$. Each vertex in the figure represents a clique of subtrees. An edge between two vertices represents an edge between every subtree in one set and every subtree in the other set. The corresponding figures for the case when $\Delta_H = 2$ are presented as Figures 4.4(a) and 4.4(b). The reader is encouraged to use Figures 4.3(b) and 4.4(b) and determine (analogous to Lemma 4.6) the upper bound on the size of maximum clique in the conflict graph $G_{\mathcal{U}_{\mathcal{R}}}$ when $\Delta_H = 3, 2$ respectively. The arguments presented in the proof of Theorem 4.7 also hold and we get the approximation ratio of 2 and 3 when $\Delta_{\vec{T}_H} = 2$ and 3, respectively.

## 4.3 Complexity Analysis

The time complexity of SUBTREE-BASED-WA wavelength assignment scheme depends on the complexity of the algorithm employed for coloring the conflict graph $G_{\mathcal{U}_{\mathcal{R}}}$. When $\Delta_{\vec{T}_H} \leq 4$, the scheme has a polynomial running time. In particular, we have the following result.

**Proposition 4.8.** *For the given instance $\{\vec{T}_H, \mathcal{R}\}$ of the MIN-MC-WA-BT problem, the running time complexity of SUBTREE-BASED-WA is:*

*(i)* $O\big(|\mathcal{R}|^2\big(|E_H| + |\mathcal{R}|\big)\big)$ *when* $\Delta_{\vec{T}_H} = 4$.

*(ii)* $O\big(|E_H||\mathcal{R}|^2\big)$ *when* $\Delta_{\vec{T}_H} = 3$.

*(iii)* $O\big(|\mathcal{R}| \log |\mathcal{R}|\big)$ *when* $\Delta_{\vec{T}_H} = 2$.

*Proof.* First note that in SUBTREE-BASED-WA, for constructing the conflict graph $G_{\mathcal{U}_\mathcal{R}}$, we need to decide for every pair of subtrees in the set $\mathcal{U}_\mathcal{R}$, whether the subtrees in that pair collide or not. For each pair we have to check for collision on a maximum of $|E_H|$ host tree edges. Therefore, the conflict graph can be constructed in $O\big(|E_H||\mathcal{R}|^2\big)$ time.

The complexity of minimum vertex coloring in a weakly chordal graph $W$ is $O\big(|V_W|^3\big)$ [36]. Also, as stated before, for the case when $\Delta_{\vec{T}_H} = 4$ the conflict graph $G_{\mathcal{U}_\mathcal{R}}$ is a weakly chordal graph. Therefore, in this case the complexity of SUBTREE-BASED-WA is $O\big(|\mathcal{R}|^2\big(|E_H| + |\mathcal{R}|\big)\big)$.

Minimum vertex coloring in a chordal graph $C$ is solvable in $O\big(|V_C| + |E_C|\big)$ time [43]. Also as stated before, for the case when $\Delta_{\vec{T}_H} = 3$ the conflict graph $G_{\mathcal{U}_\mathcal{R}}$ is a chordal graph. Therefore, in this case the complexity of SUBTREE-BASED-WA is determined by the complexity of constructing the conflict graph, i.e., the complexity of SUBTREE-BASED-WA is $O\big(|E_H||\mathcal{R}|^2\big)$.

As stated before, when $\Delta_{\vec{T}_H} = 2$ the conflict graph $G_{\mathcal{U}_\mathcal{R}}$ is an interval graph. In fact, in this case compared to first constructing and then coloring conflict graph $G_{\mathcal{U}_\mathcal{R}}$, it is much more efficient to treat the subtrees as intervals and straightaway assign colors to them. The complexity of coloring a given set $I$ of intervals is $O\big(|I| \log |I|\big)$ [44]. Therefore the complexity of SUBTREE-BASED-WA in the case when $\Delta_{\vec{T}_H} = 2$ is $O\big(|\mathcal{R}| \log |\mathcal{R}|\big)$. $\square$

# Chapter 5

## NP Completeness Results for Multicast Wavelength Assignment in Bidirected Trees

In this chapter, we state and prove three NO completeness results. Two of these are for a pair of restricted versions of the MIN-MC-WA-BT problem. The third result is for a problem related to another restricted version of the MIN-MC-WA-BT problem.

### 5.1 Motivation and Background

In this section, we discuss the significance of the hardness results that we later prove in Sections 5.2, 5.3 and 5.4. We also give some background required for the proofs.

In Sections 5.2, and 5.3, we prove that the decision version of the MIN-MC-WA-BT problem defined in Section 2.2 is NP complete even under the following restricted settings:

(i) The bidirected tree is restricted to being a bidirected star.

(ii) The bidirected tree is restricted to being a bidirected path.

Analogous to the definition of a bidirected tree, a bidirected star (path) is defined as the directed graph generated by replacing the edges of a star (path) by pairs of anti-parallel directed edges.

These results are interesting because in both these cases, if we add an additional restriction that the set of rooted subtrees is restricted to being directed paths, the problem becomes tractable. This is because directed path coloring in bidirected stars is equivalent to the problem of edge coloring in bipartite graphs, and directed path coloring in bidirected paths is equivalent to the problem of interval coloring. Both interval coloring [44] and edge coloring in bipartite graphs [45] are solvable in polynomial time. Observe that the restricted MIN-MC-WA-BT problem, where the rooted subtrees are restricted to being directed paths, is nothing but the routing and wavelength assignment problem for unicast traffic requests in bidirected trees under all-optical networking paradigm. Therefore, the hardness results show that the multicast routing and wavelength assignment is inherently harder than the unicast case when we restrict the fiber topology to being a bidirected path or a bidirected star (both of which are interesting topologies from practical standpoint). This suggests that simply tweaking the algorithms developed for unicast routing and wavelength assignment may not result in good algorithms for the multicast case, and there is a need to develop and study new techniques that are dedicated to the multicast problem.

In Section 5.4, we prove a hardness result related to the MIN-MC-WA-BT problem restricted to the case when the bidirected tree has degree at most 3. Recall that in Section 2.2, we showed that any given instance of the MIN-MC-WA-BT problem is equivalent to the problem of minimum vertex coloring of the conflict graph corresponding to the given set of rooted subtrees of the given bidirected tree. Since the clique number of any graph provides a 'good' lower bound for its chromatic number, in order to get a good lower bound on the minimum number of wavelengths required by any traffic grooming solution for a given instance of the MIN-MC-WA-

BT problem, it makes sense to study the problem of determining the clique number of the corresponding conflict graph. We prove that the decision version of the problem of determining the clique number of the conflict graphs corresponding to the set of MIN-MC-WA-BT problems restricted to the case when the degree of the bidirected tree is at most 3, is NP complete.

Before proceeding any further, let us state MC-WA-BT, the decision version of the MIN-MC-WA-BT problem.

**Problem 5.1** (MC-WA-BT). *Given a triple $\{\vec{T}_H, \mathcal{R}, k\}$, where $\vec{T}_H$ is a bidirected tree, $\mathcal{R}$ is a set of rooted subtrees on $\vec{T}_H$ and $k$ is a positive integer; consider a set of mappings $\Lambda_{\{\vec{T}_H,\mathcal{R}\}}$ from $\mathcal{R}$ to $\mathbb{N}$, such that for any mapping $\lambda \in \Lambda_{\{\vec{T}_H,\mathcal{R}\}}$, if a pair of rooted subtrees $\vec{R}_i, \vec{R}_j \in \mathcal{R}$ collide, then $\lambda(\vec{R}_i) \neq \lambda(\vec{R}_j)$.*

*Is there a mapping $\lambda^* \in \Lambda_{\{\vec{T}_H,\mathcal{R}\}}$ such that $|\lambda^*(\mathcal{R})| \leq k$?*

To show the NP completeness of MC-WA-BT problem in the desired restricted settings, we first prove that the general MC-WA-BT problem is in NP.

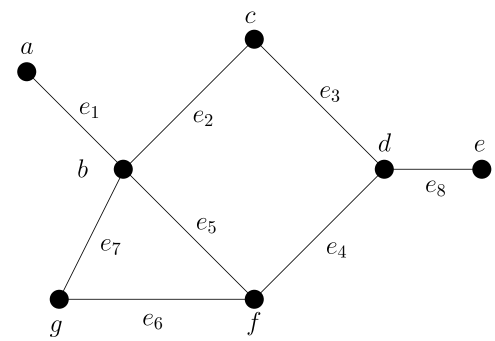
**Lemma 5.2.** *MC-WA-BT is in NP.*

*Proof.* Given any instance of $\{\vec{T}_H, \mathcal{R}, k\}$, of the MC-WA-BT problem, and any mapping $\lambda : \mathcal{R} \longrightarrow \mathbb{N}$, we can verify in $O\big(|E_{\vec{T}_H}||\mathcal{R}|^2\big)$ time, whether $\lambda$ is a certificate (as defined in the definition of the MC-WA-BT problem) for the given instance of the problem or not. Hence, MC-WA-BT is in NP. $\square$

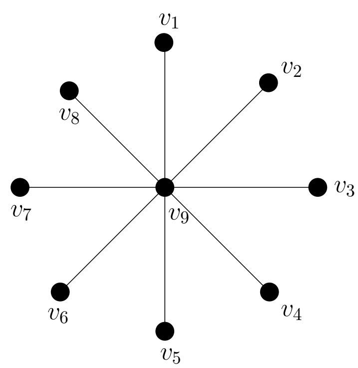
## 5.2 Bidirected Stars

In this section, we prove that the MC-WA-BT problem described in Section 5.1, restricted to the case when the bidirected tree is a bidirected star, is NP complete.

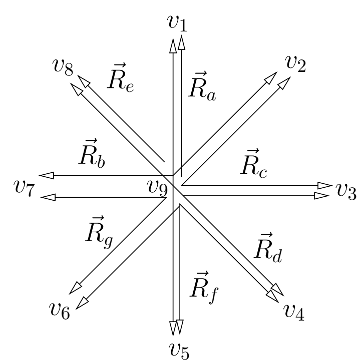
We prove the hardness result by reduction from the problem of vertex coloring in graphs (COL). For completeness, the exact definition of the COL problem is given next.




(a) Graph $G$ from the given instance of COL.




(b) Host star used to generated the bidirected star $\vec{S}$.




(c) Set $\mathcal{R}$ of rooted subtrees.

Figure 5.1: Construction of an instance of MC-WA-BT restricted to bidirected stars, equivalent to a given instance of COL.

**Problem 5.3** (COL). *Given a pair $\{G, k\}$, where $G$ is a graph and $k$ is a positive integer; consider the set of mappings $\Psi_G$ from $V_G$ to $\mathbb{N}$ such that for any mapping $\psi \in \Psi_G$, if a pair of vertices $u, v \in V_G$ are adjacent to each other, i.e., if there is an edge $\{u, v\} \in E_G$, then $\psi(u) \neq \psi(v)$.*

*Does there exist a mapping $\psi^* \in \Psi_G$ such that $|\psi^*(V_G)| \leq k$?*

It is known that COL is NP complete [46].

**Theorem 5.4.** *MC-WA-BT restricted to bidirected stars is NP complete.*

*Proof.* Let $\{G, k\}$ be any given instance of the COL problem. Label the edges of the graph $G$ from $e_1$ to $e_{|V_G|}$. Generate an instance of MC-WA-BT problem as follows:

(i) Construct a bidirected star $\vec{S}$ with $|E_G| + 1$ vertices. Label the leaf vertices from $v_1$ to $v_{|E_G|}$ starting from any leaf vertex and traversing clockwise through all the leaves. Label the eye of the star as $v_{|E_G|+1}$.

(ii) Corresponding to each vertex $a \in V_G$, construct a rooted subtree $\vec{R}_a$ of $\vec{S}$ with directed edge set $E_{\vec{R}_a} = \{(v_{|E_G|+1}, v_i) : e_i \in E_G \setminus E_{G[V_G \setminus \{a\}]}\}$, and vertex set $V_{\vec{R}_a} = \{v_{|E_G|+1}\} \cup \{v_i : e_i \in E_G \setminus E_{G[V_G \setminus \{a\}]}\}$. In other words, the rooted subtree $\vec{R}_a$, corresponding to the vertex $a \in V_G$, contains the directed edge $(v_{|E_G|+1}, v_i)$ if and only if edge $e_i \in E_G$ is adjacent to $a$.

The time required for this construction is linear in the size of the graph $G$. An example construction is presented in Figure 5.1. Let the graph $G$ specified by the given instance of the COL problem be as shown in Figure 5.1(a). In Figure 5.1(b), we present the host star graph that is used to generate the bidirected star $\vec{S}$. Finally, in Figure 5.1(c), we present the set $\mathcal{R}$ of rooted subtrees of the bidirected star $\vec{S}$. For clarity, in the figure we annotate the set of vertices of $\vec{S}$, even though we have omitted the directed edges and vertices of $\vec{S}$.

By construction, there is an edge $\{a, b\} \in E_G$ if an only if the corresponding pair of rooted subtrees $\vec{R}_a, \vec{R}_b \in \mathcal{R}$ collide. To observe this, first assume that the edge $\{a, b\}$ is labeled as $e_i$ during the edge labeling. In this case, since the directed edge $(v_{|E_G|+1}, v_i) \in E_{\vec{S}}$ is contained in both the rooted subtrees $\vec{R}_a$ and $\vec{R}_b$, they collide. Next assume that the rooted subtree pair $\vec{R}_a, \vec{R}_b \in \mathcal{R}$ collide. By construction, all the directed edges present in any rooted subtree in the set $\mathcal{R}$ are of the form $(v_{|E_G|+1}, v_j)$ where $j \in \{1, \ldots, |E_G|\}$. Without loss of any generality, since $\vec{R}_a, \vec{R}_b$ collide, assume that they collide on the directed edge $(v_{|E_G|+1}, v_i) \in E_{\vec{S}}$. Since the directed edge $(v_{|E_G|+1}, v_i) \in E_{\vec{R}_a}$, the edge $e_i$ is incident on the vertex $a$. Similarly we show that the edge $e_i$ is incident on the vertex $b$. Hence, the edge $e_i$ is nothing but the edge $\{a, b\}$, i.e., $\{a, b\} \in E_G$.

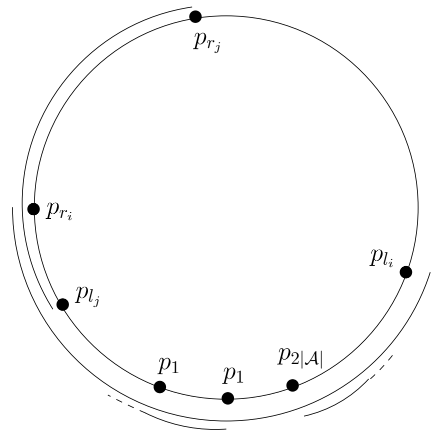
From the above claim, we have a bijection between the set of mappings $\Psi_G$ and $\Lambda_{\{\vec{S},\mathcal{R}\}}$. Moreover, for any $\psi \in \Psi_G$ and the corresponding $\lambda \in \Lambda_{\{\vec{S},\mathcal{R}\}}$, $|\psi(V_G)| = |\lambda(\mathcal{R})|$. This proves that the instance $\{G, k\}$ of COL is equivalent to the instance $\{\vec{S}, \mathcal{R}, k\}$ of MC-WA-BT where $\vec{S}$ is a bidirected star. Hence, the problem COL is reducible to the problem MC-WA-BT restricted to bidirected stars. Finally, applying Lemma 5.2 completes the proof. □

## 5.3 Bidirected Paths

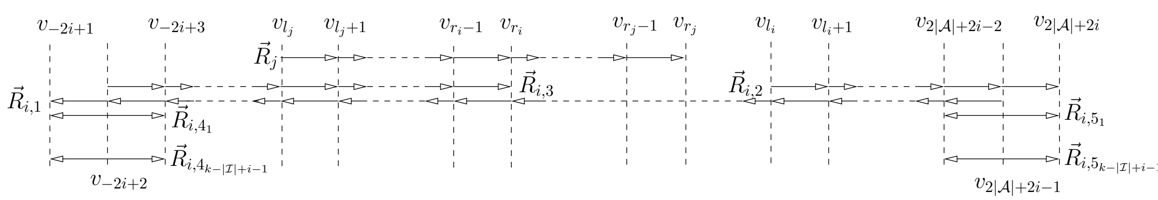
In this section, we prove that the MC-WA-BT problem described in Section 5.1, restricted to the case when the bidirected tree is a bidirected path, is NP complete.

We prove the hardness result by reduction from the circular arc coloring problem (ARC-COL). For completeness, we give the exact definition of the ARC-COL problem. But before presenting the ARC-COL problem, we need to tie down a few notations. Given a circle $C$, an arc on $C$ is denoted by an ordered pair $(p_l, p_r)$, where $p_l$ and $p_r$ are points on the circle $C$. The arc $(p_l, p_r)$ is the set of all the points on $C$ encountered while traversing the circle in clockwise direction starting from point $p_l$ and ending at point $p_r$. In this case $p_l$ and $p_r$ are referred to as the *end points* of the arc. More specifically, $p_l$ is the *left* end point and $p_r$ is the *right* end point. Arcs $(p_{l_i}, p_{r_i}), (p_{l_j}, p_{r_j})$ on circle $C$ are said to *overlap*, if they share some common point on the circle, i.e., if $(p_{l_i}, p_{r_i}) \cap (p_{l_j}, p_{r_j}) \neq \emptyset$. An arc $(p_{l_i}, p_{r_i})$ on circle $C$ is said to *contain* another arc $(p_{l_j}, p_{r_j})$ on $C$, if all the points of $(p_{l_j}, p_{r_j})$ are also in $(p_{l_i}, p_{r_i})$, i.e., if $(p_{l_j}, p_{r_j}) \subseteq (p_{l_i}, p_{r_i})$.

Next we state the ARC-COL problem.




(a) Set of circular arcs from the given instance of ARC-COL.




(b) Set $\mathcal{R}_i \cup \mathcal{R}_j$ of rooted subtrees on the bidirected path $\vec{P}$.

Figure 5.2: Construction of an instance of MC-WA-BT restricted to bidirected paths, equivalent to a given instance of ARC-COL.

**Problem 5.5** (ARC-COL). *Given a triple $\{C, \mathcal{A}, k\}$, where $C$ is a circle, $\mathcal{A}$ is a set of arcs on the circle with distinct end points, and $k$ is a positive integer; consider a set of mappings $\Theta_{\{C,\mathcal{A}\}}$ from $\mathcal{A}$ to $\mathbb{N}$ such that for any mapping $\theta \in \Theta_{\{C,\mathcal{A}\}}$, if a pair of arcs $(p_{l_i}, p_{r_i}), (p_{l_j}, p_{r_j}) \in \mathcal{A}$ overlap, then $\theta((p_{l_i}, p_{r_i})) \neq \theta((p_{l_j}, p_{r_j}))$.*

*Is there a mapping $\theta^* \in \Theta_{\{C,\mathcal{A}\}}$ such that $|\theta^*(\mathcal{A})| \leq k$?*

It is known that ARC-COL is NP complete [47].

**Theorem 5.6.** *MC-WA-BT restricted to bidirected paths is NP complete.*

*Proof.* Let $\{C, \mathcal{A}, k\}$ be any given instance of the ARC-COL problem. From among the $2|\mathcal{A}|$ end points belonging to all the arcs in $\mathcal{A}$, select a point $p$ that satisfies the following:

(i) Point $p$ is the left end point of some arc in $\mathcal{A}$.

(ii) The first end point encountered on traversing the circle $C$ in anticlockwise direction while starting from point $p$, is a right end point of some arc in $\mathcal{A}$.

Such an end point must exist because, of the $2|\mathcal{A}|$ end points belonging to all the arcs in $\mathcal{A}$, exactly $|\mathcal{A}|$ are left end points and $|\mathcal{A}|$ are right end points. Next, label the end points of the arcs from $p_1$ to $p_{2|\mathcal{A}|}$ starting by labeling the selected end point $p$ as $p_1$, and moving clockwise on the circle.

Partition the set $\mathcal{A}$ into subsets $\mathcal{I}$ and $\mathcal{J}$ where $\mathcal{I}$ is the set of all the arcs in $\mathcal{A}$ that contain the arc $(p_{2|\mathcal{A}|}, p_1)$, i.e., $\mathcal{I} := \{(p_l, p_r) \in \mathcal{A} : (p_{2|\mathcal{A}|}, p_1) \subseteq (p_l, p_r)\}$ and $\mathcal{J} := \mathcal{A} \backslash \mathcal{I}$. Without loss of any generality, assume that $\mathcal{I} = \{(p_{l_1}, p_{r_1}), \ldots, (p_{l_{|\mathcal{I}|}}, p_{r_{|\mathcal{I}|}})\}$. Therefore, $\mathcal{J} = \{(p_{l_{|\mathcal{I}|+1}}, p_{r_{|\mathcal{I}|+1}}), \ldots, (p_{l_{|\mathcal{A}|}}, p_{r_{|\mathcal{A}|}})\}$. A consequence of the labeling described above is that none of the arcs in the set $\mathcal{I}$ have either $p_1$ or $p_{2|\mathcal{A}|}$ as an end point.

Next, construct a bidirected path $\vec{P}$ with $2|\mathcal{A}| + 4|\mathcal{I}|$ vertices that are labeled from $v_{-2|\mathcal{I}|+1}$ to $v_{2|\mathcal{A}|+2|\mathcal{I}|}$ starting from one leaf and traversing the path to reach the other leaf. For every arc $(p_{l_i}, p_{r_i}) \in \mathcal{A}$, construct a set of rooted subtrees $\mathcal{R}_i$ of $\vec{P}$. If the arc $(p_{l_i}, p_{r_i}) \in \mathcal{I}$, then

$$\mathcal{R}_i = \bigcup_{j=1}^{5} \mathcal{R}_{i,j},$$

where

$$\mathcal{R}_{i,j} = \begin{cases} \{\vec{R}_{i,j}\} & \text{for } j \in \{1,2,3\}, \\ \{\vec{R}_{i,j_1}, \vec{R}_{i,j_2}, \ldots, \vec{R}_{i,j_{k-|\mathcal{I}|+i-1}}\} & \text{for } j \in \{4,5\}. \end{cases}$$

The vertex sets and the directed edge sets of the various rooted subtrees constructed above are defined as

$$\begin{aligned}
V_{\vec{R}_{i,1}} &= \{v_{2|\mathcal{A}|+2i-1}, v_{2|\mathcal{A}|+2i-2}, \dots, v_{-2i+1}\},\\
E_{\vec{R}_{i,1}} &= \{(v_{2|\mathcal{A}|+2i-1}, v_{2|\mathcal{A}|+2i-2}), (v_{2|\mathcal{A}|+2i-2}, v_{2|\mathcal{A}|+2i-3}), \dots, (v_{-2i+2}, v_{-2i+1})\},\\
V_{\vec{R}_{i,2}} &= \{v_{l_i}, v_{l_i+1}, \dots, v_{2|\mathcal{A}|+2i}\},\\
E_{\vec{R}_{i,2}} &= \{(v_{l_i}, v_{l_i+1}), (v_{l_i+1}, v_{l_i+2}), \dots, (v_{2|\mathcal{A}|+2i-1}, v_{2|\mathcal{A}|+2i})\},\\
V_{\vec{R}_{i,3}} &= \{v_{-2i+2}, v_{-2i+3}, \dots, v_{r_i}\},\\
E_{\vec{R}_{i,3}} &= \{(v_{-2i+2}, v_{-2i+3}), (v_{-2i+3}, v_{-2i+4}), \dots, (v_{r_i-1}, v_{r_i})\},
\end{aligned}$$

and for every $j \in \{1, 2, \dots, k - |\mathcal{I}| + i - 1\}$,

$$\begin{aligned}
V_{\vec{R}_{i,4_j}} &= \{v_{2|\mathcal{A}|+2i-2}, v_{2|\mathcal{A}|+2i-1}, v_{2|\mathcal{A}|+2i}\},\\
E_{\vec{R}_{i,4_j}} &= \{(v_{2|\mathcal{A}|+2i-1}, v_{2|\mathcal{A}|+2i-2}), (v_{2|\mathcal{A}|+2i-1}, v_{2|\mathcal{A}|+2i})\},\\
V_{\vec{R}_{i,5_j}} &= \{v_{-2i+1}, v_{-2i+2}, v_{-2i+3}\},\\
E_{\vec{R}_{i,5_j}} &= \{(v_{-2i+2}, v_{-2i+1}), (v_{-2i+2}, v_{-2i+3})\}.
\end{aligned}$$

Otherwise, if the arc $(p_{l_i}, p_{r_i}) \in \mathcal{J}$, then $\mathcal{R}_i = \{\vec{R}_i\}$ having the vertex set and the directed edge set defined as

$$\begin{aligned}
V_{\vec{R}_i} &= \{v_{l_i}, v_{l_i+1}, \dots, v_{r_i}\},\\
E_{\vec{R}_i} &= \{(v_{l_i}, v_{l_i+1}), \dots (v_{r_i-1}, v_{r_i})\}.
\end{aligned}$$

Let $\mathcal{R} := \bigcup_{i=1}^{|\mathcal{A}|} \mathcal{R}_i$. This is a polynomial time construction. An example construction is presented in Figure 5.2. In Figure 5.2(a), we show two overlapping arcs $(p_{l_i}, p_{r_i})$ and $(p_{l_j}, p_{r_j})$ and the set of interesting points on the circle. Observe that $(p_{l_i}, p_{r_i}) \in \mathcal{I}$ and $(p_{l_j}, p_{r_j}) \in \mathcal{J}$. In Figure 5.2(b), we present the set $\mathcal{R}_i \cup \mathcal{R}_j$ of rooted subtrees of the bidirected path $\vec{P}$, corresponding to the arcs $(p_{l_i}, p_{r_i})$ and $(p_{l_j}, p_{r_j})$. For clarity, in the figure we only annotate the set of vertices of the rooted subtrees that are required to present the structure of the rooted subtrees, and do not show all the directed edges and vertices of $\vec{P}$.

We claim that the answer to the ARC-COL problem $\{C, \mathcal{A}, k\}$ is *YES* if and only if the answer to the MC-WA-BT problem $\{\vec{P}, \mathcal{R}, k\}$ is *YES*. To prove this claim, first assume that the answer to the ARC-COL problem $\{C, \mathcal{A}, k\}$ is *YES*. Let $\theta^*$ be the mapping as described in the definition of the ARC-COL problem. Without loss of any generality, assume that $\theta^*(\mathcal{A}) = \{1, 2, \ldots, |\theta^*(\mathcal{A})|\}$, where $|\theta^*(\mathcal{A})| \leq k$. Construct a mapping $\lambda : \mathcal{R} \longrightarrow \mathbb{N}$ using the mapping $\theta^*$ as described next. First, for every $i \in \{1, 2, \ldots, |\mathcal{I}|\}$,

$$\lambda(\vec{R}_{i,1}) = \lambda(\vec{R}_{i,2}) = \lambda(\vec{R}_{i,3}) = \theta^*((p_{l_i}, p_{r_i})), \tag{5.1}$$

and for every $i \in \{|\mathcal{I}| + 1, |\mathcal{I}| + 2, \ldots, |\mathcal{A}|\}$,

$$\lambda(\vec{R}_i) = \theta^*((p_{l_i}, p_{r_i})). \tag{5.2}$$

Next, for every $i \in \{1, \ldots, |\mathcal{I}|\}$, $j \in \{1, \ldots, k - |\mathcal{I}| + i - 1\}$,

$$\lambda(\vec{R}_{i,4_j}) = \lambda(\vec{R}_{i,5_j}) = \min \left\{ \{1, 2, \ldots, k\} \setminus \mathbb{F}_{i,j} \right\}, \tag{5.3}$$

where

$$\mathbb{F}_{i,j} = \bigcup_{m=1}^{j-1} \left\{ \lambda\left(\vec{R}_{i,4_m}\right) \right\} \cup \theta^*\left( \left\{ (p_{l_i}, p_{r_i}), (p_{l_{i+1}}, p_{r_{i+1}}) \ldots, (p_{l_{|\mathcal{I}|}}, p_{r_{|\mathcal{I}|}}) \right\} \right). \tag{5.4}$$

Later in this proof, we shall show that for every $i \in \{1, \ldots, |\mathcal{I}|\}$, $j \in \{1, \ldots, k - |\mathcal{I}| + i - 1\}$, the set $\{1, 2, \ldots, k\} \setminus \mathbb{F}_{i,j} \neq \emptyset$. Hence, the mapping $\lambda$ is well defined.

Observe that according to our construction, collisions between rooted subtrees in the set $\mathcal{R}$ can be classified as follows:

(i) A pair of rooted subtrees $\vec{R}_i, \vec{R}_j \in \{\vec{R}_{|\mathcal{I}|+1}, \ldots, \vec{R}_{|\mathcal{A}|}\}$ collide if and only if the arcs $(p_{l_i}, p_{r_i}), (p_{l_j}, p_{r_j}) \in \mathcal{J}$ overlap.

(ii) Let $\mathcal{S}_i = \{\vec{R}_{1,i}, \ldots, \vec{R}_{|\mathcal{I}|,i}\}$, for $i \in \{1, 2, 3\}$. For $i \in \{1, 2, 3\}$, all the rooted subtrees in the set $\mathcal{S}_i$ collide. Note that all the arcs in the set $\mathcal{I}$ contain the arc $(p_{2|\mathcal{A}|}, p_1)$ and therefore, are mutually overlapping.

(iii) For $i \in \{|\mathcal{I}|+1, \ldots, |\mathcal{A}|\}$ and $j \in \{1, \ldots, |\mathcal{I}|\}$, the rooted subtree $\vec{R}_i$ collides with at least one of the rooted subtrees $\vec{R}_{j,2}, \vec{R}_{j,3}$ if and only if the arcs $(p_{l_i}, p_{r_i}) \in \mathcal{J}$ and $(p_{l_j}, p_{r_j}) \in \mathcal{I}$ overlap.

(iv) Let $\mathcal{S}_{i,j} = \{\vec{R}_{i,j_1}, \ldots, \vec{R}_{i,j_{k-|\mathcal{I}|+i-1}}\}$ for $i \in \{1, \ldots, |\mathcal{I}|\}$ and $j \in \{4, 5\}$. For $i \in \{1, \ldots, |\mathcal{I}|\}$ all the rooted subtrees in the set $\mathcal{S}_{i,4}$ collide with each other, and also with all the rooted subtrees in the sets $\{\vec{R}_{i,1}, \ldots, \vec{R}_{|\mathcal{I}|,1}\}$ and $\{\vec{R}_{i,2}, \ldots, \vec{R}_{|\mathcal{I}|,2}\}$; and all the rooted subtrees in the set $\mathcal{S}_{i,5}$ collide with each other, and also with all the rooted subtrees in the sets $\{\vec{R}_{i,1}, \ldots, \vec{R}_{|\mathcal{I}|,1}\}$ and $\{\vec{R}_{i,3}, \ldots, \vec{R}_{|\mathcal{I}|,3}\}$.

Besides the collisions described above, there can be no other collisions between the rooted subtrees in the set $\mathcal{R}$.

Consider a collision of type (i). Since the arcs $(p_{l_i}, p_{r_i}), (p_{l_j}, p_{r_j})$ overlap, $\theta^*((p_{l_i}, p_{r_i})) \neq \theta^*((p_{l_j}, p_{r_j}))$. Also, the mapping $\lambda$ for rooted subtrees $\vec{R}_i, \vec{R}_j$ is defined according to equation (5.2). Hence $\lambda(\vec{R}_i) \neq \lambda(\vec{R}_j)$.

Consider a collision of type (ii). Since the arcs in the set $\mathcal{I}$ are mutually overlapping, $\theta^*$ maps distinct arcs in the set to distinct values. Also, the mapping $\lambda$ for rooted subtrees in the sets $\mathcal{S}_i$, for $i \in \{1, 2, 3\}$ is defined according to equation (5.1). Hence, for $i \in \{1, 2, 3\}$, distinct rooted subtrees in the set $\mathcal{S}_i$ are assigned distinct values by the mapping $\lambda$.

Consider a collision of type (iii). Since the arcs $(p_{l_i}, p_{r_i}), (p_{l_j}, p_{r_j})$ overlap, $\theta^*((p_{l_i}, p_{r_i})) \neq \theta^*((p_{l_j}, p_{r_j}))$. Also, the mapping $\lambda$ for the rooted subtree $\vec{R}_i$ is defined according to equation (5.2), and for the rooted subtrees $\vec{R}_{j,2}, \vec{R}_{j,3}$, it is defined according to equation (5.1). Hence $\lambda(\vec{R}_{j,2}), \lambda(\vec{R}_{j,3}) \neq \lambda(\vec{R}_i)$.

Consider a collision of type (iv). Equation (5.1) ensures that

$$\theta^*\Big(\Big\{(p_{l_i}, p_{r_i}), \ldots, (p_{l_{|\mathcal{I}|}}, p_{r_{|\mathcal{I}|}})\Big\}\Big) = \lambda\Bigg(\Bigg\{\vec{R}_{i,1}, \vec{R}_{i,2}, \vec{R}_{i,3}, \ldots, \vec{R}_{|\mathcal{I}|,1}, \vec{R}_{|\mathcal{I}|,2}, \vec{R}_{|\mathcal{I}|,3}\Bigg\}\Bigg). \quad (5.5)$$

From equations (5.4) and (5.5), we get

$$\mathbb{F}_{i,j} = \lambda\left(\bigcup_{m=1}^{j-1}\left\{\vec{R}_{i,4_m}\right\} \cup \left\{\vec{R}_{i,1}, \vec{R}_{i,2}, \vec{R}_{i,3}, \ldots, \vec{R}_{|\mathcal{I}|,1}, \vec{R}_{|\mathcal{I}|,2}, \vec{R}_{|\mathcal{I}|,3}\right\}\right). \tag{5.6}$$

Mapping $\lambda$ for rooted subtrees in the sets $\mathcal{S}_{i,j}$, for $i \in \{1, \ldots, |\mathcal{I}|\}$ and $j \in \{4, 5\}$ is defined according to equation (5.3). Hence, for $i \in \{1, \ldots, |\mathcal{I}|\}$ and $j \in \{4, 5\}$, distinct rooted subtrees in the set $\mathcal{S}_{i,j}$ are assigned values by the mapping $\lambda$ that are distinct not only with each other, but also from the values assigned by the mapping $\lambda$ to the rooted subtrees in the set $\{\vec{R}_{i,1}, \vec{R}_{i,2}, \vec{R}_{i,3}, \ldots, \vec{R}_{|\mathcal{I}|,1}, \vec{R}_{|\mathcal{I}|,2}, \vec{R}_{|\mathcal{I}|,3}\}$.

Hence, the mapping $\lambda$ respects all the collisions among rooted subtrees in the set $\mathcal{R}$ and is as described in the definition of the MC-WA-BT problem, i.e., $\lambda \in \Lambda_{\{\vec{P},\mathcal{R}\}}$.

Next we shall show that, for every $i \in \{1, \ldots, |\mathcal{I}|\}$, $j \in \{1, \ldots, k - |\mathcal{I}| + i - 1\}$, the set $\{1, 2, \ldots, k\} \setminus \mathbb{F}_{i,j} \neq \emptyset$. Hence, mapping $\lambda$ is well defined. For this, observe that for every $i \in \{1, \ldots, |\mathcal{I}|\}$,

$$|\mathbb{F}_{i,1}| = |\mathcal{I}| - i + 1,$$

and for every $j \in \{2, \ldots, k - |\mathcal{I}| + i - 1\}$,

$$|\mathbb{F}_{i,j}| = |\mathbb{F}_{i,j-1}| + 1 = |\mathcal{I}| - i + j.$$

Hence, for every $i \in \{1, \ldots, |\mathcal{I}|\}$

$$\max_{j \in \{1,\ldots,k-|\mathcal{I}|+i-1\}} |\mathbb{F}_{i,j}| = |\mathbb{F}_{i,k-|\mathcal{I}|+i-1}| + 1 = k - 1.$$

The above analysis also shows that, $|\lambda(\mathcal{R})| = k$. Hence, the answer to the MC-WA problem $\{\vec{P}, \mathcal{R}, k\}$ is also *YES*.

Next assume that the answer to the MC-WA-BT problem $\{\vec{P}, \mathcal{R}, k\}$ is *YES*. Let $\lambda^*$ be a mapping as described in the definition of the MC-WA-BT problem. First observe that for any mapping $\lambda \in \Lambda_{\{\vec{P},\mathcal{R}\}}$, $|\lambda(\mathcal{R})| \geq k$. This is because

$$|\lambda(\mathcal{R})| \geq \left|\lambda\left(\bigcup_{m=1}^{k-1}\left\{\vec{R}_{|\mathcal{I}|,4_m}\right\} \cup \left\{\vec{R}_{|\mathcal{I}|,2}\right\}\right)\right| = k.$$

The equality is because in the set $\bigcup_{m=1}^{k-1}\{\vec{R}_{|\mathcal{I}|,4_m}\}\cup\{\vec{R}_{|\mathcal{I}|,2}\}$, there are exactly $k$ rooted subtrees and all of them collide on the directed edge $(v_{2|\mathcal{A}|+2|\mathcal{I}|-1}, v_{2|\mathcal{A}|+2|\mathcal{I}|}) \in E_{\vec{P}}$, therefore every mapping in the set $\Lambda_{\{\vec{P},\mathcal{R}\}}$ is forced to assign distinct values to all the rooted subtrees in the set. Since the mapping $\lambda^* \in \Lambda_{\{\vec{P},\mathcal{R}\}}$ is a certificate for MC-WA-BT problem $\{\vec{P}, \mathcal{R}, k\}$, $|\lambda^*(\mathcal{R})| = k$.

Observing all the collisions among the rooted subtrees on the directed edges $(v_{2|\mathcal{A}|+2|\mathcal{I}|-1}, v_{2|\mathcal{A}|+2|\mathcal{I}|})$ and $(v_{2|\mathcal{A}|+2|\mathcal{I}|-1}, v_{2|\mathcal{A}|+2|\mathcal{I}|-2})$, we note that the distinct rooted subtrees in the set $\bigcup_{m=1}^{k-1}\{\vec{R}_{|\mathcal{I}|,4_m}\}$ are assigned different values according to the mapping $\lambda^*$, and also $\lambda^*(\vec{R}_{|\mathcal{I}|,1}) = \lambda^*(\vec{R}_{|\mathcal{I}|,2}) \notin \lambda^*(\bigcup_{m=1}^{k-1}\{\vec{R}_{|\mathcal{I}|,4_m}|\})$. Continuing similar line of reasoning and observing for every $i \in \{|\mathcal{I}|, \ldots, 1\}$, pairs of directed edges $(v_{2|\mathcal{A}|+2i-1}, v_{2|\mathcal{A}|+2i})$, $(v_{2|\mathcal{A}|+2i-1}, v_{2|\mathcal{A}|+2i-2})$, and $(v_{-2i+2}, v_{-2i+1})$, $(v_{-2i+2}, v_{-2i+3})$, we have, for distinct $i, j \in \{|\mathcal{I}|, \ldots, 1\}$,

$$\lambda^*(\vec{R}_{i,1}) = \lambda^*(\vec{R}_{i,2}) = \lambda^*(\vec{R}_{i,3}). \tag{5.7}$$

and

$$\lambda^*(\vec{R}_{i,1}) \neq \lambda^*(\vec{R}_{j,1}). \tag{5.8}$$

Consider a mapping $\theta : \mathcal{A} \longrightarrow \mathbb{N}$ defined as

$$\theta((p_{l_i}, p_{r_i})) = \lambda^*(\vec{R}_{i,1}) \tag{5.9}$$

for $i \in \{1, \ldots, |\mathcal{I}|\}$, and

$$\theta((p_{l_i}, p_{r_i})) = \lambda^*(\vec{R}_i) \tag{5.10}$$

for $i \in \{|\mathcal{I}|+1, \ldots, |\mathcal{A}|\}$. First note that

$$|\theta(\mathcal{A})| \leq |\lambda^*(\mathcal{R})| = k.$$

Next we prove that $\theta$ is a mapping as defined in the definition of the ARC-COL problem. Suppose the arcs $(p_{l_i}, p_{r_i}), (p_{l_j}, p_{r_j}) \in \mathcal{A}$ overlap. If $(p_{l_i}, p_{r_i}), (p_{l_j}, p_{r_j}) \in \mathcal{I}$

(in which case, they necessarily overlap on arc $(p_{2|\mathcal{A}|}, p_1)$), then by equations (5.8) and (5.9), $\theta((p_{l_i}, p_{r_i})) \neq \theta((p_{l_j}, p_{r_j}))$. If $(p_{l_i}, p_{r_i}), (p_{l_j}, p_{r_j}) \in \mathcal{J}$, then by equation (5.10) and the fact that $\lambda^* \in \Lambda_{\{\vec{P},\mathcal{R}\}}$, $\theta((p_{l_i}, p_{r_i})) \neq \theta((p_{l_j}, p_{r_j}))$. If $(p_{l_i}, p_{r_i}) \in \mathcal{I}$ and $(p_{l_j}, p_{r_j}) \in \mathcal{J}$, then $(p_{l_i}, p_{r_i}), (p_{l_j}, p_{r_j})$'s overlap ensures that the rooted subtree $\vec{R}_j$ collides with at least one of the rooted subtree $\vec{R}_{i,2}, \vec{R}_{i,3}$. Hence, by equations (5.9), (5.10) and the fact that $\lambda^* \in \Lambda_{\{\vec{P},\mathcal{R}\}}$, $\theta((p_{l_i}, p_{r_i})) \neq \theta((p_{l_j}, p_{r_j}))$. This shows that mapping $\theta$ is indeed as described in the definition of the ARC-COL problem. Hence, the answer to the ARC-COL problem instance $\{C, \mathcal{A}, k\}$ is also *YES*.

This proves that the ARC-COL problem is reducible to the MC-WA-BT problem restricted to bipartite paths. Finally, applying Lemma 5.2 completes the proof. $\square$

## 5.4 Bidirected Trees

In this section, we prove that the problem CLIQUE-MC-WA-BT (defined next) restricted to the case where the degree of the bidirected tree is at most 3, is NP complete.

**Problem 5.7** (CLIQUE-MC-WA-BT). *Given a triple $\{\vec{T}_H, \mathcal{R}, k\}$, where $\vec{T}_H$ is a bidirected tree, $\mathcal{R}$ is a set of rooted subtrees on $\vec{T}_H$ and $k$ is a positive integer; consider the conflict graph $G_{\mathcal{R}}$ of the set $\mathcal{R}$ of rooted subtrees. Is there a set $\mathcal{C} \subseteq \mathcal{R}$ of rooted subtrees such that $G_{\mathcal{C}}$ is a clique, and $|\mathcal{C}| \geq k$?*

To show the NP completeness of CLIQUE-MC-WA-BT problem in the desired restricted settings, we first prove that the general CLIQUE-MC-WA-BT problem is in NP.

**Lemma 5.8.** *CLIQUE-MC-WA-BT is in NP.*

*Proof.* Given any instance of $\{\vec{T}_H, \mathcal{R}, k\}$, of the CLIQUE-MC-WA-BT problem, and any set $\mathcal{C} \subseteq \mathcal{R}$ of rooted subtrees, we can verify in $O\big(|E_{\vec{T}_H}||\mathcal{R}|^2\big)$ time, whether $\mathcal{C}$ is a certificate, as defined in the definition of the CLIQUE-MC-WA-BT problem, for the given instance of the problem or not. Hence, MC-WA-BT is in NP. $\square$

We prove the hardness result by reduction from the independent set problem in tripartite graphs (TRIPARTITE-IS). For completeness, the exact definition of the TRIPARTITE-IS problem is given next.

**Problem 5.9** (TRIPARTITE-IS)**.** *Given a pair $\{T, k\}$, where $T$ is a tripartite graph and $k$ is a positive integer. Is there an independent set $S \subseteq V_T$ such that $|S| \geq k$?*

It is known that TRIPARTITE-IS is NP complete [48].

**Theorem 5.10.** *CLIQUE-MC-WA-BT restricted to bidirected trees having degree at most* 3 *is NP complete.*

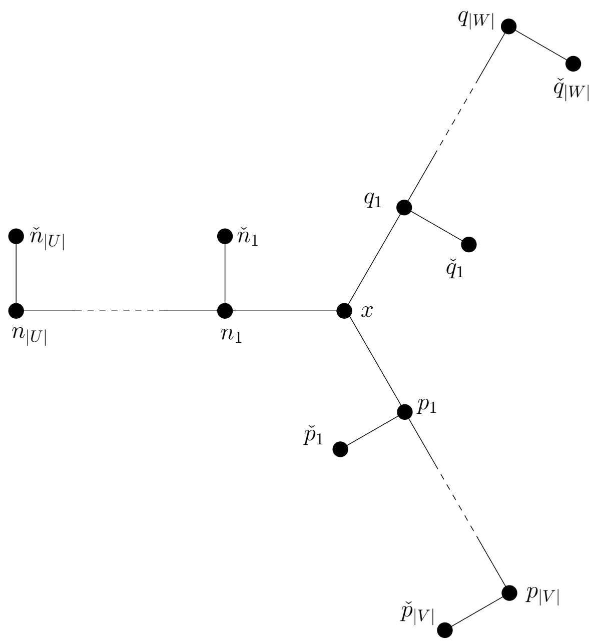
*Proof.* Let $\{T, k\}$ be any given instance of the TRIPARTITE-IS problem. We shall construct an instance $\{\vec{T}_H, \mathcal{R}, k\}$ of the CLIQUE-MC-WA-BT problem, where $\vec{T}_H$ is a bidirected tree having degree at most 3, $\mathcal{R}$ is a set of rooted subtrees on $\vec{T}_H$ such that the conflict graph $G_{\mathcal{R}}$ is isomorphic to the complementary tripartite graph $\bar{T}$. Since cliques in $\bar{T}$ are equivalent to independent sets in $T$, this would show that the TRIPARTITE-IS problem is reducible to the CLIQUE-MC-WA-BT problem restricted to bidirected trees having degree at most 3. Next, we present the construction of the required instance $\{\vec{T}_H, \mathcal{R}, k\}$ of the CLIQUE-MC-WA-BT problem.

By the definition of tripartite graphs, we can partition the vertex set $V_T$ into three independent sets [49]. Let these be

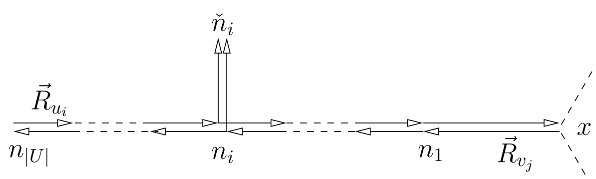
$$U := \{u_1, \ldots, u_{|U|}\}, \quad V := \{v_1, \ldots, v_{|V|}\}, \quad W := \{w_1, \ldots, w_{|W|}\}.$$

Obviously, the vertex sets $U$, $V$ and $W$ form cliques in the complementary tripartite graph $\bar{T}$. Also, they partition the set $V_{\bar{T}}$. We define the following four sets of vertices

$$N := \{n_1, \check{n}_1, \ldots, n_{|U|}, \check{n}_{|U|}\}, \qquad P := \{p_1, \check{p}_1, \ldots, p_{|V|}, \check{p}_{|V|}\},$$
$$Q := \{q_1, \check{q}_1, \ldots, q_{|W|}, \check{q}_{|W|}\}, \qquad X := \{x\}.$$




(a) Host tree $H$ used to generated the bidirected tree $\vec{T}_H$.




(b) Structure of rooted subtrees $\vec{R}_{u_i}, \vec{R}_{v_j}$ on the directed edge set $E_{\vec{T}_H[N \cup X]}$.

Figure 5.3: Construction of an instance of CLIQUE-MC-WA-BT restricted to bidirected trees having degree at most 3, equivalent to a given instance of ARC-COL.

Next, we construct a graph $H$ having the vertex set and the edge set defined as

$$\begin{aligned}
V_H :=& X \cup N \cup P \cup Q, \\
E_H :=& \big\{\{n_i, n_{i+1}\} : i \in \{1, \ldots, |U|-1\}\big\} \cup \big\{\{n_i, \check{n}_i\} : i \in \{1, \ldots, |U|\}\big\} \\
& \cup \big\{\{p_i, p_{i+1}\} : i \in \{1, \ldots, |V|-1\}\big\} \cup \big\{\{p_i, \check{p}_i\} : i \in \{1, \ldots, |V|\}\big\} \\
& \cup \big\{\{q_i, q_{i+1}\} : i \in \{1, \ldots, |W|-1\}\big\} \cup \big\{\{q_i, \check{q}_i\} : i \in \{1, \ldots, |W|\}\big\} \\
& \cup \big\{\{x, n_1\}, \{x, p_1\}, \{x, q_1\}\big\}.
\end{aligned}$$

Observe that $H$ is a tree with degree $\Delta_H = 3$. We construct the bidirected tree $\vec{T}_H$ from the tree $H$ as the host. Hence, the degree of the bidirected tree $\vec{T}_H$ is also equal to 3.

Next, we construct three sets of rooted subtrees of $\vec{T}_H$ denoted as

$$\mathcal{R}_U := \{\vec{R}_{u_1}, \ldots, \vec{R}_{u_{|U|}}\}, \quad \mathcal{R}_V := \{\vec{R}_{v_1}, \ldots, \vec{R}_{v_{|V|}}\}, \quad \mathcal{R}_W := \{\vec{R}_{w_1}, \ldots, \vec{R}_{w_{|W|}}\}.$$

We shall describe the vertex sets and the directed edge sets for these sets of rooted subtrees, in a moment. Let us define the set $\mathcal{R}$ of rooted subtrees to be

$$\mathcal{R} := \mathcal{R}_U \cup \mathcal{R}_V \cup \mathcal{R}_W.$$

From the notations used for the rooted subtrees in the sets $\mathcal{R}_U$, $\mathcal{R}_V$ and $\mathcal{R}_W$, and the vertices in the sets $U$, $V$ and $W$, observe that there is an obvious bijection between the sets $\mathcal{R}$ and $V_{\bar{T}}$. Note that we can partition the set of edges $E_{\bar{T}}$ into six subsets $E_{\bar{T}[U]}$, $E_{\bar{T}[V]}$, $E_{\bar{T}[W]}$, $E_{U,V}$, $E_{V,W}$ and $E_{W,U}$. As described before, $\bar{T}[U]$, $\bar{T}[V]$ and $\bar{T}[W]$ are cliques. The other three edge sets are defined as

$$\begin{aligned}
E_{U,V} &:= \big\{\{u, v\} \in E_{\bar{T}} : u \in U \text{ and } v \in V\big\}, \\
E_{V,W} &:= \big\{\{v, w\} \in E_{\bar{T}} : v \in V \text{ and } w \in W\big\}, \\
E_{W,U} &:= \big\{\{w, u\} \in E_{\bar{T}} : w \in W \text{ and } u \in U\big\}.
\end{aligned}$$

Observe that set $E_{\vec{T}_H}$ of directed edges can be partitioned into three subsets $E_{\vec{T}_H[N \cup X]}$, $E_{\vec{T}_H[P \cup X]}$ and $E_{\vec{T}_H[Q \cup X]}$. We shall study these three directed edge sets one at a time, and describe which of the directed edges in each of these sets are contained in the rooted subtrees in the set $\mathcal{R}$. The vertex set of each rooted subtree can then be determined from its set of directed edges.

We start with the set $E_{\vec{T}_H[N \cup X]}$. For every $\vec{R} \in \mathcal{R}_U$, we have

$$\big\{(n_{i+1}, n_i) : i \in \{1, \ldots, |U|\}\big\} \cup \big\{(n_1, x)\big\} \subseteq E_{\vec{R}}. \tag{5.11}$$

For every $\vec{R} \in \mathcal{R}_V$, we have

$$\big\{(n_i, n_{i+1}) : i \in \{1, \ldots, |U|\}\big\} \cup \big\{(x, n_1)\big\} \subseteq E_{\vec{R}}. \tag{5.12}$$

For every $i \in \{1, \ldots, |U|\}$, we have

$$(n_i, \check{n}_i) \in E_{\vec{R}_{u_i}}. \tag{5.13}$$

Finally, for every $i \in \{1, \ldots, |U|\}$ and for every $j \in \{1, \ldots, |V|\}$, if there is an edge $\{u_i, v_j\} \in E_{\bar{T}}$, we have

$$(n_i, \check{n}_i) \in E_{\vec{R}_{v_j}}. \tag{5.14}$$

Observe that according to equation (5.11), all the rooted subtrees in the set $\mathcal{R}_U$ collide, therefore the set $\mathcal{R}_U$ forms a clique in the conflict graph $G_{\mathcal{R}}$. Similarly, according to equation (5.12), all the rooted subtrees in the set $\mathcal{R}_V$ collide, therefore the set $\mathcal{R}_V$ forms a clique in the conflict graph $G_{\mathcal{R}}$. Recall that these cliques are desired since the corresponding vertex sets $U$ and $V$ form cliques in the graph $\bar{T}$. Next, observe that according to equations (5.13) and (5.14), if there is an edge $\{u_i, v_j\} \in E_{\bar{T}}$, then the corresponding rooted subtrees $\mathcal{R}_{u_i}$ and $\mathcal{R}_{v_j}$ collide on the directed edge $(n_i, \check{n}_i)$, hence the edge $\{\mathcal{R}_{u_i}, \mathcal{R}_{v_j}\}$ exists in the conflict graph $G_{\mathcal{R}}$. Moreover, if vertices $u_i$ and $v_j$ are independent in the graph $\bar{T}$, then the corresponding rooted subtrees $\mathcal{R}_{u_i}$ and $\mathcal{R}_{v_j}$ are also independent in the conflict graph $G_{\mathcal{R}}$. Hence, the graph $G_{\mathcal{R}_U \cup \mathcal{R}_V}$ is isomorphic to the complementary bipartite graph $\bar{T}[U \cup V]$. An example construction is presented in Figure 5.3. In Figure 5.3(a), we present the host tree $H$ that is used to generate the bidirected tree $\vec{T}_H$. In Figure 5.3(b), we present the rooted subtrees $\vec{R}_{u_i}$ and $\vec{R}_{v_j}$ corresponding to the vertices $u_i \in U$ and $v_j \in V$ such that the edge $\{u_i, v_j\}$ is in the complementary tripartite graph $\bar{T}$. We only show the structure of the rooted subtree $\vec{R}_{u_i}$ on the directed edge set $E_{\vec{T}_H[N \cup X]}$, and the structure of the rooted subtree $\vec{R}_{v_j}$ on the directed edge set $E_{\vec{T}_H[\{x, \check{n}_i, n_1, \ldots, n_{|U|}\}]}$. For clarity, in the figure we only show the structure of the rooted subtrees on the interesting directed edges, and do not show all the directed edges and vertices of $\vec{T}_H$. Also, we only annotate the set of interesting vertices of $\vec{T}_H$.

Arguing similarly for the sets $E_{\vec{T}_H[P \cup X]}$ and $E_{\vec{T}_H[Q \cup X]}$, we get the complete characterization of the rooted subtrees in the set $\mathcal{R}$. We advise the reader to check that the directed graphs in the set $\mathcal{R}$ are indeed rooted subtrees of the bidirected tree $\vec{T}_H$. More specifically, the rooted subtrees in the sets $\mathcal{R}_U$, $\mathcal{R}_V$ and $\mathcal{R}_W$ have the vertices $n_U$, $p_V$ and $q_W$ as the roots. Moreover, with a little effort, we note that the conflict graph $G_{\mathcal{R}}$ is indeed isomorphic to the complementary tripartite graph $\bar{T}$.

This completes the characterization of the required problem instance $\{\vec{T}_H, \mathcal{R}, k\}$ of the TRIPARTITE-IS problem that we set out to construct. Finally, applying Lemma 5.8 completes the proof. $\square$

# Chapter 6

# Multicast Traffic Grooming in Unidirectional Rings

As stated in Section 1.2.2, we are interested in the problem of grooming multicast traffic in all-optical unidirectional ring networks. In this chapter, we define the exact problem that we wish to study. We also present the work that is closely related to our problem of interest.

## 6.1 Model

As described in Section 1.2.2, a unidirectional ring is the directed graph $\vec{C}$ having vertex set $V_{\vec{C}} = \{v_1, v_2 \ldots, v_{|V_{\vec{C}}|}\}$ and edge set $E_{\vec{C}} = \{(v_1, v_2), \ldots, (v_{|V_{\vec{C}}|}, v_1)\}$. The skeleton $\|\vec{C}\|$ of the unidirectional ring $\vec{C}$ is a cycle of size $|V_{\vec{C}}|$.

Observe that the set of leaves of any rooted subtree of a unidirectional ring is singleton. Hence, as described in Section 1.3, a rooted subtree of a unidirectional ring is nothing but a directed path on the ring.

Consider a triple $\{\vec{C}, \mathcal{M}, g\}$ where $\vec{C}$ is a unidirectional ring that models the fiber network, $\mathcal{M}$ is a set of multicast traffic requests on $\vec{C}$ and $g$ is the grooming ratio as defined in Section 1.4.3. Note that for a multicast traffic request $\{s, D\}$ on a unidirectional ring $\vec{C}$, there is a unique directed path on the unidirectional ring that satisfies the properties stated in Section 1.4.2, i.e, the set $\mathcal{R}_{\{\vec{C},\{s,D\}\}}$ of interesting directed paths defined in Section 1.4.2, contains exactly one directed path. Hence, the routing of every multicast traffic request in the set $\mathcal{M}$ is fixed, i.e., the mapping $\pi$ described in MC-TG in Section 1.4.3, is trivially determined. As a consequence, the traffic grooming problem MIN-WAVE-MC-TG simply reduces to a problem of assigning wavelengths and sub-wavelength channels to the set of directed paths on the given unidirectional ring, corresponding to the given set of multicast traffic requests. On the other hand, the traffic grooming problem MIN-ADM-MC-TG also reduces to assigning wavelengths and sub-wavelength channels to the set of directed paths on the unidirectional ring; but in this case, unlike the MIN-WAVE-MC-TG problem, corresponding to each directed path we also have a set of ring vertices that act as either the source or the destinations of the corresponding multicast traffic request.

Observe that the start vertex of the directed path corresponding to any multicast traffic request $\{s, D\}$ on a unidirectional ring $\vec{C}$, is the source node $s$ and the end vertex is a node from the set of destinations $D$. This end vertex is referred to as the *final destination* of the multicast traffic request $\{s, D\}$ and is denoted by $d$. All the other destinations in the set $D \setminus \{d\}$ are referred to as the *intermediate destinations*.

## 6.2 Problem Statement

As discussed in Section 6.1, grooming a given set of multicast traffic requests on a unidirectional ring network is equivalent to assigning wavelengths and sub-wavelength channels to the set of rooted subtrees of the unidirectional ring corresponding to the given set of multicast traffic requests. More precisely, grooming a set of multicast traffic requests on a unidirectional ring network can be modeled as follows.

**Definition 6.1** (MC-TG-UR). *Given a triple $\{\vec{C}, \mathcal{R}, g\}$, where $\vec{C}$ is a unidirectional ring, $\mathcal{R}$ is a set of directed paths on $\vec{C}$ and $g$ is a positive integer; a traffic grooming solution is a pair of mappings $\{\lambda, \omega\}$ as described next.*

*(i) Mapping $\lambda : \mathcal{R} \longrightarrow \mathbb{N}$ solves the wavelength assignment problem in the sense that it maps each directed path $\vec{R} \in \mathcal{R}$ to a wavelength (described as a positive integer).*

*(ii) Mapping $\omega : \mathcal{R} \longrightarrow \mathbb{N}$ solves the sub-wavelength channel assignment problem in the sense that it maps each directed path $\vec{R} \in \mathcal{R}$ to a sub-wavelength channel (described as a positive integer).*

*(iii) Jointly the two mappings should satisfy the constraints that for every pair of rooted subtrees $\vec{R}_i, \vec{R}_j \in \mathcal{R}$, if they collide, then $(\lambda(\vec{R}_i), \omega(\vec{R}_i)) \neq (\lambda(\vec{R}_j), \omega(\vec{R}_j))$; and the number of sub-wavelength channels in any wavelength must not exceed $g$, i.e., $max_{k \in \lambda(\mathcal{R})} |\omega(\{\vec{R} \in \mathcal{R} : \lambda(\vec{R}) = k\})| \leq g$.*

*We denote the set of all such pairs of mappings $\{\lambda, \omega\}$ for the triple $\{\vec{C}, \mathcal{R}, g\}$ by $\Xi_{\{\vec{C}, \mathcal{R}, g\}}$.*

Using the definition presented above, we define the problem of grooming a set of multicast traffic requests on a unidirectional ring with the objective of minimizing the total number of wavelengths used, as follows.

**Problem 6.2** (MIN-WAVE-MC-TG-UR). *Given a triple $\{\vec{C}, \mathcal{R}, g\}$, where $\vec{C}$ is a unidirectional ring, $\mathcal{R}$ is a set of directed paths on $\vec{C}$ and $g$ is a positive integer; determine a traffic grooming solution $\{\lambda, \omega\} \in \Xi_{\{\vec{C}, \mathcal{R}, g\}}$ as defined in MC-TG-UR with the objective of minimizing $|\lambda(\mathcal{R})|$, the total number of wavelengths required.*

An interesting observation is that the problems of grooming unicast and multicast traffic requests on all-optical unidirectional rings with the objective of minimizing the total number of wavelengths are equivalent. This is because modeling multicast traffic requests as directed paths on a unidirectional ring only preserves the information about the source nodes and the final sink nodes of the multicast traffic requests. Consider a set of unicast traffic requests having the source and the destination nodes coinciding with the source and the final destination nodes of the given set of multicast traffic requests. Observe that the problem of grooming these two sets of traffic requests on the given unidirectional ring with the objective of minimizing the number of wavelengths used, is exactly the same.

Next we define the problem of grooming a set of multicast traffic requests on a unidirectional ring, with the objective of minimizing the total number of ADMs used.

**Problem 6.3** (MIN-ADM-MC-TG-UR). *Given a triple* $\{\vec{C}, \mathcal{M}, g\}$, *where* $\vec{C}$ *is a unidirectional ring,* $\mathcal{M}$ *is a set of multicast traffic requests on* $\vec{C}$ *and* $g$ *is a positive integer; let* $\mathcal{R}$ *be the set of directed paths corresponding to the multicast traffic requests in the set* $\mathcal{M}$ *and let* $\vec{R}_{\{s,D\}}$ *denote the directed path corresponding to any multicast traffic request* $\{s, D\} \in \mathcal{M}$. *For the triple* $\{\vec{C}, \mathcal{R}, g\}$, *determine a traffic grooming solution* $\{\lambda, \omega\} \in \Xi_{\{\vec{C},\mathcal{R},g\}}$ *as defined in MC-TG-UR, with the objective of minimizing the total number of ADMs required. The number of ADMs required by the traffic grooming solution* $\{\lambda, \omega\}$ *can be determined as* $\sum_{v \in V_{\vec{G}}} |\lambda(\mathcal{M}_v)|$, *where, as defined in the problem MIN-ADM-MC-TG described in Section 1.4.3, for any vertex* $v \in V_{\vec{G}}$, $\mathcal{M}_v$ *is the set of all the multicast traffic requests that have vertex* $v$ *as the source node or as one of the destination nodes, i.e.,* $\mathcal{M}_v := \{\{s, D\} \in \mathcal{M} : v \in \{s\} \cup D\}$.

The total number of ADMs required by any traffic grooming solution in the MIN-ADM-MC-TG-UR problem is calculated exactly as in the problem MIN-ADM-MC-TG described in Section 1.4.3.

## 6.3 Related Work

Grooming static unicast sub-wavelength traffic to minimize either the number of ADMs or the number of wavelengths required per fiber in WDM ring networks is a well studied problem [11][50][51][52]. Different traffic scenarios such as uniform all-to-all traffic [51][53], distance dependent traffic [50] and non-uniform traffic [11][54] have been studied. Work has also been done with other cost functions such as the overall network cost [55], which includes the cost of transceivers, wavelengths and the number of required hops. Recently there has been a lot of work on grooming both static [56] as well as dynamic [57][58][59] traffic in mesh networks.

The past few years have seen a spurt of research in the problem of grooming multicast traffic in WDM networks. And although a lot of literature is available, not many results are known for the multicast traffic grooming problem. Most of the work in the multicast case has focused on heuristics for grooming multicast traffic in WDM mesh networks under non-uniform static [60] as well as dynamic traffic [61][62][63][64][65][66] scenarios. Although multicast traffic grooming in mesh WDM networks is a general case of the same problem in WDM rings, the ideas that are applied for mesh networks in [60][61][62][63][64][65][66] are not very attractive for unidirectional rings. The difference between the mesh and the unidirectional ring case is that, in mesh networks there are many possible routings for each traffic demand whereas in unidirectional rings the routing is fixed and we have control over wavelength assignment only. All of the heuristics for grooming multicast traffic in mesh networks take advantage of the multiple routings possible and the wavelength assignment is usually trivial (first fit). This is clearly not desired for grooming in unidirectional rings, since the routing is already fixed and the only way to effectively groom traffic is by using intelligent wavelength assignment.

Although, most of the work on multicast traffic grooming looks at mesh WDM networks, there has been some work in the case of WDM rings also. More specifically, in [67] the authors look at the problem of grooming given multicast traffic demands in a bidirectional WDM ring. They present a heuristic algorithm inspired by the algorithm to groom unicast traffic demands on WDM rings given in [11]. The problem that we study here is somewhat different from the problem studied in [67]. The main difference, other than the fact that we study unidirectional rings while [67] looks at bidirectional rings, is that the cost function used is different. We consider the number of ADMs and the number of wavelengths required per fiber as our cost, whereas in [67], the total number of ports of *e-DAC* nodes in the network is considered as the cost. In [67], the authors define two different types of nodes, *o-DAC* and *e-DAC* nodes. When all the traffic on all the incoming wavelengths needs to be forwarded, o-DAC nodes are used since the splitting can be done in the optical domain. If this is not the case then e-DAC nodes are used. Note that the cost functions are not the same since we require ADMs at all the nodes where some traffic needs to be dropped whereas in [67], even the nodes where there is some drop traffic can be treated as o-DAC nodes. Another important difference is that, unlike us, the authors in [67] do not consider all-optical networking.

# Chapter 7

## Algorithms for Multicast Traffic Grooming in Unidirectional Rings

In this chapter, we study the MIN-WAVE-MC-TG-UR and the MIN-ADM-MC-TG-UR problems described in Section 6.2.

We show that any 'good' circular arc graph coloring algorithm can be used to generate a 'good' traffic grooming solution for the MIN-WAVE-MC-TG-UR problem. For the MIN-ADM-MC-TG-UR problem, we analyze the worst case performances of several very simple traffic grooming strategies. We also present a new traffic grooming heuristic and study its performance by simulations.

### 7.1 Minimizing Wavelengths

We start with the MIN-WAVE-MC-TG-UR problem. We discuss how it is related to the problem of vertex coloring in circular arc graphs, and how we can use the approximation algorithms developed for the circular arc graph coloring problem for the MIN-WAVE-MC-TG-UR problem.

### 7.1.1 Relation to Vertex Coloring

Given an instance $\{\vec{C}, \mathcal{R}, g\}$ of the MIN-WAVE-MC-TG-UR problem described in Section 6.2, consider the conflict graph $G_{\mathcal{R}}$ corresponding to the set $\mathcal{R}$ of directed paths on the unidirectional ring $\vec{C}$. The conflicts modeled in this graph correspond to all the pairwise collisions between the directed paths in the set $\mathcal{R}$, i.e., for any pair of directed paths $\vec{R}_i, \vec{R}_j \in \mathcal{R}$, there is an edge $\{\vec{R}_i, \vec{R}_j\} \in E_{G_{\mathcal{R}}}$ in the conflict graph if and only if they collide and therefore, cannot be assigned the same sub-wavelength channel on the same wavelength. Assuming the sub-wavelength channels in distinct wavelengths to be distinct, it is straightforward to argue that grooming the set $\mathcal{R}$ of directed paths on the unidirectional ring $\vec{C}$ in order to minimize the total number of sub-wavelength channels used is equivalent to the problem of finding a minimum vertex coloring of the corresponding conflict graph $G_{\mathcal{R}}$, where each color signifies a sub-wavelength channel. Moreover, observe that since each wavelength supports a maximum of $g$ sub-wavelength channels, the minimum number of wavelengths required by the instance $\{\vec{C}, \mathcal{R}, g\}$ of the MIN-WAVE-MC-TG-UR problem is given by

$$\min_{\{\lambda,\omega\}\in\Xi_{\{\vec{C},\mathcal{R},g\}}} |\lambda(\mathcal{R})| = \left\lceil \frac{\chi_{G_{\mathcal{R}}}}{g} \right\rceil, \tag{7.1}$$

where $\chi_{G_{\mathcal{R}}}$ is the chromatic number of the conflict graph $G_{\mathcal{R}}$ and $\Xi_{\{\vec{C},\mathcal{R},g\}}$ is the set of all the possible traffic grooming solutions for the triple $\{\vec{C}, \mathcal{R}, g\}$ as defined in MC-TG-UR described in Section 6.2.

### 7.1.2 NP Completeness

Consider the problem of coloring a set of arcs of a circle, described as the ARC-COL problem in Section 5.3. Corresponding to any instance of the ARC-COL problem, we can construct an equivalent instance of the decision version of the MIN-WAVE-MC-TG-UR problem in polynomial time. Since ARC-COL is NP complete, MIN-WAVE-MC-TG-UR is NP hard.

### 7.1.3 Approximation Algorithms

Since the MIN-WAVE-MC-TG-UR problem is NP hard, it makes sense to study approximation algorithms for the problem. As described above, for an instance $\{\vec{C}, \mathcal{R}, g\}$ of the MIN-WAVE-MC-TG-UR problem, a vertex coloring of the conflict graph $G_{\mathcal{R}}$ trivially determines a grooming solution (partitioning the colors into sets of size $g$ and treating each set as a wavelength and the colors as the sub-wavelength channels). Such a scheme is presented as Algorithm 6 (ARC-COL-BASED-TG). Let us denote the traffic grooming solution generated by ARC-COL-BASED-TG by the pair $\{\lambda^{\text{ARC}}, \omega^{\text{ARC}}\}$. Observe that ARC-COL-BASED-TG requires an algorithm ARC-COL-ALGO for coloring circular arc graphs. If ARC-COL-ALGO is an approximation algorithm for the problem of minimum vertex coloring of circular arc graphs, we can prove the following theorem for ARC-COL-BASED-TG.

**Theorem 7.1.** *If ARC-COL-ALGO is an $\alpha$-approximation algorithm for the problem of minimum vertex coloring of circular arc graphs, the total number of wavelengths required by the grooming solution generated by ARC-COL-BASED-TG for a given MIN-WAVE-MC-TG-UR problem instance $\{\vec{C}, \mathcal{R}, g\}$ is bounded as*

$$|\lambda^{\text{ARC}}(\mathcal{R})| \leq \alpha\Big(\min_{\{\lambda,\omega\}\in\Xi_{\{\vec{C},\mathcal{R},g\}}} |\lambda(\mathcal{R})|\Big) + 1$$

*Proof.* The total number of wavelengths required by the traffic grooming solution generated by ARC-COL-BASED-TG can be upper bounded as

$$|\lambda^{\text{ARC}}(\mathcal{R})| = \left\lceil \frac{|\psi^{\text{ARC}}(\mathcal{R})|}{g} \right\rceil \leq \left\lceil \frac{\alpha \cdot \chi_{G_{\mathcal{R}}}}{g} \right\rceil \leq \alpha \left\lceil \frac{\chi_{G_{\mathcal{R}}}}{g} \right\rceil + 1 \tag{7.2}$$

The first equality follows from the workings of the ARC-COL-BASED-TG scheme and the first inequality follows from the fact that ARC-COL-ALGO is an $\alpha$-approximation algorithm for the problem of minimum vertex coloring of circular arc graphs. The result follows from equations (7.1) and (7.2). ∎

For completion, we review the best approximation algorithms for coloring circular arc graphs that are available in the literature. Kumar et. al. [68] give a randomized algorithm with approximation ratio $(1 + \frac{1}{e} + o(1))$ for instances of the problem needing at least $\omega(\ln(n))$ colors, where $n$ is the number of arcs to be colored. In [69], Karapetian et. al. present a $\frac{3}{2}$-approximation algorithm for circular arc coloring. The lower bound on chromatic number used in the analysis of Karapetian's algorithm, is the clique number. This along with the fact that the approximation ratio of $\frac{3}{2}$ is strict, suggests that it might not be easy to design deterministic coloring algorithms with better approximation ratios.

**Subroutine 6** ARC-COL-BASED-TG

**Require:** MIN-WAVE-MC-TG-UR problem instance $\{\vec{C}, \mathcal{R}, g\}$ and an algorithm ARC-COL-ALGO for vertex coloring circular arc graphs.

**Ensure:** A traffic grooming solution $\{\lambda^{\text{ARC}}, \omega^{\text{ARC}}\} \in \Xi_{\{\vec{C},\mathcal{R},g\}}$.

1: Determine the conflict graph $G_{\mathcal{R}}$.

2: Using ARC-COL-ALGO, determine a vertex coloring $\psi^{\text{ARC}}$ for the conflict graph $G_{\mathcal{R}}$.
{The conflict graph is a circular arc graph.}

3: Partition $\mathcal{R}$ into subsets $\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{\lceil \frac{|\psi^{\text{ARC}}(\mathcal{R})|}{g} \rceil}\}$ such that the following hold:

  (i) For every $\mathcal{P}_i$, $|\psi^{\text{ARC}}(\mathcal{P}_i)| \leq g$.

  (ii) For every pair $\mathcal{P}_i, \mathcal{P}_j$, $\psi^{\text{ARC}}(\mathcal{P}_i) \cap \psi^{\text{ARC}}(\mathcal{P}_j) = \emptyset$.

4: **for all** sets $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{\lceil \frac{|\psi^{\text{ARC}}(\mathcal{R})|}{g} \rceil}$ **do**

5: $\quad \lambda^{\text{ARC}}(\vec{R}) \leftarrow i$ for every $\vec{R} \in \mathcal{P}_i$

6: **end for**

7: **for all** $\vec{R} \in \mathcal{R}$ **do**

8: $\quad \omega^{\text{ARC}}(\vec{R}) \leftarrow \min\{k \in \mathbb{N} : \nexists \vec{S} \in \mathcal{R}$ such that $\lambda^{\text{ARC}}(\vec{R}) = \lambda^{\text{ARC}}(\vec{S})$ and $\omega^{\text{ARC}}(\vec{S}) = k\}$

9: **end for**

## 7.2 Minimizing ADMs: Bounds and Simple Schemes

In this section we present and analyze some very simple schemes for the MIN-ADM-MC-TG-UR problem described in Section 6.2. We start by developing a lower bound on the number of ADMs required by any traffic grooming scheme for a given instance of the MIN-ADM-MC-TG-UR problem. This lower bound acts as a benchmark against which we compare the various traffic grooming schemes.

### 7.2.1 Lower Bound

Consider an instance $\{\vec{C}, \mathcal{M}, g\}$ of the MIN-ADM-MC-TG-UR problem where $\vec{C}$ is a unidirectional ring, $\mathcal{M}$ is a set of multicast traffic requests on $\vec{C}$ and $g$ is the grooming ratio. Observe that the total number of ADMs required by any traffic grooming solution for the MIN-ADM-MC-TG-UR problem is determined by summing the ADMs required at each vertex in the unidirectional ring. Hence, a lower bound for the MIN-ADM-MC-TG-UR problem can simply be obtained by determining lower bounds on the number of ADMs required at each vertex of the unidirectional ring and summing over all the ring vertices. As per our notation, for any vertex $v \in V_{\vec{C}}$, $\mathcal{M}_v$ is the set of all the multicast traffic requests that have the ring vertex $v$ as either the source node or as one of the destination nodes. Let us denote the set of directed paths corresponding to the multicast traffic requests in the set $\mathcal{M}_v$ by $\mathcal{R}_v$. We claim that the minimum number of ADMs required by any traffic grooming solution, at any ring vertex $v \in V_{\vec{C}}$, is equal to $\left\lceil \frac{\chi_{G_{\mathcal{R}_v}}}{g} \right\rceil$, where $\chi_{G_{\mathcal{R}_v}}$ is the chromatic number of the conflict graph $G_{\mathcal{R}_v}$ of the set $\mathcal{R}_v$ of directed paths on the unidirectional ring $\vec{C}$. To observe this claim, we note that since all the multicast traffic requests $\mathcal{M}_v$ corresponding to the set of directed paths $\mathcal{R}_v$ have the ring vertex $v$ as either the source or as one of the destinations, $v$ must be equipped with ADMs corresponding to all the wavelengths on which any of these requests are groomed.

Hence, in order to use the minimum number of ADMs at ring vertex $v$ (irrespective of the number of ADMs required at other vertices of the ring), we need to groom the set $\mathcal{M}_v$ of multicast traffic requests represented by the set $\mathcal{R}_v$ of directed paths, on as few wavelengths as possible. The claim follows using the arguments presented in Section 7.1 for establishing equation (7.1). Hence, the minimum number of ADMs required by the instance $\{\vec{C}, \mathcal{M}, g\}$ of the MIN-ADM-MC-TG-UR problem is lower bounded as

$$\min_{\{\lambda,\omega\}\in\Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v\in V_{\vec{C}}} |\lambda(\mathcal{M}_v)| \geq \sum_{v\in V_{\vec{C}}} \left\lceil \frac{\chi_{G_{\mathcal{R}_v}}}{g} \right\rceil . \tag{7.3}$$

Moreover, due to Lemma 7.2, we claim that the lower bound described above is easy to calculate.

**Lemma 7.2.** *Consider a unidirectional ring $\vec{C}$ and a set of multicast traffic requests $\mathcal{M}_v$ containing the ring vertex $v \in V_{\vec{C}}$ as either the source or as one of the destinations. The conflict graph $G_{\mathcal{R}_v}$ of the set $\mathcal{R}_v$ of directed paths corresponding to the multicast traffic requests $\mathcal{M}_v$ on $\vec{C}$, is a complementary bipartite graph.*

*Proof.* We define $\mathcal{M}_{v=s}$ to be the set of multicast traffic requests having the ring vertex $v$ as the source node, i.e., $\mathcal{M}_{v=s} := \{\{s, D\} \in \mathcal{M}_v : v = s\}$. Similarly, we define $\mathcal{M}_{v=d}$ to be the set of multicast traffic requests having the ring vertex $v$ as the final destination, i.e., $\mathcal{M}_{v=d} := \{\{s, D\} \in \mathcal{M}_v : v = d\}$. Clearly, $\mathcal{M}_{v=s}, \mathcal{M}_{v=d} \subseteq \mathcal{M}_v$, and $\mathcal{M}_{v=s} \cap \mathcal{M}_{v=d} = \emptyset$. Let the set of corresponding directed paths be $\mathcal{R}_{v=s}$ and $\mathcal{R}_{v=d}$. Since $\vec{C}$ is a unidirectional ring, $\delta^{\mathrm{i}}_{\vec{C}}(v) = \delta^{\mathrm{o}}_{\vec{C}}(v) = 1$. Therefore, without loss of generality, we assume that $(u, v), (v, w) \in E_{\vec{C}}$ are the only two directed edges adjacent to $v$. Observe that the set of directed paths $\mathcal{R}_{v=s}$ collide on the directed edge $(v, w)$, the set of directed paths $\mathcal{R}_{v=d}$ collide on the directed edge $(u, v)$, and the set of directed paths $\mathcal{R}_v \setminus (\mathcal{R}_{v=s} \cup \mathcal{R}_{v=d})$ collide on both the directed edges $(u, v)$ and $(v, w)$. Moreover, every directed path in the set $\mathcal{R}_v \setminus (\mathcal{R}_{v=s} \cup \mathcal{R}_{v=d})$ collides with every directed path in the set $\mathcal{R}_{v=s} \cup \mathcal{R}_{v=d}$ on one of the two directed edges $(u, v)$ and $(v, w)$. Consequently, the induced subgraphs $G_{\mathcal{R}_v}[\mathcal{R}_v \setminus \mathcal{R}_{v=s}]$ and $G_{\mathcal{R}_v}[\mathcal{R}_v \setminus \mathcal{R}_{v=d}]$ of the conflict graph $G_{\mathcal{R}_v}$ are cliques. Hence, the conflict graph $G_{\mathcal{R}_v}$ is a complementary bipartite graph. $\square$

### 7.2.2 Worst Case

Next we investigate the absolute worst that any traffic grooming solution can perform for the MIN-ADM-MC-TG-UR problem. The maximum number of ADMs is required when we use a different wavelength for each multicast traffic request, i.e., we do no traffic grooming and wavelength reuse. Hence, given a MIN-ADM-MC-TG-UR problem instance $\{\vec{C}, \mathcal{M}, g\}$, the absolute worst that any traffic grooming solution can do in terms of the number of ADMs required, is upper bounded as

$$\max_{\{\lambda,\omega\}\in\Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v\in V_{\vec{C}}} |\lambda(\mathcal{M}_v)| \leq \sum_{v\in V_{\vec{C}}} |\mathcal{M}_v|, \tag{7.4}$$

where, as defined before, $\mathcal{M}_v$ is the set of multicast traffic requests having the ring vertex $v$ as either the source node or as one of the destination nodes.

Consider the conflict graph $G_{\mathcal{R}_v}$ of the set $\mathcal{R}_v$ of directed paths, corresponding to the multicast traffic requests $\mathcal{M}_v$ on $\vec{C}$. According to the proof of Lemma 7.2, the induced subgraphs $G_{\mathcal{R}_v}[\mathcal{R}_v \setminus \mathcal{R}_{v=s}]$ and $G_{\mathcal{R}_v}[\mathcal{R}_v \setminus \mathcal{R}_{v=d}]$ of the conflict graph $G_{\mathcal{R}_v}$ are cliques. Hence, its chromatic number is lower bounded as

$$\chi_{G_{\mathcal{R}_v}} \geq \max\left\{|\mathcal{R}_v \setminus \mathcal{R}_{v=s}|, |\mathcal{R}_v \setminus \mathcal{R}_{v=d}|\right\} \geq \frac{|\mathcal{R}_v|}{2} = \frac{|\mathcal{M}_v|}{2}. \tag{7.5}$$

The second inequality is due to the facts that $\mathcal{R}_{v=s}, \mathcal{R}_{v=d} \subseteq \mathcal{R}_v$ and $\mathcal{R}_{v=s} \cap \mathcal{R}_{v=d} = \emptyset$. By equations (7.3), (7.4) and (7.5),

$$\begin{aligned}\max_{\{\lambda,\omega\}\in\Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v\in V_{\vec{C}}} |\lambda(\mathcal{M}_v)| &\leq \sum_{v\in V_{\vec{C}}} |\mathcal{M}_v| \leq \sum_{v\in V_{\vec{C}}} 2\chi_{G_{\mathcal{R}_v}} \leq 2g \sum_{v\in V_{\vec{C}}} \left\lceil \frac{\chi_{G_{\mathcal{R}_v}}}{g} \right\rceil \\ &\leq 2g\Big(\min_{\{\lambda,\omega\}\in\Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v\in V_{\vec{C}}} |\lambda(\mathcal{M}_v)|\Big)\end{aligned} \tag{7.6}$$

This analysis shows that any traffic grooming scheme for the MIN-ADM-MC-TG-UR problem is an approximation algorithm with approximation ratio $2g$. An interesting observation is that in the case of no grooming ($g = 1$), any wavelength assignment will be within twice the optimal as far as the number of ADMs required in the unidirectional ring network is concerned.

### 7.2.3 Random Traffic Grooming

A very simple traffic grooming scheme for the MIN-ADM-MC-TG-UR problem described in Section 6.2 is the random traffic grooming strategy. Let the triple $\{\vec{C}, \mathcal{M}, g\}$ be the given instance of the MIN-ADM-MC-TG-UR problem. We randomly partition the set $\mathcal{M}$ of the given multicast traffic requests into subsets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{\lceil \frac{|\mathcal{M}|}{g} \rceil}$, each containing a maximum of $g$ requests. The grooming solution is to assign a single wavelength to all the multicast traffic requests in a particular partition. Multicast traffic requests in different partitions are assigned distinct wavelengths. This is clearly possible since we are providing a separate sub-wavelength channel for each traffic request. The complete scheme is presented as Algorithm 7 (RANDOM-TG).

**Subroutine 7** RANDOM-TG

**Require:** MIN-ADM-MC-TG-UR problem instance $\{\vec{C}, \mathcal{M}, g\}$.

**Ensure:** A traffic grooming solution $\{\lambda^{\text{RAND}}, \omega^{\text{RAND}}\} \in \Xi_{\{\vec{C},\mathcal{R},g\}}$, where $\mathcal{R}$ is the set of directed paths corresponding to the set $\mathcal{M}$ of multicast traffic requests on the unidirectional ring $\vec{C}$.

Partition $\mathcal{M}$ into subsets $\{\mathcal{S}_1, \mathcal{S}_2 \ldots, \mathcal{S}_{\lceil \frac{|\mathcal{M}|}{g} \rceil}\}$ such that $|\mathcal{S}_i| \leq g$ for every $i \in \{1, 2, \ldots, \left\lceil \frac{|\mathcal{M}|}{g} \right\rceil\}$.
**for all** sets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{\lceil \frac{|\mathcal{M}|}{g} \rceil}$ **do**
    $\lambda^{\text{RAND}}(\vec{R}) \leftarrow i$ for every $\vec{R}$ such that the corresponding multicast request is in $\mathcal{S}_i$.
**end for**
**for all** $\vec{R} \in \mathcal{R}$ **do**
    $\omega^{\text{RAND}}(\vec{R}) \leftarrow \min\{k \in \mathbb{N} : \nexists \vec{S} \in \mathcal{R}$ such that $\lambda^{\text{RAND}}(\vec{R}) = \lambda^{\text{RAND}}(\vec{S})$ and $\omega^{\text{RAND}}(\vec{S}) = k\}$
**end for**

Let us denote the traffic grooming solution generated by RANDOM-TG by the pair $\{\lambda^{\mathrm{RAND}}, \omega^{\mathrm{RAND}}\}$. According to our notations, the ring vertex $v \in V_{\vec{C}}$ acts as the source or as one of the destination nodes for the set $\mathcal{M}_v$ of multicast traffic requests. Hence, $v$ must be equipped with an ADM corresponding to all the wavelengths in the set $\lambda^{\mathrm{RAND}}(\mathcal{R}_v)$, where $\mathcal{R}_v$ is the set of directed paths corresponding to the multicast traffic requests in the set $\mathcal{M}_v$. Let $U_{\vec{C}} \subseteq V_{\vec{C}}$ be the set of ring vertices that act as the source node or as one of the destination nodes for at least one traffic request, i.e., $U_{\vec{C}} := \{v \in V_{\vec{C}} : \mathcal{M}_v \neq \emptyset\}$. For a ring vertex $v \in U_{\vec{C}}$, the worst that can happen is that we have to equip $v$ with ADMs corresponding to all the wavelengths used by the traffic grooming solution. On the other hand, since any ring vertex $v \in V_{\vec{C}} \setminus U_{\vec{C}}$ does not act as the source or as one of the destinations for any multicast traffic request, no traffic is being added or dropped at $v$ and therefore, there is no need to equip $v$ with ADM corresponding to any wavelength. Using these two arguments, we can upper bound the number of ADMs required by the traffic grooming solution generated by RANDOM-TG as

$$\sum_{v \in V_{\vec{C}}} |\lambda^{\mathrm{RAND}}(\mathcal{R}_v)| \leq \sum_{v \in U_{\vec{C}}} |\lambda^{\mathrm{RAND}}(\mathcal{R})| = \sum_{v \in U_{\vec{C}}} \left\lceil \frac{|\mathcal{M}|}{g} \right\rceil = |U_{\vec{C}}| \left\lceil \frac{|\mathcal{M}|}{g} \right\rceil. \tag{7.7}$$

We define the *size* of a multicast traffic request $\{s, D\}$ as $|\{s\} \cup D| = 1 + |D|$. Let us define $z_{\mathrm{avg}}$ to be the average size of the multicast traffic requests in the set $\mathcal{M}$, i.e., $z_{\mathrm{avg}} := \frac{1}{|\mathcal{M}|} \sum_{v \in V_{\vec{C}}} |\mathcal{M}_v|$. Equations (7.5) and (7.7) give us

$$\begin{aligned} \sum_{v \in V_{\vec{C}}} |\lambda^{\mathrm{RAND}}(\mathcal{R}_v)| &\leq |U_{\vec{C}}| \left\lceil \frac{\sum_{v \in V_{\vec{C}}} |\mathcal{M}_v|}{g z_{\mathrm{avg}}} \right\rceil \leq |U_{\vec{C}}| \left\lceil \frac{2 \sum_{v \in V_{\vec{C}}} \chi_{G_{\mathcal{R}_v}}}{g z_{\mathrm{avg}}} \right\rceil \\ &\leq |U_{\vec{C}}| \sum_{v \in V_{\vec{C}}} \left\lceil \frac{\chi_{G_{\mathcal{R}_v}}}{g} \right\rceil \leq |V_{\vec{C}}| \Big( \min_{\{\lambda, \omega\} \in \Xi_{\{\vec{C}, \mathcal{R}, g\}}} \sum_{v \in V_{\vec{C}}} |\lambda(\mathcal{M}_v)| \Big). \end{aligned} \tag{7.8}$$

The third inequality holds because of the fact that $z_{avg} \geq 2$. This is true since every multicast traffic request has one source node and at least one destination node.

If we further assume a large enough average multicast traffic request size, we can achieve a better bound. In particular, using equations (7.5) and (7.7),

$$
\begin{aligned}
\sum_{v \in V_{\vec{C}}} |\lambda^{\mathrm{RAND}}(\mathcal{R}_v)| \leq |U_{\vec{C}}| \left\lceil \frac{2\sum_{v\in V_{\vec{C}}} \chi_{G_{\mathcal{R}_v}}}{g z_{\mathrm{avg}}} \right\rceil &\leq \frac{|U_{\vec{C}}|}{z_{\mathrm{avg}}} \left\lceil \frac{2\sum_{v\in V_{\vec{C}}} \chi_{G_{\mathcal{R}_v}}}{g} + z_{\mathrm{avg}} \right\rceil \\
&\leq \frac{2|U_{\vec{C}}|}{z_{\mathrm{avg}}} \left\lceil \frac{\sum_{v\in V_{\vec{C}}} \chi_{G_{\mathcal{R}_v}}}{g} \right\rceil + |U_{\vec{C}}| \leq \frac{2|U_{\vec{C}}|}{z_{\mathrm{avg}}} \sum_{v\in V_{\vec{C}}} \left\lceil \frac{\chi_{G_{\mathcal{R}_v}}}{g} \right\rceil + |U_{\vec{C}}| \\
&\leq \left( \frac{2|U_{\vec{C}}|}{z_{\mathrm{avg}}} + 1 \right) \sum_{v\in V_{\vec{C}}} \left\lceil \frac{\chi_{G_{\mathcal{R}_v}}}{g} \right\rceil \\
&\leq \left( \frac{2|V_{\vec{C}}|}{z_{\mathrm{avg}}} + 1 \right) \Big( \min_{\{\lambda,\omega\}\in \Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v\in V_{\vec{C}}} |\lambda(\mathcal{M}_v)| \Big). \qquad (7.9)
\end{aligned}
$$

The second last inequality is due to the fact that if a ring vertex $v$ acts as the source or as one of the destinations for at least one multicast traffic request, then the conflict graph $G_{\mathcal{R}_v}$ has at least one vertex and therefore $\chi_{G_{\mathcal{R}_v}} \geq 1$. Observing that there are $|U_{\vec{C}}|$ such nodes, we get $\sum_{v\in V_{\vec{C}}} \left\lceil \frac{\chi_{G_{\mathcal{R}_v}}}{g} \right\rceil \geq |U_{\vec{C}}|$.

From the analysis presented above, we see that RANDOM-TG, the simple strategy of grooming any $g$ traffic requests on the same wavelength, is an approximation algorithm with approximation ratio $|V_{\vec{C}}|$. Moreover, if the average size of the given multicast traffic requests satisfies the inequality $z_{\mathrm{avg}} \geq \frac{2|V_{\vec{C}}|}{|V_{\vec{C}}|-1}$, then we can prove a better approximation ratio of $1 + \frac{2|V_{\vec{C}}|}{z_{\mathrm{avg}}}$ for RANDOM-TG.

### 7.2.4 Arc Coloring Based Traffic Grooming

Another simple traffic grooming scheme for the MIN-ADM-MC-TG-UR problem described in Section 6.2, is to employ the ARC-COL-BASED-TG algorithm presented in Section 7.1.3. Let the triple $\{\vec{C}, \mathcal{M}, g\}$ be the given instance of the MIN-ADM-MC-TG-UR problem. Let $\mathcal{R}$ be the set of directed paths corresponding to the set of multicast traffic requests $\mathcal{M}$ on the unidirectional ring $\vec{C}$. We simply use the traffic grooming solution generated by ARC-COL-BASED-TG for the MIN-WAVE-MC-TG-UR problem instance $\{\vec{C}, \mathcal{R}, g\}$. For ease of exposition, we refer to this modification of ARC-COL-BASED-TG by the same name.

For the subsequent discussion, we assume that we employ the $\frac{3}{2}$-approximation algorithm for coloring circular arc graphs, developed by Karapetian [69], as ARC-COL-ALGO in ARC-COL-BASED-TG. According to the notation defined previously, $U_{\vec{C}}$ is the set of ring vertices that act as the source or as one of the destinations for at least one multicast traffic request in the set $\mathcal{R}$. Following similar line of reasoning as we did for the analysis of RANDOM-TG, we can argue that the total number of ADMs required by the traffic grooming solution $\{\lambda^{\text{ARC}}, \omega^{\text{ARC}}\}$ generated by ARC-COL-BASED-TG for the given MIN-ADM-MC-TG-UR problem instance $\{\vec{C}, \mathcal{M}, g\}$ can be upper bounded as

$$\sum_{v \in V_{\vec{C}}} |\lambda^{\text{ARC}}(\mathcal{R}_v)| \leq \sum_{v \in U_{\vec{C}}} |\lambda^{\text{ARC}}(\mathcal{R})| \leq \sum_{v \in U_{\vec{C}}} \left\lceil \frac{|\psi^{\text{ARC}}(\mathcal{R})|}{g} \right\rceil \leq |U_{\vec{C}}| \left\lceil \frac{\left\lfloor \frac{3}{2} \chi_{G_{\mathcal{R}}} \right\rfloor}{g} \right\rceil. \quad (7.10)$$

The final inequality is due to the fact that ARC-COL-ALGO is assumed to be a $\frac{3}{2}$-approximation algorithm for the problem of coloring circular arc graphs.

Let us define $z_{\min}$ to be the size of the smallest multicast traffic request in the set $\mathcal{M}$, i.e., $z_{\min} := \min_{\{s,D\} \in \mathcal{M}} 1 + |D|$. The minimum number of ADMs required for each wavelength is $z_{\min}$. Hence, using equation (7.1), a lower bound (other than our primary lower bound given in equation (7.3)) on the total number of ADMs required by any traffic grooming solution for the given instance of the MIN-ADM-MC-TG-UR problem is

$$\begin{aligned}
\min_{\{\lambda,\omega\} \in \Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v \in V_{\vec{C}}} |\lambda(\mathcal{M}_v)| &= \min_{\{\lambda,\omega\} \in \Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{k \in \lambda(\mathcal{R})} \Big| \bigcup_{\{s,D\} \in \mathcal{M}_{\lambda=k}} \{s\} \cup D \Big| \\
&\geq \min_{\{\lambda,\omega\} \in \Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{k \in \lambda(\mathcal{R})} z_{\min} = z_{\min} \left( \min_{\{\lambda,\omega\} \in \Xi_{\{\vec{C},\mathcal{R},g\}}} |\lambda(\mathcal{R})| \right) \\
&\geq z_{\min} \left\lceil \frac{\chi_{G_{\mathcal{R}}}}{g} \right\rceil, \qquad (7.11)
\end{aligned}$$

where $\mathcal{M}_{\lambda=k}$ is defined to be the set of multicast traffic requests that are assigned wavelength $k$ according to the traffic grooming solution $\{\lambda, \omega\}$, i.e., $\mathcal{M}_{\lambda=k} := \{\{s, D\} \in \mathcal{M} : \lambda(\vec{R}_{\{s,D\}}) = k\}$. The first equality is simply another way of counting the number of ADMs required by any traffic grooming solution. Instead of counting the number of ADMs required on each ring vertex and adding these, we are counting the number of ring vertices on which an ADM corresponding to a particular wavelength is required, and then we sum over all the wavelengths required by the traffic grooming solution. The first inequality is simply by the definition of $z_{\min}$ and the final inequality is by equation (7.1).

Using equations (7.10) and (7.11), we see that

$$\sum_{v \in V_{\vec{C}}} |\lambda^{\mathrm{ARC}}(\mathcal{R}_v)| \leq |U_{\vec{C}}| \left\lceil \frac{\lfloor \frac{3}{2}\chi_{G_{\mathcal{R}}} \rfloor}{g} \right\rceil \leq |U_{\vec{C}}| \left\lceil \frac{2\chi_{G_{\mathcal{R}}}}{g} \right\rceil \leq 2|U_{\vec{C}}| \left\lceil \frac{\chi_{G_{\mathcal{R}}}}{g} \right\rceil$$
$$\leq \frac{2|V_{\vec{C}}|}{z_{\min}} \Big( \min_{\{\lambda,\omega\} \in \Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v \in V_{\vec{C}}} |\lambda(\mathcal{M}_v)| \Big). \tag{7.12}$$

Hence, ARC-COL-BASED-TG is a $\frac{2|V_{\vec{C}}|}{z_{\min}}$-approximation algorithm for the MIN-ADM-MC-TG-UR problem.

We can arrive at a different (better in some cases) approximation ratio by following a separate line of analysis. For the conflict graph $G_{\mathcal{R}_v}$, let

$$\frac{\chi_{G_{\mathcal{R}_v}}}{g} = 2n + \delta + \epsilon, \tag{7.13}$$

where $n$ is a non-negative integer, $\delta \in \{0, 1\}$ and $0 \leq \epsilon < 1$. From equations (7.11) and (7.13), we get

$$\min_{\{\lambda,\omega\} \in \Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v \in V_{\vec{C}}} |\lambda(\mathcal{M}_v)| \geq z_{\min} \lceil 2n + \delta + \epsilon \rceil = z_{\min}(2n + \delta + \lceil \epsilon \rceil). \tag{7.14}$$

Again from equations (7.10) and (7.13), we get

$$\sum_{v \in V_{\vec{C}}} |\lambda^{\mathrm{ARC}}(\mathcal{R}_v)| \leq |U_{\vec{C}}| \left\lceil \frac{\lfloor \frac{3}{2}\chi_{G_{\mathcal{R}}} \rfloor}{g} \right\rceil \leq |U_{\vec{C}}| \left\lceil \frac{3\chi_{G_{\mathcal{R}}}}{2g} \right\rceil$$
$$= |U_{\vec{C}}| \left\lceil \frac{3}{2}(2n + \delta + \epsilon) \right\rceil \leq |U_{\vec{C}}| \left( 3n + 3 \left\lceil \frac{\delta + \epsilon}{2} \right\rceil \right). \tag{7.15}$$

The following are the only two cases possible:

(i) $\delta + \epsilon = 0 \Rightarrow \delta = \epsilon = 0$

In this case, from equation (7.14), the lower bound becomes

$$\min_{\{\lambda,\omega\}\in\Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v\in V_{\vec{C}}} |\lambda(\mathcal{M}_v)| \geq 2nz_{\min}. \tag{7.16}$$

From equations (7.15) and (7.16), we get

$$\sum_{v\in V_{\vec{C}}} |\lambda^{\mathrm{ARC}}(\mathcal{R}_v)| \leq 3n|U_{\vec{C}}| \leq \frac{3|V_{\vec{C}}|}{2z_{\min}}\Big(\min_{\{\lambda,\omega\}\in\Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v\in V_{\vec{C}}} |\lambda(\mathcal{M}_v)|\Big). \tag{7.17}$$

(ii) $\delta + \epsilon > 0$

In this case, from equation (7.14) we get

$$\min_{\{\lambda,\omega\}\in\Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v\in V_{\vec{C}}} |\lambda(\mathcal{M}_v)| \geq z_{\min}(2n + \delta + \lceil\epsilon\rceil) \geq z_{\min}(2n+1). \tag{7.18}$$

From equation (7.15), we get

$$\sum_{v\in V_{\vec{C}}} |\lambda^{\mathrm{ARC}}(\mathcal{R}_v)| \leq |U_{\vec{C}}| \left(3n + 3\left\lceil\frac{\delta+\epsilon}{2}\right\rceil\right) = |U_{\vec{C}}|(3n+3), \tag{7.19}$$

where the equality is based on the fact that since $\delta \in \{1,0\}, \epsilon \in [0,1)$ and $\delta+\epsilon > 0$, we have $0 < \delta+\epsilon < 2$. Observe that at least one ADM is required at all the ring vertices in the set that act as the source or as one of the destinations for at least one multicast traffic request in the set $\mathcal{M}$. Therefore, we have

$$\min_{\{\lambda,\omega\}\in\Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v\in V_{\vec{C}}} |\lambda(\mathcal{M}_v)| \geq |U_{\vec{C}}|. \tag{7.20}$$

Using equations (7.18), (7.19) and (7.20), we get

$$\begin{aligned}\sum_{v\in V_{\vec{C}}} |\lambda^{\mathrm{ARC}}(\mathcal{R}_v)| &\leq \frac{3|U_{\vec{C}}|}{2}(2n+1) + \frac{3|U_{\vec{C}}|}{2} \\ &\leq \frac{3}{2}\Big(\frac{|V_{\vec{C}}|}{z_{\min}} + 1\Big)\Big(\min_{\{\lambda,\omega\}\in\Xi_{\{\vec{C},\mathcal{R},g\}}} \sum_{v\in V_{\vec{C}}} |\lambda(\mathcal{M}_v)|\Big). \end{aligned} \tag{7.21}$$

From equations (7.17) and (7.21), we observe that ARC-COL-BASED-TG has an approximation ratio $\frac{3(|V_{\vec{C}}|+z_{\min})}{2z_{\min}}$ for the MIN-ADM-MC-TG-UR problem. Also note that whenever $z_{\min} < \frac{|V_{\vec{C}}|}{3}$, this approximation ratio is better than the previously computed approximation ratio of $\frac{2|V_{\vec{C}}|}{z_{\min}}$.

## 7.3 Minimizing ADMs: A Heuristic

In this section, we present a heuristic traffic grooming approach for the MIN-ADM-MC-TG-UR problem described in Section 6.2. We shall not prove any results for the worst case performance of this scheme, but we shall study the average performance by simulations.

The complete scheme is presented as Algorithm 8 (ITER-IMPROVE-TG). As the name suggests, the scheme is based on the idea of iteratively improving a traffic grooming solution. Let the triple $\{\vec{C}, \mathcal{M}, g\}$ be the given instance of MIN-ADM-MC-TG-UR problem. Let $\mathcal{R}$ be the set of directed paths corresponding to the set $\mathcal{M}$ of multicast traffic requests on the unidirectional ring $\vec{C}$. We start with an initial wavelength assignment $\lambda_0$ which assigns different wavelengths to each of the directed paths in the set $\mathcal{R}$. The ITER-IMPROVE-TG algorithm proceeds iteratively and in the $n$-th iteration, we generate the wavelength assignment $\lambda_n$ which is an improvement over the previous wavelength assignment $\lambda_{n-1}$, i.e., $|\lambda_n(\mathcal{R})| < |\lambda_{n-1}(\mathcal{R})|$. At the $n$-th iteration of ITER-IMPROVE-TG, we define a pair of wavelengths $i, j \in \lambda_{n-1}(\mathcal{R})$ to be *reducible* if all the directed paths that have been assigned either of the two wavelengths $i$ or $j$, can actually be assigned a single wavelength. Let us define $\mathcal{R}_{\lambda_{n-1}=i}$ to be the set of all the directed paths that have been assigned wavelength $i$ by the wavelength assignment $\lambda_{n-1}$, i.e., $\mathcal{R}_{\lambda_{n-1}=i} := \{\vec{R} \in \mathcal{R} : \lambda_{n-1}(\vec{R}) = i\}$. Observe that the pair of wavelengths $i, j \in \lambda_{n-1}(\mathcal{R})$ is reducible if and only if the chromatic number of the conflict graph corresponding to the set $\mathcal{R}_{\lambda_{n-1}=i} \bigcup \mathcal{R}_{\lambda_{n-1}=j}$ of directed paths is at most $g$, i.e., if and only if $\chi_{G_{\mathcal{R}_{\lambda_{n-1}=i} \cup \mathcal{R}_{\lambda_{n-1}=j}}} \leq g$.

Let $\mathcal{M}_{\lambda_{n-1}=i}$ be the set of multicast traffic requests corresponding to the set $\mathcal{R}_{\lambda_{n-1}=i}$ of directed paths that have been assigned wavelength $i$ by the wavelength assignment $\lambda_{n-1}$, i.e., $\mathcal{M}_{\lambda_{n-1}=i} := \{\{s, D\} \in \mathcal{M} : \lambda_{n-1}(\vec{R}_{\{s,D\}}) = i\}$. We define $V_{\vec{C}}^{\lambda_{n-1}=i}$ to be the set of ring vertices that act as either the source node or as one of the destination nodes for at least one multicast traffic request whose corresponding directed path has been assigned wavelength $i$ by the wavelength assignment $\lambda_{n-1}$, i.e., $V_{\vec{C}}^{\lambda_{n-1}=i} := \{v \in V_{\vec{C}} : v \in \bigcup_{\{s,D\} \in \mathcal{M}_{\lambda_{n-1}=i}} \{s\} \cup D\}$. Observe that, according to the wavelength assignment $\lambda_{n-1}$, $V_{\vec{C}}^{\lambda_{n-1}=i}$ is the set of ring vertices that must be equipped with ADMs corresponding to the wavelength $i$. Moreover, for any reducible wavelength pair $i, j \in \lambda_{n-1}(\mathcal{R})$, if we indeed use a single wavelength for all the directed paths in the set $\mathcal{R}_{\lambda_{n-1}=i} \bigcup \mathcal{R}_{\lambda_{n-1}=j}$, we would end up saving ADMs precisely on those ring vertices which require ADMs corresponding to both wavelengths $i$ and $j$ according to the current wavelength assignment, i.e., the set of vertices given as $V_{\vec{C}}^{\lambda_{n-1}=i} \bigcap V_{\vec{C}}^{\lambda_{n-1}=j}$. Hence, the number of ADMs thus saved is equal to $|V_{\vec{C}}^{\lambda_{n-1}=i} \bigcap V_{\vec{C}}^{\lambda_{n-1}=j}|$.

During the $n$-th iteration of ITER-IMPROVE-TG, in order to generate the wavelength assignment $\lambda_n$ from the wavelength assignment $\lambda_{n-1}$, we find the reducible wavelength pair $a, b \in \lambda_{n-1}(\mathcal{R})$ such that for every reducible wavelength pair $i, j \in \lambda_{n-1}(\mathcal{R})$, $|V_{\vec{C}}^{\lambda_{n-1}=a} \bigcap V_{\vec{C}}^{\lambda_{n-1}b}| \geq |V_{\vec{C}}^{\lambda_{n-1}=i} \bigcap V_{\vec{C}}^{\lambda_{n-1}=j}|$. If there is more than one such wavelength pair, among all the wavelength pairs satisfying the constraint, we select the wavelength pair $a, b \in \lambda_{n-1}(\mathcal{R})$ having the minimum value of $|V_{\vec{C}}^{\lambda_{n-1}=a} \bigcup V_{\vec{C}}^{\lambda_{n-1}b}|$. This is motivated by the fact that if $|V_{\vec{C}}^{\lambda_{n-1}=i}|$ is large for wavelength $i$, then there is a high chance that at some later iteration we have a reducible wavelength pair containing $i$, corresponding to larger ADM savings; therefore we may not want to use the wavelength $i$ in the current step for smaller ADM savings. Any remaining ties are broken uniformly randomly. After selecting the reducible wavelength pair $a, b \in \lambda_{n-1}(\mathcal{R})$ as described above, we generate the wavelength assignment $\lambda_n$ by assigning $\lambda_n(\vec{R}) = \lambda_{n-1}(\vec{R})$ for every directed path $\vec{R} \in \mathcal{R} \setminus \mathcal{R}_{\lambda_{n-1}=b}$ and $\lambda_n(\vec{R}) = a$ for every directed path $\vec{R} \in \mathcal{R}_{\lambda_{n-1}=b}$.

We continue until there are no reducible wavelength pairs left. Observe that initially the number of wavelengths is equal to the number of multicast traffic requests and each iteration reduces the number of wavelengths by one, therefore

**Subroutine 8** ITER-IMPROVE-TG

**Require:** MIN-ADM-MC-TG-UR problem instance $\{\vec{C}, \mathcal{M}, g\}$.

**Ensure:** A traffic grooming solution $\{\lambda^{\text{ITER}}, \omega^{\text{ITER}}\} \in \Xi_{\{\vec{C},\mathcal{R},g\}}$, where $\mathcal{R}$ is the set of directed paths corresponding to the set $\mathcal{M}$ of multicast traffic requests on the unidirectional ring $\vec{C}$.

**for all** $\vec{R} \in \mathcal{R}$ **do**
    $\lambda_0(\vec{R}) \leftarrow \min\{k \in \mathbb{N} : \nexists \vec{S} \in \mathrm{R} \text{ such that } \lambda_0(\vec{S}) = k\}$
**end for**
$n \leftarrow 1$
**while** $\exists$ some reducible wavelength pair in the set $\lambda_{n-1}(\mathrm{R})$ **do**
    Determine the reducible wavelength pair $a, b \in \lambda_{n-1}(\mathcal{R})$ having the largest value of $|V_{\vec{C}}^{\lambda_{n-1}=a} \cap V_{\vec{C}}^{\lambda_{n-1}=b}|$. If there are several such pairs, select the one with the smallest value of $|V_{\vec{C}}^{\lambda_{n-1}=a} \cup V_{\vec{C}}^{\lambda_{n-1}=b}|$. If there are still ties, then randomly pick any of the possible choices.
    **for all** $\vec{R} \in \mathcal{R}$ **do**
        **if** $\vec{R} \in \mathcal{R}_{\lambda_{n-1}=b}$ **then**
            $\lambda_n(\vec{R}) \leftarrow a$
        **else**
            $\lambda_n(\vec{R}) \leftarrow \lambda_{n-1}(\vec{R})$
        **end if**
    **end for**
    $n \leftarrow n + 1$
**end while**
$\lambda^{\text{ITER}} := \lambda_{n-1}$.
**for all** $\vec{R} \in \mathcal{R}$ **do**
    $\omega^{\text{ITER}}(\vec{R}) \leftarrow \min\{k \in \mathbb{N} : \nexists \vec{S} \in \mathcal{R} \text{ such that } \lambda^{\text{ITER}}(\vec{R}) = \lambda^{\text{ITER}}(\vec{S}) \text{ and } \omega^{\text{ITER}}(\vec{S}) = k\}$
**end for**

$|\lambda_n(\mathcal{R})| = |\mathcal{M}| - n$. This shows that the maximum number of iterations is upper bounded by the number of multicast traffic requests.

During the $n$-th iteration of ITER-IMPROVE-TG, determining whether or not any wavelength pair $i, j \in \lambda_{n-1}(\mathcal{R})$ is reducible or not, is NP Complete. This is because, as explained before, it involves solving the problem of coloring the conflict graph $G_{\mathcal{R}_{\lambda_{n-1}=i} \bigcup \mathcal{R}_{\lambda_{n-1}=j}}$, which in general belongs to the family of circular arc graphs. Instead of doing this, in ITER-IMPROVE-TG, we color the conflict graph $G_{\mathcal{R}_{\lambda_{n-1}=i} \bigcup \mathcal{R}_{\lambda_{n-1}=j}}$ using Tucker's algorithm for coloring circular arcs [70] and see if we need more than $g$ colors. Clearly this is sub-optimal because we may not be able to find all the reducible wavelength pairs, but we still use this because in general Tucker's algorithm gives a good bound on the chromatic number [71].

## 7.4 Complexity Analysis

In this section, we present the complexity analysis for RANDOM-TG, ARC-COL-BASED-TG and ITER-IMPROVE-TG, the three traffic grooming schemes presented in this chapter for the MIN-ADM-MC-TG-UR problem.

### 7.4.1 Random Traffic Grooming

First, we consider RANDOM-TG traffic grooming scheme described in Section 7.2.3. In this scheme, we randomly partition $\mathcal{M}$, the set of the given multicast traffic requests into $\left\lceil \frac{|\mathcal{M}|}{g} \right\rceil$ subsets denoted by $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{\lceil \frac{|\mathcal{M}|}{g} \rceil}$, each having cardinality at most $g$. This partitioning requires $O(|\mathcal{M}|)$ steps. The set of wavelengths employed by the traffic grooming solution generated by RANDOM-TG is $\lambda^{\text{RAND}}(\mathcal{R})$, where $\mathcal{R}$ is the set of directed paths corresponding to the set $\mathcal{M}$ of multicast traffic requests on the unidirectional ring $\vec{C}$. The generated wavelength assignment $\lambda^{\text{RAND}}$ assigns wavelength $i$ to the directed paths corresponding to all the multicast traffic request in subset $\mathcal{S}_i$. Clearly the number of wavelengths required by the mapping $\lambda^{\text{RAND}}$ is equal to $\left\lceil \frac{|\mathcal{M}|}{g} \right\rceil$. Moreover, corresponding to any wavelength $i \in \lambda^{\text{RAND}}(\mathcal{R})$, we require ADMs at all the ring vertices in the set $\bigcup_{\{s,D\}\in\mathcal{S}_i}\{s\} \cup D$. Since $\{s\} \cup D \subseteq V_{\vec{C}}$ for every multicast traffic request $\{s, D\} \in \mathcal{M}$, the number of steps required for evaluating $\bigcup_{\{s,D\}\in\mathcal{S}_i}\{s\} \cup D$ is equal to $|V_{\vec{C}}||\mathcal{S}_i|$. Hence, the total number of steps required for determining the placement of ADMs on all the ring vertices, corresponding to all the wavelengths being employed by the traffic grooming solution, is

$$\sum_{i\in\lambda^{\text{RAND}}(\mathcal{R})} |V_{\vec{C}}||\mathcal{S}_i| = |V_{\vec{C}}| \sum_{i\in\lambda^{\text{RAND}}(\mathcal{R})} |\mathcal{S}_i| = |V_{\vec{C}}||\mathcal{M}|. \tag{7.22}$$

Therefore the overall complexity of RANDOM-TG is $O\big(|V_{\vec{C}}||\mathcal{M}|\big)$.

### 7.4.2 Arc Coloring Based Traffic Grooming

Next, we consider ARC-COL-BASED-TG described in Section 7.2.4. In this scheme, we have to color the conflict graph $G_{\mathcal{R}}$ of the set $\mathcal{R}$ of directed paths corresponding to the set $\mathcal{M}$ of multicast traffic requests. As described previously, Karapetian's circular arc graph coloring algorithm [69] is used for this purpose. Hence, generating this coloring requires $O\big(|\mathcal{M}|^2\big)$ time. We denote the generated coloring by $\psi^{\text{ARC}}$. The set of directed paths $\mathcal{R}$ is partitioned into $\left\lceil \frac{|\psi^{\text{ARC}}(\mathcal{R})|}{g} \right\rceil$ subsets denoted by $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{\left\lceil \frac{|\psi^{\text{ARC}}(\mathcal{R})|}{g} \right\rceil}$, such that each subset requires at most $g$ colors and directed paths in different subsets are assigned different colors by the coloring $\psi^{\text{ARC}}$. This partitioning requires $O\big(|\mathcal{M}|\big)$ steps. For any $i \in \{1, \ldots, \left\lceil \frac{|\psi^{\text{ARC}}(\mathcal{R})|}{g} \right\rceil\}$, we define $\mathcal{S}_i$ to be the set of multicast traffic requests corresponding to the directed paths in the set $\mathcal{P}_i$. The generated wavelength assignment $\lambda^{\text{ARC}}$ assigns wavelength $i$ to all the directed paths in the set $\mathcal{P}_i$, therefore corresponding to wavelength $i$, we require ADMs at all the ring vertices in the set $\bigcup_{\{s,D\}\in\mathcal{S}_i}\{s\} \cup D$. Following the arguments presented in Section 7.4.1, we argue that the total number of steps required for determining the placement of ADMs on all the ring vertices, corresponding to all the wavelengths being employed by the traffic grooming solution, is equal to $|V_{\vec{C}}||\mathcal{M}|$. Therefore the overall complexity of ARC-COL-BASED-TG is $O\big(|V_{\vec{C}}||\mathcal{M}| + |\mathcal{M}|^2\big)$.

### 7.4.3 Iterative Improvement Based Traffic Grooming

Finally we consider ITER-IMPROVE-TG described in Section 7.3. In this scheme, we start off with the simple wavelength assignment $\lambda_0$, which assigns a different wavelength to every directed path in the set $\mathcal{R}$ corresponding to the set $\mathcal{M}$ of multicast traffic requests. In each iteration of ITER-IMPROVE-TG, we update the wavelength assignment by first determining the 'best' (as described in Section 7.3) reducible wavelength pair and then assigning a single wavelength to all the directed paths that were previously assigned either of the two wavelengths of the pair. We continue to update the wavelength assignment iteratively till there are no reducible wavelength pairs left.

Before the start of the $n$-th step of ITER-IMPROVE-TG, we assume that we have the following:

(i) Corresponding to every wavelength $i \in \lambda_{n-1}(\mathcal{R})$, the set $V_{\vec{C}}^{\lambda_{n-1}=i}$ of ring vertices that require ADMs corresponding to the wavelength $i$ according to the wavelength assignment $\lambda_{n-1}$.

(ii) The identity of every reducible wavelength pair in the set $\lambda_{n-1}(\mathcal{R})$.

(iii) Corresponding to every reducible wavelength pair $i, j \in \lambda_{n-1}(\mathcal{R})$, the values of $|V_{\vec{C}}^{\lambda_{n-1}=i} \bigcap V_{\vec{C}}^{\lambda_{n-1}=j}|$ and $|V_{\vec{C}}^{\lambda_{n-1}=i} \bigcup V_{\vec{C}}^{\lambda_{n-1}=j}|$. Note that these two values are required for deciding which reducible wavelength pair should be selected in the $n$-th iteration of ITER-IMPROVE-TG.

We shall not discuss any schemes for maintaining this data, but just point out that it can be done in a graphical manner.

First, let us determine the complexity of the $n$-th iteration of ITER-IMPROVE-TG. Since in each iteration we reduce the number of wavelengths by 1, $|\lambda_n(\mathcal{R})| = |\lambda_0(\mathcal{R})| - n = |\mathcal{R}| - n = |\mathcal{M}| - n$. Therefore, the number of wavelength pairs to consider in the $n$-th iteration of ITER-IMPROVE-TG is $\frac{(|\mathcal{M}|-n+1)(|\mathcal{M}|-n)}{2}$. Since we know the identities of the reducible wavelength pairs in the set $\lambda_{n-1}(\mathcal{R})$, and we have the values of $|V_{\vec{C}}^{\lambda_{n-1}=i} \bigcap V_{\vec{C}}^{\lambda_{n-1}=j}|$ and $|V_{\vec{C}}^{\lambda_{n-1}=i} \bigcup V_{\vec{C}}^{\lambda_{n-1}=j}|$ for every reducible wavelength pair $i, j \in \lambda_{n-1}(\mathcal{R})$, the number of steps required to determine the best reducible wavelength pair is linear in the number of wavelength pairs. After determining the best reducible wavelength pair $a, b \in \lambda_{n-1}(\mathcal{R})$, we generate a new wavelength assignment $\lambda_n$ from $\lambda_{n-1}$ by assigning the wavelength $a$ to all the directed paths in the set $\mathcal{R}_{\lambda_{n-1}=b}$. In order to update the data that we maintain, we need to determine the following:

(i) The set $V_{\vec{C}}^{\lambda_n=a}$.

(ii) For every wavelength $i \in \lambda_n(\mathcal{R}) \setminus \{a\}$, whether the pair $a, i$ is reducible or not.

(iii) Corresponding to every reducible pair $a, i \in \lambda_n(\mathcal{R})$, the values $|V_{\vec{C}}^{\lambda_n=a} \bigcap V_{\vec{C}}^{\lambda_n=i}|$ and $|V_{\vec{C}}^{\lambda_n=a} \bigcup V_{\vec{C}}^{\lambda_n=i}|$.

We do not need to determine the set $V_{\vec{C}}^{\lambda_n=i}$ for any wavelength $i \in \lambda_n(\mathcal{R}) \setminus \{a\}$, because for any such wavelength, $V_{\vec{C}}^{\lambda_n=i} = V_{\vec{C}}^{\lambda_{n-1}=i}$. Similarly, for any wavelength pair $i, j \in \lambda_n(\mathcal{R}) \setminus \{a\}$, whether the pair is reducible or not is not affected during the $n$-th iteration, therefore we do not need to determine this again. Next, observe that $V_{\vec{C}}^{\lambda_n=a} = V_{\vec{C}}^{\lambda_{n-1}=a} \bigcup V_{\vec{C}}^{\lambda_{n-1}=b}$. Since $V_{\vec{C}}^{\lambda_{n-1}=a}, V_{\vec{C}}^{\lambda_{n-1}=b} \subseteq V_{\vec{C}}$, determining $V_{\vec{C}}^{\lambda_n=a}$ requires $O\big(|V_{\vec{C}}|\big)$ steps. Therefore, for any wavelength $i \in \lambda_n(\mathcal{R})$, we can determine the sets $V_{\vec{C}}^{\lambda_n=i}$ in $O\big(|V_{\vec{C}}|\big)$ steps. Since, for any wavelength $i \in \lambda_n(\mathcal{R})$, $V_{\vec{C}}^{\lambda_n=i} \subseteq V_{\vec{C}}$, we can determine the values of $|V_{\vec{C}}^{\lambda_n=a} \bigcup V_{\vec{C}}^{\lambda_n=i}|$ and $|V_{\vec{C}}^{\lambda_n=a} \bigcap V_{\vec{C}}^{\lambda_n=i}|$ in

$O\big(|V_{\vec{C}}|\big)$ steps. Moreover in ITER-IMPROVE-TG, for checking whether the wavelength pair $a, i \in \lambda_n(\mathcal{R})$ is reducible or not, we check if we can color the conflict graph $G_{\mathcal{R}_{\lambda_n=a} \bigcup \mathcal{R}_{\lambda_n=i}}$ corresponding to the set of directed paths that are assigned either of the two wavelengths by the wavelength assignment $\lambda_n$, using at most $g$ colors or not. We employ Tucker's algorithm [70] for coloring the circular arc graph which requires $O\big(|\mathcal{R}_{\lambda_n=a} \bigcup \mathcal{R}_{\lambda_n=i}|^2\big)$ time. Since $|\mathcal{R}_{\lambda_n=a} \bigcup \mathcal{R}_{\lambda_n=i}| \leq |\mathcal{R}| = |\mathcal{M}|$, checking if wavelength pair $a, i \in \lambda_n(\mathcal{R})$ is reducible or not takes $O\big(|\mathcal{M}|^2\big)$ time. Therefore, the number of steps required in $n$-th iteration of ITER-IMPROVE-TG is $\big(|\mathcal{M}| - n - 1\big)O\big(|V_{\vec{C}}| + |\mathcal{M}|^2\big)$. As already explained in Section 7.3, the number of iterations in ITER-IMPROVE-TG is upper bounded by $|\mathcal{M}|$. Hence, the iterations in ITER-IMPROVE-TG require $O\Big(|\mathcal{M}|^2\big(|V_{\vec{C}}| + |\mathcal{M}|^2\big)\Big)$ steps.

Next, we count the number of steps required to initialize the data that we maintain at the start of ITER-IMPROVE-TG. Checking whether a wavelength pair $i, j \in \lambda_0(\mathcal{R})$ is reducible or not requires $O(1)$ steps. This is because for every wavelength $i \in \lambda_0(\mathcal{R})$, $|\mathcal{R}_{\lambda_0=i}| = 1$. The set $V_{\vec{C}}^{\lambda_0=i}$ corresponds to a single multicast traffic request and is therefore trivially determined. Determining the values of $|V_{\vec{C}}^{\lambda_0=i} \bigcap V_{\vec{C}}^{\lambda_0=j}|$ and $|V_{\vec{C}}^{\lambda_0=i} \bigcup V_{\vec{C}}^{\lambda_0=j}|$ for any wavelength pair $i, j \in \lambda_0(\mathcal{R})$ require $O\big(|V_{\vec{C}}|\big)$ steps. Since there are $\frac{|\mathcal{M}|(|\mathcal{M}|-1)}{2}$ wavelength pairs, the initialization requires $O\big(|V_{\vec{C}}||\mathcal{M}|^2\big)$ steps.

Therefore the overall complexity of ITER-IMPROVE-TG is $O\Big(|\mathcal{M}|^2\big(|V_{\vec{C}}| + |\mathcal{M}|^2\big)\Big)$.

## 7.5 Simulation Results

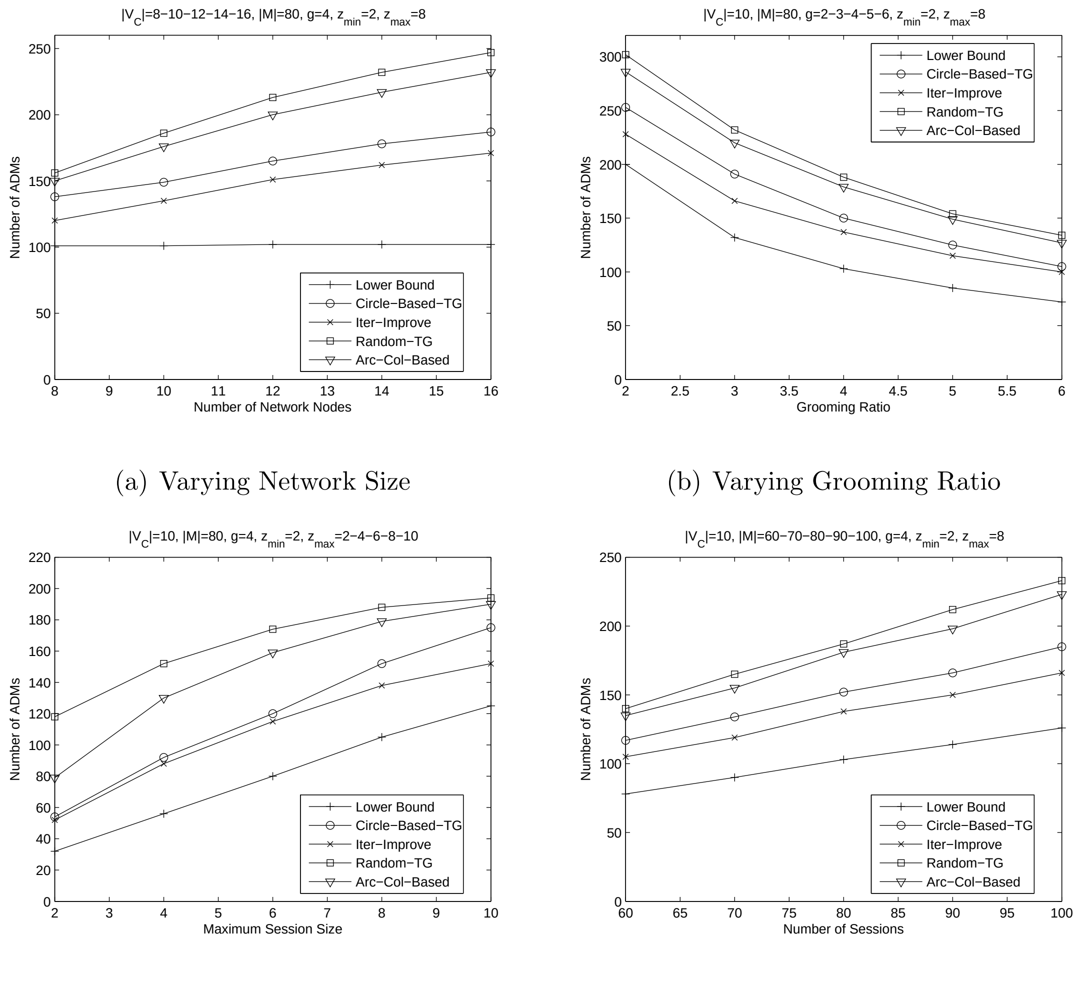
As we stated at the start of Section 7.3, we study the performance of ITER-IMPROVE-TG via simulations. The simulation results and the associated discussion is presented in this Section.

### 7.5.1 Circle Based Traffic Grooming

Since presently there is no other heuristic for grooming multicast traffic requests in unidirectional rings against with which we can compare the performance of ITER-IMPROVE-TG developed in Section 7.3, we extend the unicast traffic grooming scheme presented in [11] to the multicast case. In [11], the authors assume each of the given unicast traffic requests to be a *connection*. First, they combine pairs of connections with common end points to form complete circles. After constructing the maximum possible circles in this way, they apply 'Algorithm IV:Construct Circles - Non-Uniform Traffic' to construct the rest of the circles. Each circle corresponds to a sub-wavelength channel. After all the connections have been assigned to some circle, the circles are groomed into wavelengths. We extend their algorithm for multicast traffic by simply starting with the multicast traffic requests in place of the unicast traffic requests in the circle construction phase, i.e., we consider the multicast traffic requests to be the starting connections and construct the circles in exactly the same way. After we have the circles, the circle grooming heuristic is exactly the same as in [11]. We refer to this extended heuristic as CIRCLE-BASED-TG.

### 7.5.2 Results

We evaluate the performance of both ITER-IMPROVE-TG and CIRCLE-BASED-TG in terms of the number of ADMs required. For a more complete picture, we also compare the performance of both the heuristics to the lower bound on the number of ADMs required by any traffic grooming solution developed in equation (7.3). Since the number of wavelengths required also contributes to the network cost (albeit, not as much as the ADMs), we also compare the wavelengths required by the two heuristics. For the sake of completeness we also compare the performance of




(a) Varying Network Size

(b) Varying Grooming Ratio

(c) Varying Session Size

(d) Varying Number of Sessions

Figure 7.1: Number of ADMs required by ITER-IMPROVE-TG, CIRCLE-BASED-TG, RANDOM-TG, ARC-COL-BASED-TG and the lower bound in equation (7.3).

RANDOM-TG and ARC-COL-BASED-TG, the two simple multicast traffic grooming algorithms presented and analyzed in Section 7.2.

We parameterize the problem of grooming the given set $\mathcal{M}$ of multicast traffic requests on the given unidirectional ring $\vec{C}$ by the following five variables:

(i) $|V_{\vec{C}}|$: the number of vertices on the ring.

(ii) $|\mathcal{M}|$: the number of multicast traffic requests.

(iii) $g$: the grooming ratio.

(iv) $z_{\min}$: the size of the smallest multicast traffic request.

(v) $z_{\max}$: the size of the largest multicast traffic request.

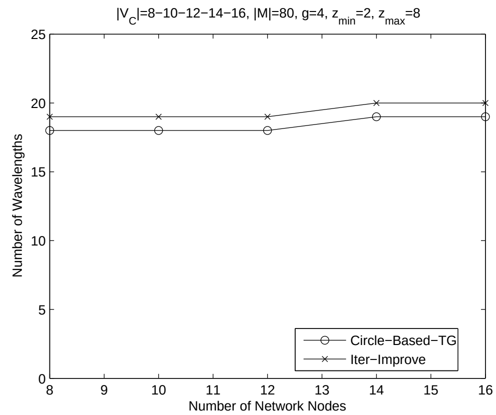
During the simulation, while generating a multicast traffic request, each ring vertex is given equal probability of being selected as the source node. The size of each multicast traffic request is selected uniformly randomly from $z_{\min}$ to $z_{\max}$. After the source node and the size $z$ of the multicast session are fixed, destination nodes are selected such that every subset of size $z-1$ of the remaining $|V_{\vec{C}}|-1$ ring vertices (since one ring vertex has already been selected as the source) has equal probability of being the set of destination nodes.

For simulation, we consider a nominal ring network having 10 vertices, 80 multicast traffic requests, with each session size selected uniformly randomly between 2 to 8 and having grooming ratio 4. We study the performance of both the schemes ITER-IMPROVE-TG and CIRCLE-BASED-TG by varying one parameter of the problem at a time in this nominal network. More specifically, we vary the grooming ratio from 2 to 6, the network size (number of vertices in the ring) from 8 to 16, the number of multicast traffic requests from 60 to 100 and the maximum size of the multicast traffic requests from 2 to 10. Figure 7.1 presents the simulation results comparing the total number of ADMs required by the various grooming schemes as well as the number of ADMs specified by the lower bound developed in equation (7.3). The simulation results comparing the total number of wavelengths required by ITER-IMPROVE-TG and CIRCLE-BASED-TG are presented in Figure 7.2. Each point in the plots is generated by taking an average of 20 randomly selected grooming problem instances with the required parameters.

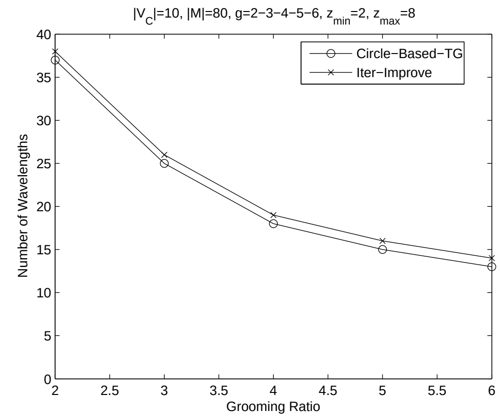
We can see from the plots that, as measured by the number of ADMs required, ITER-IMPROVE-TG always outperforms CIRCLE-BASED-TG. This is true even for unicast traffic (the case for which CIRCLE-BASED-TG was originally designed in [11]). We also note that ITER-IMPROVE-TG usually requires more wavelengths than CIRCLE-BASED-TG. But the increase in the number of wavelengths is never more than 2, and is overshadowed by the savings in the number of (more expensive) ADMs.

From the plots, we also observe that of the three traffic grooming schemes presented in this chapter, ITER-IMPROVE-TG always outperforms the simple grooming strategies RANDOM-TG and ARC-COL-BASED-TG. Among the two simple schemes, ARC-COL-BASED-TG always outperforms RANDOM-TG. We can justify this trend in the light of the complexity analysis of the three schemes presented in Section 7.4. Assuming that the number of multicast traffic requests to be groomed is much larger than the number of network nodes (which is usually the case and is true for our simulations as well), we observe that based on their time complexities, RANDOM-TG is the simplest, ITER-IMPROVE-TG is the most complex and ARC-COL-BASED-TG lies somewhere in-between the two. Since we get what we pay for, the relative performances of the three schemes is as expected. Although not presented in the plots, the number of wavelengths required by RANDOM-TG and ARC-COL-BASED-TG are also very similar to that required by ITER-IMPROVE-TG and CIRCLE-BASED-TG.

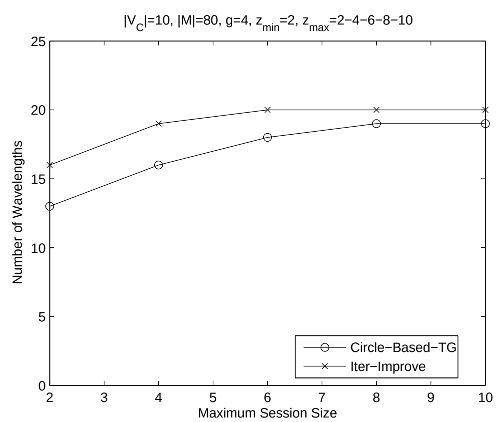
The plots also show that the lower bound on the minimum number of ADMs



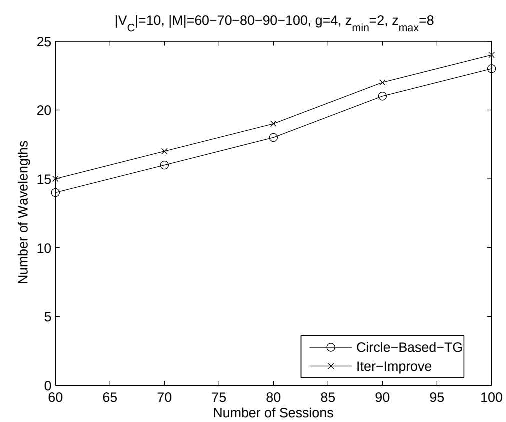

(a) Varying Network Size




(b) Varying Grooming Ratio




(c) Varying Session Size




(d) Varying Number of Sessions

Figure 7.2: Wavelengths required by ITER-IMPROVE-TG and CIRCLE-BASED-TG.

required by any traffic grooming solution for the MIN-ADM-MC-TG-UR problem developed in equation (7.3), tracks the performance curves of the simulated traffic grooming strategies as we vary the grooming ratio, the number of multicast traffic requests or the size of the multicast traffic requests. This suggests that the bound tracks the changes in these parameters quite well. But we observe that this is not so in the case of the size of ring. We discuss this anomaly next.

### 7.5.3 Discussion

During our simulations presented above, we observe an interesting property of the lower bound developed in equation (7.3) on the minimum number of ADMs required by any traffic grooming solution for the MIN-ADM-MC-TG-UR problem. It seems that the lower bound does not depend on the number of vertices in the ring. To explain this, we try to calculate the expected value of the lower bound on the number of ADMs required for grooming a set $\mathcal{M}$ of multicast traffic requests on a unidirectional ring $\vec{C}$, when the grooming ratio is assumed to be $g$. Let $z_{\{s,D\}}$ represent the size of any multicast traffic request $\{s, D\} \in \mathcal{M}$. For the purpose of our simulations (and hence for this analysis), we assume that the multicast session sizes $z_{\{s_1,D_1\}}, z_{\{s_2,D_2\}}, \ldots, z_{\{s_{|\mathcal{M}|},D_{|\mathcal{M}|}\}}$ are independent and identically distributed according to some cumulative distribution function $\mathcal{F}$ with mean $\mu_{\mathcal{F}}$. We also assume that the ring vertices (acting as the source node or as one of the destination nodes) in any multicast traffic request, are selected uniformly randomly from the set of all the vertices of the ring, i.e., for every multicast traffic request $\{s, D\} \in \mathcal{M}$ having size $z_{\{s,D\}}$, the probability that ring vertex $v \in \{s\} \cup D$ is equal to $\frac{z_{\{s,D\}}}{|V_{\vec{C}}|}$. Moreover, the selection of the source node and the destination nodes of different multicast traffic requests is assumed to be independent of each other.

Note that it is not easy to estimate the expected value of the lower bound presented in equation (7.3). As a work around, we approximate the lower bound as

$$\sum_{v \in V_{\vec{C}}} \left\lceil \frac{\chi_{G_{\mathcal{R}_v}}}{g} \right\rceil \approx \sum_{v \in V_{\vec{C}}} \left\lceil \frac{|\mathcal{R}_v|}{g} \right\rceil = \sum_{v \in V_{\vec{C}}} \left\lceil \frac{|\mathcal{M}_v|}{g} \right\rceil, \tag{7.23}$$

where, as defined before, $\mathcal{M}_v$ is the set of multicast traffic requests that have the ring vertex $v$ as either the source node or as one of the destination nodes, and $\mathcal{R}_v$ is the set of the corresponding directed paths on the unidirectional ring $\vec{C}$. This approximation holds when the conflict graph $G_{\mathcal{R}_v}$ is dense (which is the case in our simulations). Rather than estimating the expected value of the actual lower bound, we estimate the expected value of this approximate. Let us define $k_v$ to be the cardinality of the set $\mathcal{M}_v$, i.e., $k_v := |\mathcal{M}_v|$. It is easy to observe that the expected value of the approximate lower bound is given by

$$\mathrm{E}\left( \sum_{v \in V_{\vec{C}}} \left\lceil \frac{|\mathcal{M}_v|}{g} \right\rceil \right) = \mathrm{E}\left( \sum_{v \in V_{\vec{C}}} \left\lceil \frac{k_v}{g} \right\rceil \right) = \sum_{v \in V_{\vec{C}}} \mathrm{E}\left( \left\lceil \frac{k_v}{g} \right\rceil \right) = |V_{\vec{C}}| \cdot \mathrm{E}\left( \left\lceil \frac{k}{g} \right\rceil \right). \tag{7.24}$$

Here, for the final equality we are using the fact that in any multicast traffic request, ring vertices are selected with equal probability, therefore $k_v$'s are identically distributed. Hence, we can drop the subscript $v$ and assume that the number of multicast traffic requests that have ring vertex $v$ as either the source node or as one of the destination nodes is distributed according to random variable $k$.

In order to get the required estimate, we first observe that

$$\mathrm{E}\left( \frac{k}{g} \right) \leq \mathrm{E}\left( \left\lceil \frac{k}{g} \right\rceil \right) \leq \mathrm{E}\left( \frac{k}{g} + 1 \right). \tag{7.25}$$

Also, the number of multicast traffic requests selecting a particular ring vertex as the source node or as one of the destination nodes can be written as

$$k = \sum_{\{s,D\} \in \mathcal{M}} x_{\{s,D\}}, \tag{7.26}$$

where the random variable $x_{\{s,D\}}$ takes value 1 if the multicast traffic request $\{s, D\} \in \mathcal{M}$ selects the ring vertex under consideration as the source node or as one of the destination nodes, and 0 otherwise. We can evaluate $\mathrm{E}(k)$ as

$$\begin{aligned}\mathrm{E}(k) &= \sum_{\{s,D\}\in\mathcal{M}} \mathrm{E}(x_{\{s,D\}}) = \sum_{\{s,D\}\in\mathcal{M}} \mathrm{E}\big(\mathrm{E}(x_{\{s,D\}}|z_{\{s,D\}})\big) \\ &= \sum_{\{s,D\}\in\mathcal{M}} \mathrm{E}\left(\frac{z_{\{s,D\}}}{|V_{\vec{C}}|}\right) = \frac{1}{|V_{\vec{C}}|}\sum_{\{s,D\}\in\mathcal{M}} \mathrm{E}(z_{\{s,D\}}) = \frac{|\mathcal{M}|}{|V_{\vec{C}}|}\mu_{\mathcal{F}}. \end{aligned} \tag{7.27}$$

Here the third equality follows from the fact that given the size $z_{\{s,D\}}$ of the multicast traffic request $\{s, D\} \in \mathcal{M}$, the random variable $x_{\{s,D\}}$ is distributed according to a Bernoulli trial with the probability of success being $\frac{z_{\{s,D\}}}{|V_{\vec{C}}|}$. Equation (7.27) gives us

$$\mathrm{E}\left(\frac{k}{g}\right) = \frac{|\mathcal{M}|\mu_{\mathcal{F}}}{|V_{\vec{C}}|g}, \tag{7.28}$$

and

$$\mathrm{E}\left(\frac{k}{g}+1\right) = \frac{|\mathcal{M}|\mu_{\mathcal{F}}}{|V_{\vec{C}}|g} + 1. \tag{7.29}$$

Using equations (7.24), (7.25), (7.28) and (7.29), we can easily bound the required expectation as

$$\frac{|\mathcal{M}|\mu_{\mathcal{F}}}{g} \le \mathrm{E}\left(\sum_{v\in V_{\vec{C}}}\left\lceil\frac{|\mathcal{M}_v|}{g}\right\rceil\right) \le \frac{|\mathcal{M}|\mu_{\mathcal{F}}}{g} + |V_{\vec{C}}|. \tag{7.30}$$

If $\frac{|\mathcal{M}|\mu_{\mathcal{F}}}{g} \gg |V_{\vec{C}}|$ (which is the case in our simulations and is typically the case), then from equation (7.30), we note that the expected value of our lower bound can be approximated by $\frac{|\mathcal{M}|\mu_{\mathcal{F}}}{g}$, which is independent of the number of vertices on the unidirectional ring network. Moreover, we observe that the average value of the lower bound as determined by simulations is consistent with the above discussion and closely matches the estimate given in equation (7.30).

It should be clear that this behavior is mainly because the lower bound developed in equation (7.3) looks at each ring vertex in isolation. If we start considering pairs (or triplets, etc.) of ring vertices at a time, then the bound that we might develop will depend on the number of ring vertices. But it is not trivial to extend the bound in equation (7.3) and for the purpose of our discussion the presented bound suffices.

# Chapter 8

## Conclusion and Future Work

In this dissertation, we addressed the problem of routing and wavelength assignment, and traffic grooming for multicast traffic in all-optical WDM networks. Since both the problems are known to be hard for general network topologies, we confined our study to certain restricted topologies. In particular, we studied the problem of routing and wavelength assignment for multicast traffic in all-optical bidirected trees, and the problem of multicast traffic grooming in all-optical unidirectional rings.

The selected topologies are simple enough that algorithms can be developed and analyzed analytically, but still complex enough that the problems remain interesting. In particular, a strong motivation for selecting bidirected trees and unidirectional rings is that most of the fiber-optic networks that are presently deployed are either rings and trees or can be decomposed into these simple structures. Moreover, it is plausible that studying the problems on these simple topologies may give clues on how to attack the problems in more general network settings.

For the multicast routing and wavelength assignment problem in bidirected trees, we study the objective of minimizing the number of wavelengths required in the network. We argue that the topology determines the routing for each multicast traffic request, and any instance of the problem can be modeled as the problem of coloring a corresponding conflict graph of rooted subtrees on the given bidirected tree. The problem is shown to be hard even when the bidirected tree is restricted to being a bidirected star, or even a bidirected path. Since unicast routing and wavelength assignment is tractable for both these restricted topologies, these hardness results suggest that the multicast routing and wavelength assignment is inherently harder than the unicast case. Hence, simple extensions of the algorithms developed for the routing and wavelength assignment problem for unicast traffic may not work well for the multicast problem, and there is a need to develop and study algorithms designed specifically for the multicast case.

We present two algorithms, GREEDY-WA and SUBTREE-BASED-WA, for the problem of routing and wavelength assignment for multicast traffic in all-optical bidirected trees. GREEDY-WA is applicable to those instances of the problem in which the degree of the given bidirected tree is at most 3, and SUBTREE-BASED-WA is applicable to those instances of the problem in which the degree of the given bidirected tree is at most 4. GREEDY-WA proceeds in rounds, where in each round wavelengths are assigned to an appropriately selected subset of the given set of multicast traffic requests. As the name suggests, the wavelength assignment in each round is greedy in the sense that we try to use as few new wavelengths as possible. SUBTREE-BASED-WA overestimates the resources required by each multicast traffic request and then solves this overestimated problem. In particular, it models each given multicast traffic request as a subtree and generates a wavelength assignment from a minimum vertex coloring of the conflict graph of the subtrees. We analyze the worst case performance and the time complexity of both these algorithms. We prove that GREEDY-WA is a $\frac{5}{2}$-approximation algorithm for the problem of routing and wavelength assignment for multicast traffic in all-optical bidirected trees having degree at most 3. We also prove that SUBTREE-BASED-WA is an approximation algorithm with approximation ratio $\frac{10}{3}$, 3, and 2 for the problem of routing and wavelength assignment for multicast traffic in all-optical bidirected trees having degree at most 4, 3, and 2, respectively.

For the multicast traffic grooming problem in unidirectional rings, we study two different cost functions: (i) number of wavelengths required in the network, and (ii) number of ADMs required in the network. We restrict our study to the case when the bandwidth requirement of the individual multicast traffic requests are identical and are an integral fraction of the bandwidth available on individual wavelength channels. Even though the physical topology determines the routing for each multicast traffic request, the problem is still known to be hard for both the cost functions.

We present ARC-COL-BASED-TG, an algorithm for grooming multicast traffic requests in all-optical unidirectional rings. The algorithm treats each multicast traffic request as an arc on a circle and colors the resulting circular arc graph. Using the generated coloring, it trivially determines a traffic grooming solution for the original problem. We prove that if the circular arc coloring algorithm employed by ARC-COL-BASED-TG is an $\alpha$-approximation algorithm, then ARC-COL-BASED-TG has the same asymptotic approximation ratio for the traffic grooming problem when the objective is to minimize the number of wavelengths required in the network. We present RANDOM-TG, another simple scheme for grooming multicast traffic requests in all-optical unidirectional rings. As the name suggests, RANDOM-TG randomly generates a traffic grooming solution for the problem. We develop an easy to calculate lower bound on the number of ADMs required by any traffic grooming solution for any given instance of the problem, and then use this lower bound to analyze the worst case performance of ARC-COL-BASED-TG and RANDOM-TG, as determined by the number of ADMs required in the network. Next we present ITER-IMPROVE-TG, a new multicast traffic grooming scheme for all-optical unidirectional rings. ITER-IMPROVE-TG starts with a trivial traffic grooming solution and iteratively improves upon this. We study the time complexities of all the three schemes ARC-COL-BASED-TG, RANDOM-TG and ITER-IMPROVE-TG, and also compare their average performance, against each other and also against the lower bound that was developed, via simulations. During the simulation, we also study the performance of CIRCLE-BASED-TG, which is a multicast extension of the scheme developed in [11] for grooming unicast traffic in all-optical unidirectional ring networks. We discuss all the interesting observations made during the simulations.

It must be clear from this thesis that both routing and wavelength assignment as well as traffic grooming for multicast traffic in all-optical WDM networks are deep and rich areas for future research. There are several open research problems including some simple generalizations or further investigations of the problems addressed in this thesis, as well as various problems that are not exactly simple generalizations of the problems that we have studied, but are still very much related. Next, we briefly discuss some of these research directions. We shall not try to list all (or even a fraction) of these, but we hope to show that there are numerous open research problems available.

Some simple generalizations of the problems studied in this thesis that may be interesting to investigate include developing inapproximability results for the problem of multicast wavelength assignment in bidirected trees. Also, as of now, we have not done any tightness analysis for the approximation ratios of both the algorithms developed for the multicast wavelength assignment in bidirected trees. This could be an interesting question in itself. Extending the wavelength assignment algorithms to general trees is another challenging task. Investigating the use of other techniques (such as randomization) in order to develop better approximation algorithms for the problem of multicast wavelength assignment in bidirected trees is another possible direction for future research.

For the problem of multicast traffic grooming in unidirectional rings, we assumed sub-wavelength-continuity constraint. Studying the problem while relaxing this constraint is interesting because there are ADMs available that have times-lot exchange cards built into them. Another generalization could be to assume that the multicast traffic requests have non-uniform bandwidth requirements. This could model the case when the traffic is unsplittable. Obviously, an open problem is to develop a good approximation algorithm for the problem of multicast traffic grooming in unidirectional rings.

In this dissertation, we restricted our study to the topologies that fix the routing in the network. An interesting extension would be to study simple topologies where this is not so. An examples of such a topology is the bidirected ring. Another interesting variation could be to assume a constraint on the network resources such as ADMs, wavelengths, optical splitters, etc., and then try to determine a largest subset of the given set of multicast traffic requests that can be supported under these resource constraints. This problem can be easily extended by attaching weights to the given multicast traffic requests. As discussed before, both traffic grooming and routing and wavelength assignment come two flavors: static and dynamic. In this thesis, we have concentrated on the static problems. Developing algorithms for the dynamic version of both the problems is an interesting and challenging problem.

# Bibliography


[1] A. Leon-Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures.* McGraw-Hill, 2000.

[2] G. N. Rouskas and H. G. Perros, *A Tutorial on Optical Networks*, ser. LNCS. Springer, 2002, vol. 2497, pp. 155–194.

[3] J. S. Choi, N. Golmie, F. Lapeyrere, F. Mouveaux, and D. Su, "Classification of routing and wavelength assignment schemes in dwdm networks," in *7th International Conference on Optical Communications and Networks*, January 2000, pp. 1109–1115.

[4] K. Zhu and B. Mukherjee, "A review of traffic grooming in WDM optical networks: Architectures and challenges," *Optical Networks Magazine*, March/April 2003.

[5] R. Dutta and G. N. Rouskas, "Traffic grooming in WDM networks: Past and future," *IEEE Network*, Nov/Dec 2002.

[6] M. Ali and J. S. Deogun, "Cost-effective implementation of multicasting in wavelength-routed networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 18, no. 12, pp. 1628–1638, December 2000.

[7] Y. Zhou and G.-S. Poo, "Optical multicast over wavelength-routed wdm networks: A survey," *Optical Switching and Networking*, vol. 2, no. 3, pp. 176–197, November 2005.

[8] H. Zang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks," *SPIE/Baltzer Optical Networks Magazine*, vol. 1, no. 1, pp. 47–60, 2000.

[9] A. Rawat, M. Shayman, R. La, and S. Marcus, "Coloring rooted subtrees on a bounded degree host tree," Institute for Systems Research, University of Maryland, Tech. Rep. TR 2008-10, 2007.

[10] ——, "Minimum wavelength assignment for multicast traffic in all-optical WDM trees," in *Proceedings of the 5th International Conference on Broadband Communications, Networks, and Systems*, September 2008.

[11] X. Zhang and C. Qiao, "An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings," *IEEE/ACM Transactions On Networking*, October 2000.

[12] A. Rawat, R. La, S. Marcus, and M. Shayman, "Grooming multicast traffic in unidirectional SONET/WDM rings," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. S-6, pp. 70–83, August 2007.

[13] T. A. McKee and F. R. McMorris, *Topics in Intersection Graph Theory.* Philadelphia: SIAM, 1999.

[14] M. C. Golumbic, M. Lipshteyn, and M. Stern, "The edge intersection graphs of paths in a tree," *Journal of Combinatorial Theory, Series B*, vol. 38, no. 1, pp. 8–22, February 1985.

[15] I. Holyer, "The np-completeness of edge-coloring," *SIAM Journal on Computing*, vol. 10, no. 4, pp. 718–720, 1981.

[16] T. Erlebach and K. Jansen, "The complexity of path coloring and call scheduling," *Theoretical Computer Science*, vol. 255, no. 1-2, pp. 33–50, March 2001.

[17] R. Tarjan, "Decomposition by clique separators," *Discrete Mathematics*, vol. 55, no. 2, pp. 221–232, 1985.

[18] P. Raghavan and E. Upfal, "Efficient routing in all-optical networks," in *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, 1994, pp. 134–143.

[19] M. Mihail, C. Kaklamanis, and S. Rao, "Efficient access to optical bandwidth," in *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, 1995, p. 548.

[20] T. Nishizeki and K. Kashiwag, "On the 1.1 edge-coloring of multigraphs," *SIAM Journal on Discrete Mathematics*, vol. 3, no. 3, pp. 391–410, August 1990.

[21] P. Sanders and D. Steurer, "An asymptotic approximation scheme for multigraph edge coloring," in *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005, pp. 897–906.

[22] T. Erlebach and K. Jansen, "Scheduling of virtual connections in fast networks," in *Proceedings of the 4th Parallel Systems and Algorithms Workshop*, 1996.

[23] S. R. Kumar, R. Panigrahy, A. Russell, and R. Sundaram, "A note on optical routing in trees," *Information Processing Letters*, vol. 62, no. 6, pp. 295–300, June 1997.

[24] C. Kaklamanis and G. Persiano, "Efficient wavelength routing on directed fiber trees," in *Proceedings of the 4th Annual European Symposium on Algorithms*, ser. LNCS, vol. 1136. Springer, 1996, pp. 460–470.

[25] V. Kumar and E. J. Schwabe, "Improved access to optical bandwidth in trees," in *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1997, pp. 437–444.

[26] C. Kaklamanis, P. Persiano, T. Erlebach, and K. Jansen, "Constrained bipartite edge coloring with applications to wavelength routing," in *Proceedings of the 24th International Colloquium on Automata, Languages and Programming*, ser. LNCS, vol. 1256. Springer, 1997, pp. 493–504.

[27] T. Erlebach and K. Jansen, "Call scheduling in trees, rings and meshes," in *Proceedings of the 30th Hawaii International Conference on System Sciences*, 1997, p. 221.

[28] I. Caragiannis, C. Kaklamanis, and P. Persiano, "Bounds on optical bandwidth allocation in directed fiber tree technologies," in *2nd Workshop on Optics and Computer Science*, April 1997.

[29] K. Jansen, "Approximation results for wavelength routing in directed binary trees," in *2nd Workshop on Optics and Computer Science*, April 1997.

[30] V. Auletta, I. Caragiannis, C. Kaklamanis, and P. Persiano, "Randomized path coloring on binary trees," in *Proceedings of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, ser. LNCS, vol. 1913. Springer, September 2000, pp. 407–421.

[31] R. E. Jamison and H. M. Mulderb, "Constant tolerance intersection graphs of subtrees of a tree," *Discrete Mathematics*, vol. 290, no. 1, pp. 27–46, January 2005.

[32] D. R. Fulkerson and O. A. Gross, "Incidence matrices and interval graphs," *Pacific Journal of Mathematics*, vol. 15, no. 3, pp. 835–855, 1965.

[33] F. Gavril, "Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 180–187, 1972.

[34] M. C. Golumbic, M. Lipshteyn, and M. Stern, "Representations of edge intersection graphs of paths in a tree," in *Proceedings of the European Conference on Combinatorics, Graph Theory and Applications*, ser. Discrete Mathematics and Theoretical Computer Science, 2005, pp. 87–92.

[35] R. B. Hayward, "Weakly triangulated graphs," *Journal of Combinatorial Theory, Series B*, vol. 39, no. 3, pp. 200–208, December 1985.

[36] R. B. Hayward, J. P. Spinrad, and R. Sritharan, "Improved algorithms for weakly chordal graphs," *ACM Transactions on Algorithms*, vol. 3, no. 2, May 2007.

[37] M. C. Golumbic, M. Lipshteyn, and M. Stern, "Finding intersection models of weakly chordal graphs," in *Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science*, ser. LNCS, vol. 4271. Springer, 2006, pp. 241–255.

[38] I. Caragiannis, C. Kaklamanis, and P. Persiano, “Wavelength routing in all-optical tree networks: A survey,” *Computing and Informatics (formerly Computers and Artificial Intelligence)*, vol. 20, no. 2, pp. 95–120, 2001.

[39] ——, “Approximate path coloring with applications to wavelength assignment in wdm optical networks,” in *Proceedings of the 21st International Symposium on Theoretical Aspects of Computer Science*, ser. LNCS, vol. 2996. Springer, 2004, pp. 258–269.

[40] ——, *Approximation Algorithms for Path Coloring in Trees*, ser. LNCS. Springer, 2006, vol. 3484, pp. 74–96.

[41] B. Bollobas, *Modern Graph Theory.* Springer, 1998.

[42] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press and McGraw-Hill, 2001.

[43] D. J. Rose, R. E. Tarjan, and G. S. Lueker, “Algorithmic aspects of vertex elimination on graph,” *SIAM Journal on Computing*, vol. 5, no. 2, pp. 266–283, 1976.

[44] S. Olariu, “An optimal greedy heuristic to color interval graphs,” *Information Processing Letters*, vol. 37, no. 1, pp. 21–25, January 1991.

[45] R. Cole and J. E. Hopcroft, “On edge coloring bipartite graphs,” *SIAM Journal on Computing*, vol. 11, no. 3, pp. 540–546, 1982.

[46] R. M. Karp, “Reducibility among combinatorial problems,” in *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

[47] M. R. Garey, D. S. Johnson, G. L. Miller, and C. H. Papadimitriou, “The complexity of coloring circular arcs and chords,” *SIAM Journal on Matrix Analysis and Applications*, vol. 1, no. 2, pp. 216–227, 1980.

[48] V. E. Alekseev and V. V. Lozin, “Independent sets of maximum weight in (p,q)-colorable graphs,” *Discrete Mathematics*, vol. 265, no. 1-3, pp. 351–356, April 2003.

[49] R. Diestel, *Graph Theory.* New York: Springer, 2005.

[50] A. L. Chiu and E. H. Modiano, “Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks,” *IEEE/OSA Journal of Lightwave Technology*, January 2000.

[51] X. Zhang and C. Qiao, “On scheduling all-to-all personalized connections and cost-effective designs in WDM rings,” *IEEE/ACM Transactions On Networking*, June 1999.

[52] J. Wang, W. Cho, V. R. Vemuri, and B. Mukherjee, "Improved approaches for cost-effective traffic grooming in WDM rings: ILP formulations and single-hop and multihop connections," *IEEE/OSA Journal of Lightwave Technology*, November 2001.

[53] X. Zhang and C. Qiao, "Scheduling in unidirectional WDM rings and its extensions," in *Proceedings of SPIE All Optical Communication Systems: Architecture, Control and Network Issues III*, 1997.

[54] A. R. B. Billah, B. Wang, and A. A. S. Awwal, "Effective traffic grooming in WDM rings," in *Proceedings of IEEE Globecom*, 2002.

[55] O. Gerstel and R. Ramaswami, "Cost-effective traffic grooming in WDM rings," *IEEE/ACM Transactions of Networking*, October 2000.

[56] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE Journal on Selected Areas in Communications*, January 2002.

[57] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," *IEEE/ACM Transactions on Networking*, April 2003.

[58] F. Farahmandz, X. Huang, and J. P. Jue, "Efficient online traffic grooming algorithms in WDM mesh networks with drop-and-continue node architecture," in *Proceedings of IEEE Broadnets*, October 2004.

[59] X. Huang, F. Farahmandz, and J. P. Jue, "An algorithm for traffic grooming in WDM mesh networks with dynamically changing light-trees," in *Proceedings of IEEE Globecom*, 2004.

[60] G. V. Chowdhary and C. S. R. Murthy, "Grooming of multicast sessions in WDM mesh networks," in *Proceedings of IEEE Broadnets*, October 2004.

[61] X. Huang, F. Farahmandz, and J. P. Jue, "Multicast traffic grooming in wavelength-routed WDM mesh networks using dynamically changing light-trees," *IEEE/OSA Journal of Lightwave Technology*, October 2005.

[62] A. Khalil, C. Assi, A. Hadjiantonis, G. Ellinas, and M. A. Ali, "On multicast traffic grooming in WDM networks," in *Proceedings of IEEE International Symposium on Computers and Communications*, June 2004.

[63] A. Khalil, C. Assi, A. Hadjiantonis, G. Ellinas, N. Abdellatif, and M. A. Ali, "Multicast traffic grooming in WDM networks," in *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering*, May 2004.

[64] A. R. B. Billah, B. Wang, and A. A. S. Awwal, "Multicast traffic grooming in WDM optical mesh networks," in *Proceedings of IEEE Globecom*, December 2003.

[65] G. V. Chowdhary and C. S. R. Murthy, “Dynamic multicast traffic engineering in WDM groomed mesh networks,” in *Proceedings of IEEE Broadnets*, October 2004.

[66] A. Khalil, A. Hadjiantonis, G. Ellinas, and M. Ali, “Sequential and hybrid grooming approaches for multicast traffic in WDM networks,” in *Proceedings of IEEE Globecom*, 2004.

[67] H. V. Madhyastha, N. Srinivas, G. V. Chowdhary, and C. S. R. Murthy, “Grooming of multicast sessions in WDM ring networks,” in *Proceedings of SPIE Opticomm*, October 2003.

[68] V. Kumar, “An approximation algorithm for circular arc coloring,” *Algorithmica*, 2001.

[69] I. A. Karapetian, “On the coloring circular arc graphs,” *Docladi (Reports) of the Academy of Science of the Armenian Soviet Socialist Republic*, 1980.

[70] A. Tucker, “Coloring a family of circular arcs,” *SIAM Journal of Applied Mathematics*, 1975.

[71] M. Valencia-Pabon, “Revisiting tucker’s algorithm to color circular arc graphs,” *SIAM Journal on Computing*, vol. 32, no. 4, pp. 1067–1072, 2003.