ABSTRACT

CHEN, BENSONG. Hierarchical Traffic Grooming in Large-Scale WDM Networks. (Under the direction of Professor George N. Rouskas and Professor Rudra Dutta.)

The advances in fiber optics and wavelength division multiplexing (WDM) technology are viewed as the key to satisfying the data-driven bandwidth demand of today's Internet. The mismatch of bandwidths between user needs and wavelength capacity makes it clear that some multiplexing should be done to use the wavelength capacity efficiently, which will result in reduction on the cost of line terminating equipment (LTE). The technique is referred to as *traffic grooming*.

Previous studies have concentrated on different objectives, or on some special network topologies such as rings. In our study, we aim at minimizing the LTE cost to directly target on minimizing the network cost. We look into the grooming problem in elemental topologies as a starting point. First, we conduct proofs to show that traffic grooming in path, ring and star topology networks with the cost function we consider is NP-Complete. We also show the same complexity results for a Min-Max objective that has not been considered before, on the two elementary topologies. We then design polynomial-time heuristic algorithms for the grooming problem in rings (thus implicitly paths) and stars for networks of larger size. Experiments on various network sizes and traffic patterns show the effectiveness of our algorithms.

For general topology networks, we design a hierarchical approach which first partitions a large network into several clusters, then applies the method we use in star networks to each cluster, by selecting a hub node to groom traffic within the cluster. At the second hierarchy, we apply the star grooming method again only on the hub nodes. The polynomial-time hierarchical approach scales well and can cope with large networks of general topology efficiently, both for minimizing LTE cost and for lowering wavelength requirements.

We also design a clustering algorithm that can generate good results for subsequent steps in the hierarchical grooming method. Numerical results from experiments show that both the hierarchical grooming approach and our clustering algorithm generate satisfying results for the grooming on various network topologies and traffic demand patterns.

Hierarchical Traffic Grooming in Large-Scale WDM Networks

by

Bensong Chen

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Doctor of Philosophy

Department of Computer Science

Raleigh

2005

Committee Members:

Dr. Carla D. Savage	Dr. Matthias F. M. Stallmann
Dr. George N. Rouskas Chair of Advisory Committee	Dr. Rudra Dutta Co-chair of Advisory Committee

To my dear wife, **Jielin**,
And my parents and sister,
YiE Chen, **Yarong Zhang**, **Benling Chen**,
for their endless love and support.

Biography

Bensong Chen was born in Harbin, China in 1975. He received his B.S. in Management Information Systems from Northern Jiaotong University, Beijing, China in 1998. Then, he continued his study and graduated with a M.E. from the Institute of Software, Chinese Academy of Sciences, Beijing, China in 2001.

Since fall 2001, he has been a full-time PhD Student at the Computer Science Department of North Carolina State University, Raleigh, North Carolina, U.S.

Bensong is married and lives in Winston Salem, NC.

Acknowledgements

I am very lucky to have two great advisors, Dr. George N. Rouskas and Dr. Rudra Dutta, who have been giving their expert guidance throughout my research. Besides their instructions that lead me into the path of the academic world, I am also very grateful for their kindness and understanding, which allow me to have a flexible commuting schedule during the years.

The other two members in my advisory committee, Dr. Carla D. Savage and Dr. Mathias F. M. Stallmann, have been the instructors of the most valuable courses I have taken in the graduate school. I also learned the most valuable theories for my research in these courses. I would like to thank them for their continuous assistance on applying the theories into this research.

The research in this thesis was supported in part by National Science Foundation grant ANI-0322107.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 WDM Optical Networks

Wavelength division multiplexing (WDM) technology has the potential to satisfy the ever-increasing bandwidth needs of network users on a sustained basis. WDM is the process of transmitting data simultaneously at multiple carrier wavelengths over an optical fiber cable. The wavelengths are kept sufficiently far apart so that they do not interfere with each other. Thus, a single strand of fiber can be viewed as a collection of high capacity *virtual fibers*. Today, WDM systems are widely deployed in long-haul networks, and have a major presence in the metro-area networks as well. Passive Optical Networks (PONs) further extend optical networks to the end home users. Figure 1.1 is an example of an optical backbone network that connects many optical ring access networks.

In the backbone part of the figure, some *lightpaths* are set up which are shown in the color lines with arrowheads. A *lightpath* is defined as a clear channel (or *wavelength* in this case) in which the signal remains in optical form throughout the physical path between two end nodes. The set of lightpaths defines a *logical topology*, which can be designed to optimize some performance measurements for a given set of traffic demands. User demands between node pairs are then further carried on that logical topology layer, either with direct lightpath connections, or be multiplexed and carried on multiple hops from source to destination. Note that we use different colors for lightpaths using different wavelengths

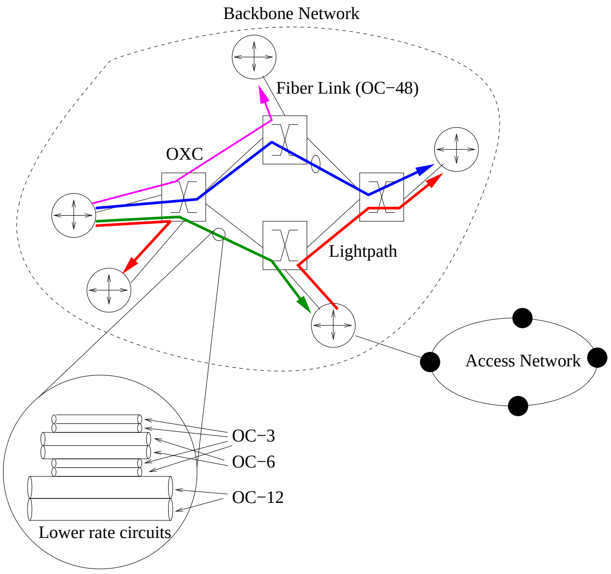


Figure 1.1: A Typical WDM Optical Mesh Network

just to facilitate illustration. In fact, all wavelengths currently used in WDM networks are invisible.

In WDM networks, nodes are equipped with *optical cross-connects* (OXCs), devices which can optically switch wavelengths, thus making it possible to establish direct *lightpath* connections between any pair of nodes that are not directly connected with a physical fiber. An OXC can cross-connect different wavelengths from the input ports to the output ports, where the connection pattern of each wavelength is independent of the others. Figure 1.2 shows the logical function of an OXC in a WDM network node. Note that for each wavelength, an optical switch is required to do optical signal switching, which is typically a mapping from the input ports to the output ports. A typical realization of the optical switch is through a Micro Electro-Magnetic System (MEMS) mirror matrix. Control signals can be sent to each mirror to adjust its reflecting angle, so laser beams can be switched to different ports. Wavelength multiplexer/demultiplexer is used for each fiber to separate different wavelengths before sending them to corresponding switches, and recombine them after the switching.

From the figure, we also see that some ‘red’ signals are added to the OXC, and some dropped to the local node. They carry traffic originating or terminating at the network node itself, as well as some traffic that is electronically switched by the node.

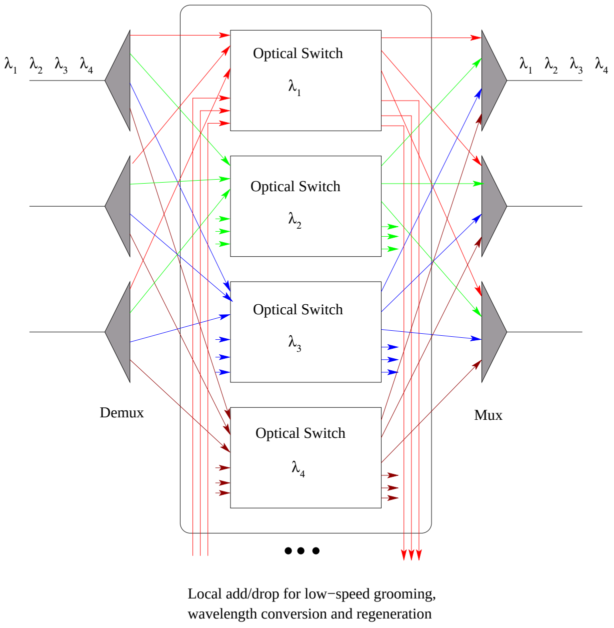

λ_1 λ_2 λ_3 λ_4
Optical Switch
λ_1
Optical Switch
λ_2
Optical Switch
λ_3
Optical Switch
λ_4
λ_1 λ_2 λ_3 λ_4
Demux
Mux
Local add/drop for low−speed grooming,
wavelength conversion and regeneration

Figure 1.2: Optical Cross-Connect, OXC, with local add/drop

In SONET WDM ring networks, OXCs are called *Optical Add/Drop Multiplexers* (OADMs). Because the ring topology assumes a single incoming and outgoing fiber at each direction, OADMs are simpler than general OXCs. The function of OADM in SONET is illustrated in Figure 1.3.

Since each network node needs to terminate data destined to the node itself, and initiate traffic originated from it to other network nodes, devices are required to add/drop signals to/from the corresponding lightpaths. Since user data are expressed in electronic signals, some kind of transformation between optical and electronic signals needs to be done. In WDM networks, the OEO (opto-electro-optical) transformation is realized in devices called *line terminating equipments* (LTEs) at each network node. Another type of devices, *digital cross-connects* (DXCs) can further switch the electronic signals to rearrange data multiplexing onto the wavelengths. Figure 1.4 shows a whole picture on the relationships of the three kinds of devices.

With the deployment of commercial WDM systems, it has become apparent that the cost of optical components, especially LTEs, dominates the cost in designing optical networks, and is a more meaningful metric to optimize than, say, the number of wavelengths

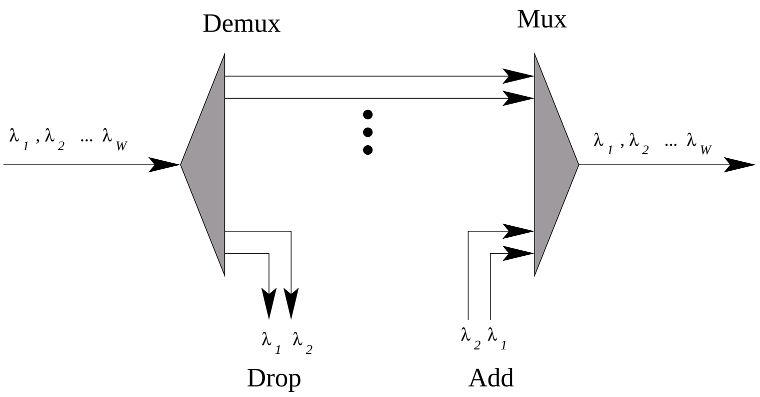


Figure 1.3: Optical Add-Drop Multiplexer, OADM in SONET WDM networks

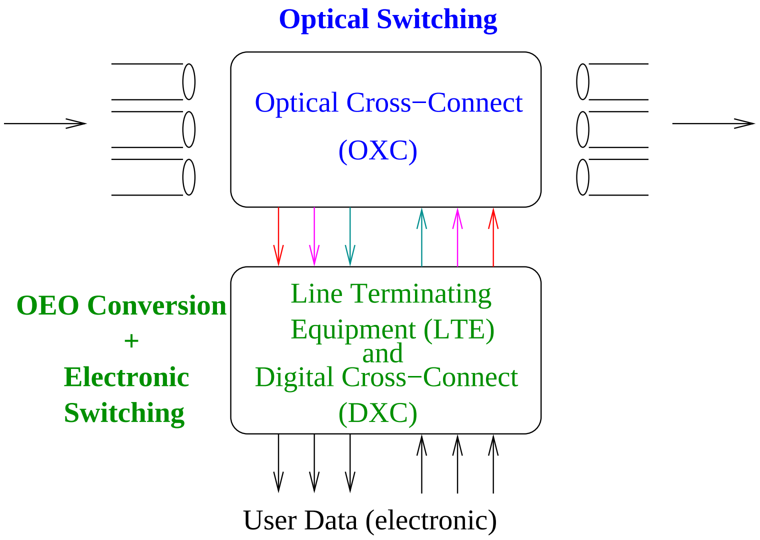


Figure 1.4: Structure of an optical network node

(with today's technology, hundreds of wavelengths can be multiplexed into one fiber). With currently available optical technology, the data rate (bandwidth) of each wavelength is in the order of 2.5-10 Gbps, while 40 Gbps rates are becoming commercially available. In the Optical Carrier (OC) hierarchy, 40 Gbps transmission rate corresponds to OC-768. In order to utilize bandwidth more effectively, new models of optical networks allow several independent traffic streams to *share* the capacity of a lightpath. These observations give rise to the concept of *traffic grooming*, a variant of logical topology design, which is concerned with the development of techniques for combining lower speed components onto wavelengths in order to minimize network cost. In Figure 1.1, the magnified part shows how lower-rate signals OC-3 (155.52 Mbps) and OC-6 are multiplexed onto a higher-rate OC-12 carrier. This is realized by time division multiplexing (TDM) technology, which uses different time slots on a high-rate channel to transmit different lower-rate data together.

Traffic grooming research has, in general, followed one of two directions. In *dynamic* grooming, it is assumed that the node grooming capabilities (in terms of available electronic ports, level of wavelength conversion, and switching capacity) are fixed and known, and the goal is to develop on-line algorithms for grooming and routing of connection requests that arrive in real time. Typical solution approaches transform the grooming problem into a shortest path problem on a new layered graph modeling both the underlying physical topology and the capabilities of individual nodes.

In this thesis, we concentrate on studying the other type of grooming, called *static* traffic grooming. In static grooming, the starting point is the set of (forecast) long-term traffic demands, and the objective is to provision the network nodes to carry all the demands while minimizing the overall network cost. Due to their different nature, the assumptions, goals and approaches are also varied.

Given the wide deployment of SONET/SDH technology and the immediate practical interest in upgrading this infrastructure to WDM, early research in traffic grooming focused a lot on ring topologies. Since the development of MPLS and GMPLS standards makes it possible to aggregate a set of MPLS streams for transport over a single lightpath, the recent research concentration has shifted to the problem of traffic grooming in general network topologies for various optimization objectives. Nonetheless, study on simple topologies gives a good starting point for looking into the complex problem more deeply.

1.2 Contributions

We research the static version of the traffic grooming problem in this work. Earlier research on this topic concentrated mainly on some objectives other than the network cost, such as minimizing the wavelength requirement, minimizing the total amount of electronic switching, or maximizing the traffic demands that can be carried within physical constraints. They do not directly aim at minimizing the network cost. Some earlier works did consider minimizing the amount of LTE in the network, but most of them have restrictions on the network topologies, confining the topologies to rings or hypercubes.

In this work, we consider the objective of minimizing the network cost, not only on elementary topologies such as paths, rings and stars, but also on general topology networks of large size. We find that there is not much related work in the literature so far. Also, in addition to the traditional objective of minimizing the overall network cost, we consider a *Min-Max* objective, which is to minimize the maximum amount of LTE at each network node, on the symmetric simple topologies (rings and stars) in our study. This is because the traditional *aggregate* objective may result in a solution in which some nodes end up with a (very) large amount of LTE while some others have only a small amount of LTE. Such a solution may have a number of undesirable properties. First, a node that requires a large amount of LTE may be too expensive or even impractical to deploy (e.g., due to high interconnection costs, high power consumption, or space requirements). Second, the resulting network can be highly heterogeneous in terms of the capabilities of individual nodes, making it difficult to operate and manage. Third, and more important, a solution minimizing the total LTE cost can be extremely sensitive to the assumptions regarding the traffic pattern, as previous studies [21] have demonstrated. Specifically, a solution that is optimal for a given set of traffic demands may be far away from optimal for a different set. Since LTE involve expensive hardware devices that are difficult to move from one node to another on demand, an approach that attempts to minimize total LTE cost may not be appropriate for dimensioning a network, unless the network operator has a clear picture of traffic demands far into the future *and* these traffic demands are unlikely to change substantially over the life of the network.

The Min-Max objective, however, can solve the problem by balancing the nodal capabilities. This is important especially in the symmetric ring and star topologies. In particular, an approach that minimizes the maximum LTE cost at any network node is likely to be attractive because, from practical design considerations, all the network nodes are likely to be provisioned with identical equipment. Effectively, all nodes will have a cost that is dictated by the node with the maximum LTE. Such a homogeneous network is easier to operate, manage and maintain, and is likely to be less expensive than a heterogeneous one due to the economies of scale that can be achieved when all nodes are subject to identical specifications. Furthermore, such an optimization approach can be of great importance to dimensioning the network for *unknown and/or dynamic future demands*. Specifically, the network designer may solve the optimization problem for a wide range of traffic scenarios, and equip each node with an amount of LTE equal to the highest solution obtained (plus a certain *fudge factor* for making the solution robust in the face of future demands). Alternatively, network designers and operators may choose to minimize the maximum LTE as well as overall LTE cost in the whole system, which we also take into consideration in our research.

Starting from the grooming problems on the simpler path and star networks, we first give proofs of NP-Completeness for the cost objectives we consider, which has not been done before this work. Then, we model the problems with Integer Linear Programming (ILP) formulations, and program them with a solver tool called CPLEX [1], so we can get optimal solutions for small-size networks in reasonable time. We then design polynomial-time heuristic algorithms that can give near-optimal results for large networks as well. Experimental results show that the solutions are close to optimal, and close to the lower bounds for instances where the optimal cannot be obtained in reasonable time constraints.

With deeper understanding of the grooming problem in simple network topologies, we concentrate our research on grooming in networks with general mesh topology. Inspired by the airline traffic strategy, we design a hierarchical grooming approach to cope with large general networks, by partitioning the nodes into clusters, and applying the lightpath design in our star grooming algorithm to traffic within each cluster, and to the inter-cluster traffic at the second level of the hierarchy as well.

For networks that are too large for manual clustering, we also come up with an automatic clustering algorithm that facilitates generating good final solutions in the entire hierarchical approach. We find methods to calculate lower bounds for the problem instances we experiment on, and numerical results show that the hierarchical grooming approach and our clustering algorithm are effective in lowering the electronic port cost in the whole network, as well as keeping the wavelength requirements close to the necessary bounds, which implies solutions more likely to fall within the given wavelength availability.

1.3 Structure of the Thesis

The thesis is organized as follows: In Chapter 2, we define the traffic grooming problem both in words and mathematical formulations. In Chapter 3, we give a survey on previous work in the literature on traffic grooming and related topics. Chapter 4 and 5 discuss grooming for elemental network topologies, with Chapter 4 concentrating on rings and paths, and Chapter 5 on star networks. Chapter 6 introduces the hierarchical grooming approach for general topology networks, and Chapter 7 further explores the clustering algorithms for the hierarchical grooming method. We conclude our work and discuss future directions in Chapter 8.

Chapter 2

The Traffic Grooming Problem

In this chapter, we first review the well-known *routing and wavelength assignment* (RWA) problem and some conclusions from previous study in the literature, and then define the *traffic grooming* problem with the context we consider in this thesis. An Integer Linear Programming (ILP) formulation is given for formalizing and better understanding the traffic grooming problem. As we mentioned, this work concentrates on the *static* traffic grooming and related problems.

2.1 The RWA Sub-problem

An optical network can be abstracted as a directed graph, with vertices representing network nodes (sites), connected with directed edges showing the optical fiber links. A traffic demand from node s to node d (denoted as $t^{(sd)}$ can be carried on a certain physical route of fiber links, and by a specific wavelength channel on each of the links. If we consider a set of such demands, which forms a traffic demand matrix T, the RWA problem becomes how to route all the traffic demands from the respective sources to the destinations using the available wavelengths. Since data carried for each demand are different, this is a *multicommodity flow* problem. To avoid signal collisions, if two demands share a same fiber link, they must be carried on different wavelengths. One objective of the RWA problem considered in the literature is to satisfy all traffic demands in T, while minimizing the number of wavelengths used in the whole network. Generally, RWA problem assumes no use of wavelength converters in the network, that is, each traffic demand is carried on the same wavelength with a single lightpath throughout the route.

Previous studies show that wavelength assignment to minimize the number of wavelengths can be solved in polynomial time in paths and stars. In addition, it is easy to see that for the general tree topology, each pair of nodes is joined by a unique path, which means that routing is fixed in trees. If the network topology is a path, wavelength assignment is equivalent to the Interval Graph Coloring problem, which can be solved in linear time by a greedy algorithm [31]. If the topology is a star, it is equivalent to Minimum Edge Coloring in a bipartite graph, which is solvable in polynomial time [54]. For wavelength assignment on trees to minimize the total number of wavelengths, the problem is NP-hard [15]. In unidirectional ring networks, wavelength assignment is equivalent to the *arc-coloring* problem, which is NP-Hard [52]; In bidirectional rings, even routing is not determined, and the problem is known to be NP-hard [55]. Following directly from NP-hardness of any special case (for instance, the ring topology), the RWA problem is NP-hard in general network topologies [32] as well.

2.2 The Traffic Grooming Problem

Current optical technologies allow for multiplexing lower-rate traffic streams onto the same wavelength using time-division multiplexing. If we require that only traffic belonging to the same source/destination pair could be multiplexed onto the same lightpath, it is equivalent to the RWA problem discussed in the previous section. However, this constraint means that we have to set up direct lightpaths for each source/destination pair, which is generally impractical due to the wavelength constraints or optical device constraints given the quadratic growth rate on possible lightpaths as networks grow larger.

For that reason, each node in the optical network needs to do both optical and electronic switching. In Chapter 1, we have shown the topology of such a network node in Figure 1.4. The network node let some lightpaths pass through with only optical switching, but terminate/originate other lightpaths. Some traffic may be switched electronically onto new lightpaths to be carried to its destination. The electronic switching with multiplexing/demultiplexing is also called *grooming*, which better utilizes the given wavelength capacity, reduces wavelength requirements, and enhances *virtual connectivity.* Expensive opto-electro-optical devices (e.g., line terminating equipment) and electronic switches (digital cross-connects) need to be equipped at the network nodes, to do signal transformation/rearrangement between the two forms. The traffic grooming problem is thus defined for balancing the advantages and costs associated with different grooming design.

We define a positive integer C as the capacity of one wavelength, expressed as units of some basic transmission rate (such as OC-3). The capacity C is also called the *grooming factor.* Let W be the number of wavelengths each fiber can carry concurrently. A traffic demand matrix $T = [t^{(sd)}]$ can be defined, where integer $t^{(sd)}$ denotes the number of basic transmission rates from node s to node d. (We allow the traffic demands to be greater than the capacity of a lightpath, i.e., it is possible that $t^{(sd)} > C$ for some s, d.) Given the traffic matrix, the traffic grooming problem involves the following conceptual sub-problems (SPs):

1. *logical topology SP:* find a set R of lightpaths that forms a virtual topology,

2. *lightpath routing and wavelength assignment SP:* solve the RWA problem on R, and

3. *traffic routing SP:* route each traffic stream through the lightpaths in R.

The first and third sub-problems together constitute the grooming aspect of the problem. Also, the number W of wavelengths per fiber link is taken into consideration as a constraint rather than as a parameter to be minimized.

The optimization goal is to minimize the overall cost of LTEs in the network, which is the same as minimizing the number of lightpaths established in the system (since each lightpath requires LTE at both ends). Note that this objective is equivalent to minimizing the number of edges in the logical topology formed by lightpaths.

As we mentioned, another goal we consider for symmetric elemental topologies is to minimize the *maximum number of lightpaths originating from or terminating at any node.* In our cost model, one unit of cost is incurred for each lightpath that terminates at, or originates from, a network node. Thus, this cost metric accurately reflects the amount of LTE needed at each network node, and therefore our objective is to minimize the LTE cost at the node where it is maximum. This is the same as minimizing the *maximum nodal degree* in the logical topology.

2.3 ILP Formulations for the Traffic Grooming Problem

The traffic grooming problem can be modeled mathematically. In fact, an exact Integer Linear Program (ILP) formulation can be defined to describe the problem. The formulation comes from previous work in [22]. It allows better understanding of the subproblems and their relationships we just describe in 2.2. We include it here for the sake of completeness, and for comparison to the ILP for special network topologies we will address in the subsequent chapters.

Given:

The physical topology, a network with N nodes, $0, 1, \ldots, N-1$, connected by bi-directional single fiber links. We use $p_{lm} \in \{0,1\}$ to indicate whether a physical link exists from node l to m.

The traffic matrix $T = [t^{(sd)}], s, d \in \{0, \cdots, (N-1)\}$, $t^{(sd)} \in \{0, 1, 2, \cdots\}, t^{(ss)} = 0, \forall s$.

The wavelength limit W, which is the number of distinct wavelengths each link can carry, and **wavelength capacity** C, the number of unit traffic rates that can be multiplexed onto a single wavelength channel.

Find:

A detailed grooming solution, in terms of lightpath counts b_{ij} for the number of direct lightpaths from node i to node j, of which $b_{ij}(l,m)$ is the number of lightpaths that traverse the physical link from l to m; lightpath wavelength indicators $c_{ij}^{(k)}(l,m) \in \{0,1\}, k \in \{0, \ldots, W-1\}$ that indicates the wavelength assigned for each lightpath; and integral traffic routing (grooming) variables $t_{ij}^{(sd)}$, the amount of traffic demands from s to d that is carried in lightpath (i,j).

Subject to:

Physical Topology Constraints:

$$b_{ij}(l,m) \leq b_{ij} p_{lm}, \forall i, j, l, m \tag{2.1}$$

$$c_{ij}^{(k)}(l,m) \leq p_{lm}, \forall i, j, k, l, m \tag{2.2}$$

Lightpath Routing SP Constraints:

$$\sum_l b_{ij}(m,l) - \sum_l b_{ij}(l,m) = \left\{ \begin{array}{ll} b_{ij}, & m = i \\ -b_{ij}, & m = j \\ 0, & m \neq i, m \neq j \end{array} \right\}, \forall m, (i,j) \tag{2.3}$$

$$\sum_{i,j} b_{ij}(l,m) \leq W, \forall l, m \tag{2.4}$$

Wavelength Assignment SP Constraints:

$$\sum_k c_{ij}^{(k)}(l,m) \quad = \quad b_{ij}(l,m), \forall i,j,l,m \tag{2.5}$$

$$\sum_{i,j} c_{ij}^{(k)}(l,m) \quad \leq \quad 1, \forall k,l,m \tag{2.6}$$

$$\sum_l c_{ij}^{(k)}(m,l) - \sum_l c_{ij}^{(k)}(l,m) \qquad \left\{ \begin{array}{ll} \leq b_{ij}, & m = i \\ \geq -b_{ij}, & m = j \\ = 0, & m \neq i, m \neq j \end{array} \right\}, \forall i,j,k,m \tag{2.7}$$

Traffic Grooming SP Constraints:

$$t_{ij} \quad = \quad \textstyle\sum_{sd} t_{ij}^{(sd)}, \quad \forall (i,j) \tag{2.8}$$

$$t_{ij} \quad \leq \quad b_{ij}C, \quad \forall (i,j) \tag{2.9}$$

$$\sum_j t_{ij}^{(sd)} - \sum_j t_{ji}^{(sd)} = \left\{ \begin{array}{ll} t^{(sd)}, & s = i \\ -t^{(sd)}, & d = i \\ 0, & s \neq i, d \neq i \end{array} \right\} \forall i, (s,d) \tag{2.10}$$

To minimize:(one of the following functions)

Total number of lightpaths:

$$\sum_{i,j} b_{ij} \tag{2.11}$$

Maximum number of lightpaths at a node:

$$\max_i \left(\max \left(\sum_j b_{ji}, \sum_j b_{ij} \right) \right) \tag{2.12}$$

Total amount of electronic switching:

$$\sum_{s,d,i,j} t_{ij}^{(sd)} - \sum_{s,d} t^{(sd)} \tag{2.13}$$

Most of the above constraints are self-explanatory. Constraint (2.4) is implicit in constraints (2.3), (2.5), and (2.6), but would be needed if full wavelength conversion capability were available at each node of the network, in which case there would be no wavelength assignment constraints at all. In addition, if we allow some wavelength conversion capability at some nodes, the corresponding wavelength continuity constraints in (2.7) can be relaxed.

The objectives we consider are expressed in Equation (2.11) for minimizing the Overall cost on LTEs, and Equation (2.12) for the Min-Max objective. Studies on objective (2.13) can be found in [23] and extended work in [18].

Chapter 3

Previous Work

Because of its importance and difficulty, the traffic grooming problem has drawn a lot of attention in recent years. In this section, we will have a survey of some recent related work on traffic grooming from the literature.

In [20], Dutta and Rouskas gave a survey and classification of relevant work on the *virtual topology design* problem. Some surveys on the general *traffic grooming* problem can be found in [22, 47, 14, 60]. Note that there are different assumptions and objectives for the traffic grooming problem, for instance, our study concentrates on the *static grooming* version, while Mukerjee *et al.*'s work [60] is mostly about *dynamic grooming.* This survey is primarily in the static grooming range with topics related to our research.

3.1 Routing and Wavelength Assignment Algorithms

There have been many studies on the RWA problem alone. Since lightpath routing and wavelength is an important sub-problem in the entire traffic grooming problem, it is important to review what has been done in the literature. The RWA methods generally fall into the following categories:

- Conversion to Graph Models. In Chlamtac *et al.*'s work [38], the RWA problem is transformed into a corresponding graph, then graph heuristics can be applied to it. In the layered graph, network nodes are duplicated and edges are created to express the relationships and constraints between connections, route paths and device/capacity constraints. In the much more complicated new graph, the original lightpath routing and wavelength assignment problem is transformed to a set of shortest path routing problems.

- ILP Relaxations, Simulated Annealing. This category of approaches relaxes the integer constraints of the ILPs, and then get approximate solutions, or uses simulated annealing in the model. Examples of studies in this direction can be found in [40, 41].
- Flow-Deviation-Like. Some works also apply flow deviation methods in some way to the RWA problem. For instance, Konda and Chow [44] first use shortest-path routing on the lightpaths regardless of wavelength capacity, to get an imbalanced initial solution, then try to reroute the lightpaths that 'surplus' the capacities to balance the link loads.
- Iterative Shortest-path with Lowest Available Wavalength. There are many variations of this approach. Basically, a greedy approach is used to accommodate as many shortest-path routes as possible, while for each wavelength, remaining capacities can be used to accommodate more lightpaths if possible. Ding and Hamdi [16] apply a *K-shortest path* method for finding alternative routes, while Sirega *et al.* [37] give two efficient algorithms in recent study that have good performance and fast running time.

As we can see, the RWA sub-problem has been studied extensively, so if we only need to apply RWA at some step of traffic grooming in general topology networks, we can simply adopt one of the methods that have been well explored and shown to be effective. In Chapter 6, we will discuss in detail and experiment with one of the algorithms in [37] for our hierarchical grooming approach.

3.2 Traffic Grooming in Rings

Previous research has concentrated on the ring topology, because of the simple topology and widespread use of SONET rings, which provide simple protection and recovery methods. WDM technology further introduces traffic grooming to the next generation SONET. Specifically, study of grooming on rings can be found in [6, 21, 27, 28, 46, 53, 57].

In [58] and [29], the authors Zhang and Qiao use the idea of forming *circles* of unit traffic demands and grooming them into wavelengths to reduce the number of LTEs in the whole network. Since the objective considered in this study corresponds to the aggregate objective we consider for ring networks, we use the results in [58] to compare with our study of ring topology in various sections of Chapter 4.

For the Min-Max objective we consider on rings, a similar approach of minimizing the maximum nodal cost on ring networks is taken by Chen and Modiano [11] in a different context, namely for routing and wavelength assignment in the presence of converters. Specifically, they develop an algorithm for distributing a number of converters uniformly across the ring nodes rather than placing them at a single hub node. We make use of this *converter distribution* algorithm in the heuristic algorithm we develop in Section 4.3.1

The work in [21] gives heuristic algorithms for ring networks with the objective of minimizing the total amount of electronic switching. Later in [18], proofs were given on the NP-Completeness of the grooming problem in path and ring networks. Some proofs in this thesis were inspired by this earlier work.

In [50], the author of the thesis, Koundinya, considers the objective of minimizing the maximum OEO device at each network node, which is one of the objectives considered in our study, too. Specifically, a decomposition approach is used to partition the ring network into several smaller rings, and solve them respectively with algorithms for path networks. Then, the solutions are re-assembled to get a 'translucent' result for the original problem.

The work of Iyer [39] studies the problem of traffic grooming in simple path networks with egress traffic only. The author proves that for a variety of cost measures, the grooming problem remains NP-Complete even in this further simplified situation. Note that this is a special case in path networks, which implies that the more general path case with arbitrary traffic demands is also NP-Complete.

3.3 Traffic Grooming in Stars

The grooming problem in the star topology has also gained interest recently. The objective considered in relevant studies is to minimize the total amount of electronic switching (and, thus, the delay introduced by OEO transformation), which is related to, but different from, the objective we consider in this study.

In [13], Choi, Garg and Choi first prove that the problem is equivalent to a Maximal Weighted Local Constraint Subgraph (MWLCS) problem, which is NP-Complete. They also describe a greedy algorithm that guarantees a solution whose cost is at most twice the optimal.

In [3], Angel, Bampis and Pascual consider two versions of the problem: minimizing the electronic switching and maximizing the optical switching. The results of the two objectives are equivalent. Besides proving NP-Completeness for both cases, they also prove an important result regarding approximation of this problem. They have shown that an approximation algorithm for either version cannot act as an approximation for the other. They also provided approximation algorithms for both versions separately, by transforming the corresponding versions of the problem to existing NP-Complete problems. A polynomial-time optimization algorithm is also given for the special case in which only two wavelengths are available on each fiber.

In earlier work [18], Dutta, Huang and Rouskas study the grooming problem in stars, among other elemental network topologies. For star networks, the paper gives complexity proofs, and by pruning the search tree, finds a method that can give a series of upper and lower bounds. A greedy heuristic is provided to make improvement towards the objective at each iteration.

We point out here that some of the complexity proofs in this thesis are inspired by the proofs in [18], though the objectives we consider are different from that in the paper.

3.4 Grooming in General Topologies

Recent research concentrates more on general topology traffic grooming. Some work for objectives other than ours can be found in [19, 44, 59, 17]. For the objective of minimizing electronic port (LTE) cost, there are also a few studies in the literature.

In [40], Hu and Leida take an ILP relaxation and decomposition approach. They first decompose the grooming problem into two parts, *GR* and *WA*, that is, Grooming+Routing and Wavelength Assignment. For the GR sub-problem, they use the ILP and relax some integer constraints to get fast solutions; for the WA sub-problem, any existing heuristic algorithm from previous studies could be applied. The authors then provide some sufficient conditions under which the decomposition method can give an optimal solution.

In [41], an ILP is formulated, and a Lagrangian-based heuristic is proposed. The authors Patrocínio and Mateus give a **Layered Graph Representation** for the grooming problem according to the underlying physical topology, and an *allocation cost* is associated with each arc. They first get lower bounds with Lagrangian relaxation, then calculate some upper bounds from the relaxed ILP. After that, a Subgradient Search Procedure is used to improve the solutions.

Ding and Hamdi's work [16] aim at minimizing the number of transceivers as well as the number of wavelengths in Mesh networks. They describe a heuristic using the *Blocking Island* (BI) paradigm. BI is an efficient way of abstracting resources (bandwidth) available in a communication network. The idea is developed from Artificial Intelligence research, namely constraint satisfaction and abstraction with phase transition. A recursive decomposition of BI graphs in decreasing order is used for the heuristic algorithm.

The authors of [45] extend the *Genetic Algorithm* (GA) to create a new model with heuristic approach to support network cost optimization for combining multiple traffic streams into a single lightpath. Lee and Park use the Bin Packing approach as framework, and the cost function is $\min(N_{tr} + \alpha N_{lp})$, with rate α to specify the balance between electronic (transceiver) and optical (wavelength) cost. The definition of 'traffic grooming' is different from the most popular one in the literature, so the work actually addresses only a special version of the problem.

A case for applying a clustering approach in the design of WDM networks that interconnect SONET rings is considered by Esfandiari *et al.* [25], but the authors concentrate on designing lightpath connections of multiple transmission rates, and with non-uniform line terminating equipments of different capability.

In Chapter 6, we design a hierarchical grooming approach for the traffic grooming in general topology networks, which also has a clustering phase. Since clustering is common in large network design problems, we will discuss some related work in Chapter 7. As the survey and subsequent experiments show, the complex nature of the traffic grooming problem requires clustering methods more complex than the existing clustering algorithms can provide. Since the survey on clustering is more related to the work we consider later, we choose to put it together with the study in Chapter 7.

Chapter 4

Grooming in the Ring Topology

In this chapter, we study the traffic grooming problem in the ring topology.

We first give the ILP formulation for the grooming problem specifically for rings, and then give complexity proofs for various cases of the path and ring problems. After that, a polynomial-time algorithm for grooming in unidirectional rings with the Min-Max objective is given. We further adapt the algorithm to cope with the problem in bidirectional rings, and the objective of minimizing the overall cost as well. Finally, numerical results are given to show the effectiveness of our algorithm in a wide range of cases.

4.1 ILP Formulation for the Ring Topology

Although an ILP formulation for the grooming problem in general topologies is given in Section 2.3, we want to show the ILP specifically for the ring topology, because in rings, traffic and lightpath routing are fixed in one way for the unidirectional case, and have only two choices (clockwise or counterclockwise) for the bidirectional case. This will allow us to simplify the general formulation for the ring case, allowing us to get exact solutions for larger instances of rings than is possible using the general formulation.

The formulation also helps us with the NP-complete analysis in the next section. Specifically, it shows that even if more constraints on wavelength assignment are removed from the ILP to make a path topology, the problem still remains hard.

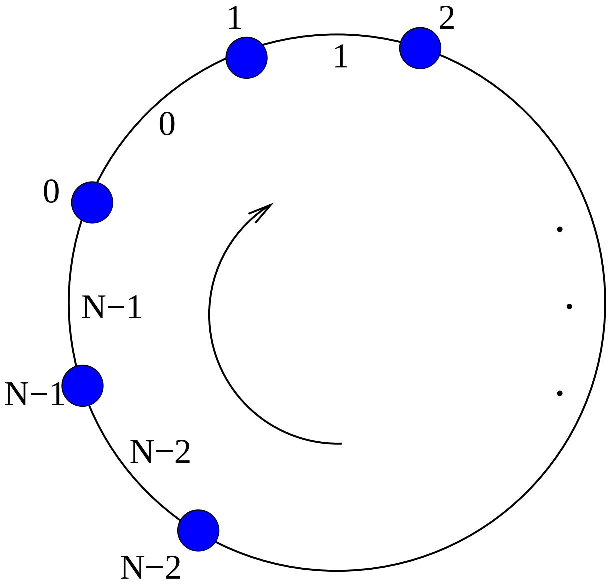


Figure 4.1: An Unidirectional WDM Ring Network with N Nodes

4.1.1 ILP for Unidirectional Rings

We define an unidirectional ring network $\mathcal{R}$ with N nodes that carries traffic in the clockwise direction; in other words, data flows from a node i to the next node $i \oplus 1$ on the ring, where $\oplus$ denotes addition modulo-N. The links of $\mathcal{R}$ are numbered from 0 to $N-1$, such that the link from node i to node $i \oplus 1$ is numbered i. We let $t(l)$ denote the aggregate traffic load on the physical link l (from node l to node $l \oplus 1$) of the ring. The value of $t(l)$ can be easily computed from the traffic matrix T. The component of the traffic load $t(l)$ due to the traffic from source node s to destination node d is denoted by $t^{(sd)}(l)$. If one or more lightpaths exist from node i to node j in the virtual topology, the traffic carried by those lightpaths is denoted by t_{ij}. The component of this load due to traffic from source node s to destination node d is denoted by $t_{ij}^{(sd)}$. In our formulation we allow for multiple lightpaths with the same source and destination nodes. We denote the *lightpath count* from node i to node j by b_{ij}, taking its value from $\{0, 1, 2, \cdots, W\}$. We also define the *potential lightpath set for a link* to be the set of lightpaths that would pass through a given link, and denote it by $B(l) = \{\ (i,j) \mid$ lightpath (i,j), if it existed, would pass through link $l\ \}$. Finally, we let $c_{ij}^{(k)}$ be the *lightpath wavelength indicator*, i.e., $c_{ij}^{(k)}$ is 1 if a lightpath from node i to node j uses wavelength k, 0 otherwise.

Using the above definitions, we can now formulate the problem of designing a virtual topology for an unidirectional ring network with the stated objectives. The following formulation is an integer linear problem (ILP) consisting of $O(N^4 + N^2W)$ constraints and $O(N^4 + N^2W)$ variables, where N is the number of nodes in the ring, and W is the number of wavelengths.

Given:

The physical topology, a unidirectional ring $\mathcal{R}$ of N nodes.

The traffic matrix $T = [t^{(sd)}], s, d \in \{0, \cdots, (N-1)\}$, $t^{(sd)} \in \{0, 1, 2, \cdots\}, t^{(ss)} = 0, \forall s.$

The wavelength limit W, which is the number of distinct wavelengths each link can carry, and **wavelength capacity** C, the number of unit traffic rates each wavelength can carry.

Find:

Complete description of grooming for all the traffic in terms of lightpath indicators b_{ij}, lightpath wavelength indicators $c_{ij}^{(k)}$, and traffic routing variables $t_{ij}^{(sd)}$.

Subject to:

Traffic Constraints

$$t_{ij}^{(sd)} \leq t^{(sd)}(i), \quad \forall (i,j), (s,d) \tag{4.1}$$

$$t_{ij}^{(sd)} \in \{0, 1, 2, \cdots\}, \quad \forall (i,j) \tag{4.2}$$

$$\sum_{(i,j) \in B(l)} t_{ij}^{(sd)} = t^{(sd)}(l), \quad \forall (s,d), l \tag{4.3}$$

$$t_{ij} = \textstyle\sum_{sd} t_{ij}^{(sd)}, \quad \forall (i,j) \tag{4.4}$$

$$t_{ij} \leq b_{ij} C, \quad \forall (i,j) \tag{4.5}$$

$$\sum_j t_{ij}^{(sd)} - \sum_j t_{ji}^{(sd)} = \left\{ \begin{array}{ll} t^{(sd)}, & s = i \\ -t^{(sd)}, & d = i \\ 0, & s \neq i, d \neq i \end{array} \right\} \forall i, (s,d) \tag{4.6}$$

Wavelength Constraints

$$\sum_{(i,j) \in B(l)} b_{ij} \leq W, \quad \forall l \tag{4.7}$$

$$\sum_{k=1}^{W} c_{ij}^{(k)} = b_{ij}, \quad \forall (i,j) \tag{4.8}$$

$$\sum_{(i,j)\in B(l)} c_{ij}^{(k)} \quad \leq \quad 1, \quad \forall l, k \tag{4.9}$$

To minimize:

The Maximum in/out Degree at Each Node

$$F \;=\; \max_i \left(\max \left(\sum_j b_{ji}, \sum_j b_{ij} \right) \right) \tag{4.10}$$

or: The Overall Degree in the Virtual Topology

$$F' = \sum_{i,j} b_{ij} \tag{4.11}$$

The traffic constraint (4.1) ensures that a lightpath can carry traffic for a source-destination node pair only if it is in the physical route of the traffic component.

Constraint (4.3) states that the physical traffic on a link due to a source-destination node pair must be equal to the sum of the traffic on all lightpaths passing through that link due to that node pair.

Constraints (4.4) and (4.5) define the total traffic on a lightpath and relate it to the lightpath count, respectively.

Because of the definition of the quantities $t^{(sd)}(l)$, constraints (4.1) and (4.3) together ensure that no traffic component can be routed completely around the ring before being delivered at the destination node. Constraint (4.6) is an expression of traffic flow conservation at lightpath endpoints.

Among the wavelength constraints, constraint (4.7) expresses the bound imposed by the number of wavelengths available, (4.8) relates the wavelength indicators to the lightpath counts, and (4.9) ensures that no wavelength clash can occur.

Finally, the objective function (4.10) is the maximum nodal degree in the ring network, and (4.11) is the overall degree in the whole network system.

4.1.2 ILP for Bidirectional Rings

The ILP formulation for Bidirectional Rings is more complicated, in that routing can be done in both directions. However, the format is very similar to the ILP for unidirectional rings we have described in the previous section. Instead of repeating the complicated formulations, we make alterations on it to get the new ILP for the bidirectional case.

We make the following changes to the variables: replace each variable b_{ij} with two variables, br_{ij} and bl_{ij}, representing the clockwise and counterclockwise routing of lightpath (i,j) respectively; similarly, replace $c_{ij}^{(k)}$ with $cr_{ij}^{(k)}$ and $cl_{ij}^{(k)}$; calculate the potential lightpath set $B(l)$ according to the respective routing direction, resulting in sets $Br(l)$ and $Bl(l)$ respectively.

For the wavelength constraints, calculate the constraints for each direction separately, which means that we have a total of 3 more Wavelength Assignment SP Constraints.

Furthermore, for the variable $t_{ij}^{(sd)}$, more combinations of (i,j,s,d) are allowed to have nonzero values than in the unidirectional case. This is done by revising Constraint (4.3) into the following:

$$\sum_{(i,j)\in Br(l)\cup Bl(l)} t_{ij}^{(sd)} = t^{(sd)}(l), \forall (s,d), l \tag{4.12}$$

We also need to further define some restrictions in bidirectional routing, so that one lightpath won't occupy both directions of any link. A brief discussion is given in Section 4.4.

4.2 Complexity Results for Paths and Rings

The traffic grooming problem in ring networks is NP-Complete since the RWA sub-problem in rings is NP-Complete (see Section 2.1). In this section, we prove that the traffic grooming problem in path networks is also NP-Complete. Since the RWA problem can be solved in linear time for path networks, our results demonstrate that traffic grooming with the objectives we consider is itself an inherently difficult problem. They also prove that the traffic grooming problem with the objectives remains NP-Complete in rings or other general topologies *even when* full wavelength conversion is available at the network nodes.

We extend our results to show that for the respective bi-directional cases, our results also hold. Moreover, we prove that for one of the grooming cases we consider, an additive constant approximation algorithm cannot be found unless $P = NP$.

4.2.1 NP-Completeness for the Non-bifurcated Case in Paths

Let us consider a network in the form of a unidirectional path $\mathcal{P}$ with N nodes. There is a single directed fiber link from node i to node $i+1$, for each $i \in \{1, 2, \cdots, N-1\}$.

An instance of the traffic grooming problem is provided by specifying a number N of nodes in the path, a traffic matrix $T = [t^{(sd)}], 1 \leq s < d \leq N$, a grooming factor C, a number of wavelengths W, and a goal F. The problem asks whether a valid logical topology may be formed on the path and all traffic in T routed over the lightpaths of the logical topology so that the number of incoming or outgoing lightpaths at any node in the path is less than or equal to F.

Here we consider the case where bifurcated routing of traffic is not allowed. Specifically, for any source-destination pair (s, d) such that $t^{(sd)} \leq C$, we require that all $t^{(sd)}$ traffic units be carried on the *same* sequence of lightpaths from source s to destination d. On the other hand, if $t^{(sd)} > C$, it is not possible to carry all the traffic on the same lightpath. In this case, we allow the traffic demand to be split into $\lfloor \frac{t^{(sd)}}{C} \rfloor$ components of magnitude C and at most one component of magnitude less than C, and the no-bifurcation requirement applies to each component independently.

Theorem 4.1 *The decision version of the grooming problem in unidirectional paths with the Min-Max objective (bifurcated routing of traffic not allowed) is NP-complete.*

Proof. The reduction is from the Subset Sum problem [26]. An instance of the Subset Sum problem consists of n elements of size $w_i \in Z^+, \forall i \in \{1, 2, \cdots, n\}$, and a goal B. The question is whether there exists a subset of elements whose sizes total B. Let $B_1 = \max\{B, \sum_i w_i - B\}$. Construct a path network using the following transformation: $N = 2n + 3$, $W = 3$, $C = \sum_i w_i + 1$, and let the objective be $F = 2$. The $2n + 3$ nodes of the path are labeled $S, 1, 2, \cdots, 2n + 1, D$, and the traffic matrix is:

$$t^{(sd)} = \begin{cases} C + 1, & s \in \{1, 2, \cdots, n-1\} \\ & \cup\{n+2, n+3, \cdots, 2n\}, d = s + 1; \\ B_1 + 1, & s = n, d = n + 1; \text{or } s = n + 1, d = n + 2; \\ C - B_1, & s = n, d = n + 2; \\ w_s, & s \in \{1, 2, \cdots, n\}, d = s + (n + 1); \\ C, & s = S, d = n + 1; \text{or } s = n + 1, d = D; \\ C, & s = S, d = 1; \text{or } s = 2n + 1, d = D \\ 0, & \text{otherwise} \end{cases}$$

The traffic matrix is set up so that we are forced to use the virtual topology shown in Figure 4.2. Specifically, the first of the three wavelengths is used to form two lightpaths,

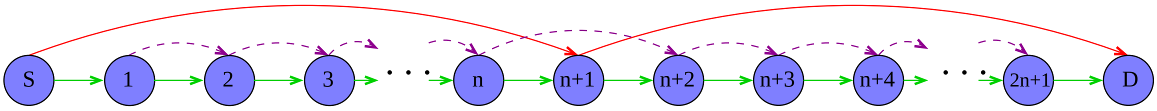


Figure 4.2: Example of path construction for the proof of Theorem 4.1, $N = 2n+3$, $W = 3$

one from node S to node $n+1$, and one from node $n+1$ to node D; both these lightpaths are filled to capacity carrying the corresponding traffic demands of magnitude C. Therefore, this first wavelength cannot be used to carry any other traffic. The second wavelength is used to form single-hop lightpaths between adjacent pairs of nodes in the path; these lightpaths are denoted by the straight arrows in Figure 4.2.

However, note that the traffic demands in the top row of the traffic matrix above are greater than the capacity C of a wavelength. Therefore, the third wavelength is used to form single-hop lightpaths between nodes $(i, i+1), i = 1, \cdots, n-1, n+2, \cdots, 2n$, in the path. The single-hop lightpaths on this third wavelength between any such pair $(i, i+1)$ of nodes carry the following traffic components: (1) the one unit of traffic from node i to node $i+1$ (the other C units of such traffic are carried by the single-hop lightpath on the second wavelength), and (2) the w_j units of traffic from node $j, j = 1, \cdots, i$, to node $j+n+1$. Since $C = \sum_i w_i + 1$, the single-hop lightpath on the third wavelength from node i to $i+1$ has enough capacity to carry this traffic, for all such i. Finally, the third wavelength is also used to form a direct lightpath from node n to node $n+2$ that bypasses node $n+1$, and the result is the virtual topology in Figure 4.2. Note that no more lightpaths can be added to this virtual topology, and no existing lightpaths can be split without violating the Min-Max objective F.

After grooming all the traffic except for w_i, which have to go through node $n+1$ with or without stopping, the capacity left in the lightpath from node n to node $n+2$ is equal to B_1, and the capacities left in the lightpaths from node n to node $n+1$ and from node $n+1$ to node $n+2$ are both equal to $\sum_i w_i - B_1$. Since bifurcation is *not* allowed, it is possible to use the virtual topology above to groom the traffic, *if and only if* there is a subset of $\{t^{(i,i+n+1)}\}$ whose sum is exactly B_1. Since the decision version of the Subset Sum problem is NP-Complete, then the new grooming problem is also NP-Complete. ■

Because of the construction in the above proof, we have the following corollary. This corollary demonstrates that, even when solutions to the first two sub-problems of the traffic grooming problem (refer to Section 2.2) are provided, the entire problem remains NP-Complete by virtue of the third sub-problem (traffic routing). Therefore, traffic grooming is inherently more difficult than the well-known NP-Complete RWA problem.

Corollary 4.1 *The decision version of the traffic grooming problem in unidirectional paths with the Min-Max objective (bifurcated routing of traffic not allowed) is NP-Complete even when a logical topology is provided.*

Now we consider the objective F', which is to minimize the overall degrees (in-degree + out-degree). We use the same construction as in Theorem 4.1, but make the overall objective $F' = 8n + 6$, which is exactly the sum of in/out degrees in Figure 4.2. Since the same assertion can be made that no more lightpaths can be added, and no existing lightpaths can be split without violating the objective, we are again forced to use the existing virtual topology. We then have the following corollary as well:

Corollary 4.2 *The decision version of the traffic grooming problem in unidirectional paths with the Overall objective (bifurcated routing of traffic not allowed) is NP-Complete, even when a logical topology is provided.*

Please note that the Subset Sum problem is not *strongly* NP-Complete, which implies that if all w_i's are not large, there exists pseudo polynomial-time algorithm to solve the Subset Sum problem. However, we cannot conclude from what we know that the path grooming problem is also not strongly NP-Complete. In fact, we map each Subset Sum instances to only a set of special instances in path networks, and with a potential of $O(N^2)$ traffic demands for all node pairs, the restriction on the size of some specific traffic demands does not have much practical values. For this reason, we do not study the strong NP-Completeness nature of the problems. The same reason holds for subsequent proofs on rings, as well as on star networks in the next chapter.

4.2.2 NP-Completeness for the Bifurcated Case in Paths

We now extend the results to the case where bifurcated routing of traffic is allowed. Specifically, a traffic component $t^{(sd)}$ is allowed to be split into various parts which may follow different routes (i.e., different lightpath sequences for a path network) from source to destination. The bifurcation is restricted to integer values.

Theorem 4.2 *The decision version of the grooming problem in unidirectional paths with the Min-Max objective (bifurcated routing of traffic allowed) is NP-complete.*

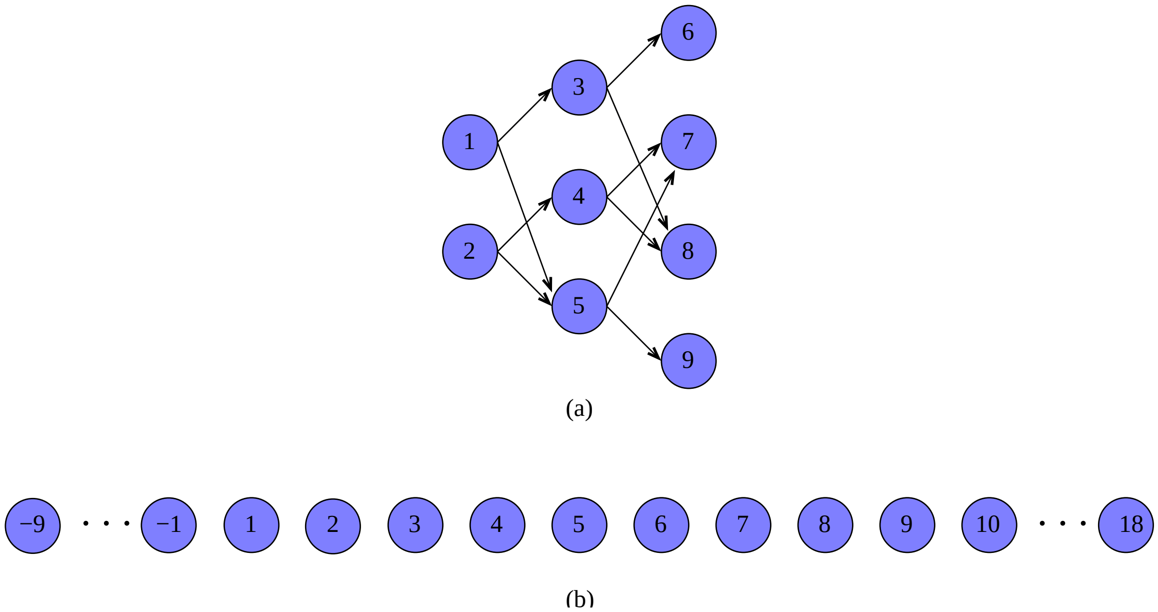
Proof. The reduction is from the constrained Multicommodity Flow (MCF) problem in three-stage networks with three nodes in the second stage, which is NP-Complete [24]. An instance of the restricted MCF problem has N_1, N_2, and N_3 as the sets of nodes forming the 3 stages, with $|N_1 \cup N_2 \cup N_3| = n$, and $|N_2| = 3$ (refer to Figure 4.3(a)). $E \subset (N_1 \times N_2) \cup (N_2 \times N_3)$ is the set of edges in the network, each of unit capacity, and $Q \subset (N_1 \times N_3)$ is the set of traffic demands, each also of unit magnitude. The problem is whether a feasible flow assignment satisfying the flow constraints exists.

To transform an instance of the MCF problem to an instance of the path grooming problem, we construct a path network with $N = 3 \times n$ nodes, labeled $-n, \cdots, -1, 1, \cdots, n, n+1, \cdots, 2n$ (see Figure 4.3(b)). Let I_j and O_j denote the indegree and outdegree, respectively, of node j of the three-stage network; for example $I_6 = 1$ and $O_6 = 0$ for the network of Figure 4.3(a). Our decision goal is set to $F = \max_j\{I_j, O_j\}$; for the instance of Figure 4.3, $F = 2$. We let the wavelength capacity be $C = 2$, and we do not impose any constraint on the number W of wavelengths; in other words, W can be as large as needed (for practical purposes, we can let $W = KN^2$, where K is an appropriate constant, so that the number of wavelengths is sufficient to set up direct lightpaths between any pair of nodes in the path).

The traffic matrix $t^{(sd)}$ for the path grooming instance is:

$$t^{(sd)} = \begin{cases} 1, & (s,d) \in E \cup Q; \\ C \times (F - I_d), & \forall d = 1, 2, \cdots, n, s = d - (n+1); \\ C \times (F - O_s), & \forall s = 1, 2, \cdots, n, d = s + n. \end{cases}$$

Because of the traffic demands of size $C(= 2)$, a direct lightpath needs to be formed to carry each such demand in order not to exceed the goal F; the resulting logical topology for the example of Figure 4.3(a) is shown in Figure 4.4(a). Recall that $I_6 = 1$ and $O_6 = 0$ in the three-stage network of Figure 4.3. Therefore, the above traffic matrix specifies $t^{(-4,6)} = C = 2$ and $t^{(6,15)} = 2C = 4$. As we can see in the figure, there is one lightpath from node -4 to node 6, and this lightpath carries the total demand $t^{(-4,6)}$, while



(a)

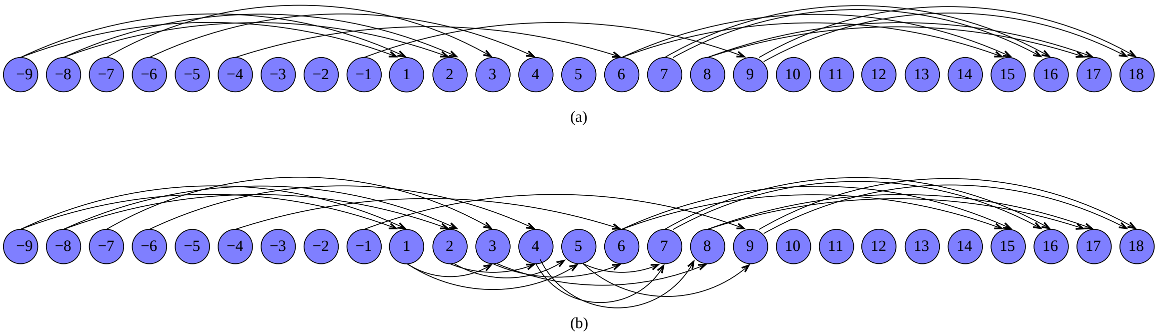
(b)

Figure 4.3: Example of path construction for the proof of Theorem 4.2: (a) the original MCF instance, and (b) the corresponding path network

there are two lightpaths from node 6 to node 15, each of them carrying C units of the traffic demand $t^{(6,15)}$.

Because of the lightpaths already set up, the only way to satisfy the unit traffic demands between node pairs in set E in the above traffic matrix, is to form a direct lightpath for each such traffic demand. The resulting logical topology is shown in Figure 4.4(b). For example, consider the source-destination pair $(1, 3) \in E$ (refer to Figure 4.3(a)). As we can see, a direct lightpath has been formed between nodes 1 and 3 in Figure 4.4(b); similarly for other node pairs in set E.

We now see that the logical topology in Figure 4.4(b) is such that exactly two lightpaths originate from, and terminate at, each node $i = 1, 2, \cdots, n$, of the path. Therefore, it is not possible to add any new lightpath with such a node as the origin or termination point without violating our stated goal $F = 2$. Consequently, we have to use the remaining capacities left on the lightpaths formed between node pairs in set E in order to carry (groom) the demands in the above traffic matrix due to node pairs in set Q. This can be done *if and only if* the original constrained MCF problem has a solution. Since the MCF problem is known to be NP-Complete, so is the path grooming problem with bifurcation of traffic allowed. ■



(a)

(b)

Figure 4.4: The logical topology for the proof of Theorem 4.2: (a) the topology after forming lightpaths for each demand of size C, and (b) the topology after forming additional lightpaths for each demand between node pairs in set E

Again, by the nature of the proof, we are able to state the following:

Corollary 4.3 *The decision version of the traffic grooming problem in unidirectional paths with the Min-Max objective (bifurcated routing of traffic allowed) is NP-Complete even when a candidate logical topology is provided.*

Similarly, we can prove the following corollary as in the non-bifurcated case for the Overall objective. This time we need only $N = n$ nodes in the path network, corresponding to the nodes $1, \ldots, n$ in the above construction. The objective overall degree $F' = 2|E|$. This will force us to use one lightpath between each node pair in set E, without adding new lightpaths or splitting existing ones. By reduction from the constrained MCF problem, we have:

Corollary 4.4 *The decision version of the traffic grooming problem in unidirectional paths with the Overall objective (bifurcated routing of traffic allowed) is NP-Complete, even when a candidate logical topology is provided.*

Note that since the Multicommodity Flow problem is NP-Complete in the strong sense, it implies strong NP-Completeness for the bifurcated case in the original problem as well as in the corollaries.

Now we have proved that the grooming problem in unidirectional paths is NP-complete no matter whether bifurcated traffic routing is allowed or not, and no matter if we consider the Min-Max or the Overall objective. We will then concentrate on the *bifurcated* case in our study, because it may generate lower min-max degree than the non-bifurcated case, and some security issues will favor bifurcation of traffic as well.

4.2.3 Approximability for the Bifurcated Case in Paths

A variant of the proof of Theorem 4.2 shows that the grooming problem in unidirectional paths, when bifurcated traffic routing is allowed, is not approximable for any given constant absolute value in polynomial time, unless P=NP. The theorem is stated as follows:

Theorem 4.3 *If P≠NP, then no polynomial time approximation algorithm A for the Traffic Grooming problem in unidirectional path networks with the Min-Max objective (bifurcated routing allowed) can guarantee* $A(Instance) - OPT(Instance) \leq K$ *for a fixed constant integer* K*.*

Proof. The reduction is from the same Multi-Commodity Flow (MCF) problem used in the proof of Theorem 4.2. Recall that in the restricted MCF problem, we have node sets N_1, N_2, and N_3, such that $|N_2| = 3$. For a given MCF problem instance, let P denote the corresponding instance of the unidirectional path problem constructed in the proof of Theorem 4.2. For this proof, we construct a path instance P' and objective F' by modifying the instance P and objective F of the previous proof as follows.

Case (a). A node of N_2 has the maximum indegree or outdegree among all in- and out- degrees. We make K duplicates of each node in N_1 and N_3, and connect them with N_2 in exactly the same way as the original sets do. Therefore, we will have node sets $N_1^1, N_1^2, \cdots, N_1^K$ as duplicates of N_1, and similarly, K duplicates of N_3. If a node m in N_1 is connected to a node set $N_2' \subseteq N_2$, we will make the corresponding nodes $m^1, \cdots, m^K$ connect to the same node set in N_2. We do the same for the new sets $N_3^1, N_3^2, \cdots, N_3^K$. We also duplicate the traffic demands between N_2 and other nodes accordingly. Moreover, if there is traffic demand from $s \in N_1$ to $d \in N_3$, we add traffic demand from $s^i \in N_1^i$ to

$d^i \in N_3^i$, $i = 1, \cdots, K$, as well. After doing this, we replace each node in N_1 by $K+1$ copies of itself in the original MCF problem, except that now the new goal $F' = (K+1)F$. Keeping $C = 2$ and using the same way to map the MCF problem to the unidirectional path, which will have $N = 3 \times (K|N_1 + N_3| + |N_2|)$ nodes, we get a new grooming problem P' with a polynomial-time transformation (since K is a constant).

Now suppose there exists a polynomial-time algorithm A that can guarantee $A(P') - OPT(P') \leq K$. we solve P' using this algorithm. From the result, we can easily decide if $OPT(P) \leq F$, because of the following:

1. $OPT(P') = (K+1)OPT(P)$, so $OPT(P')$ must be a multiple of $K+1$.
2. $A(P') - OPT(P') \leq K$, so $OPT(P') = Y$, where Y is the largest integer less than $A(P')$ that is a multiple of $K+1$.
3. Since $OPT(P) = OPT(P')/(K+1)$, we can immediately compute $OPT(P)$ and decide if $OPT(P) \leq F$.

Consequently, for any instance in the MCF problem, we can construct an instance P' in polynomial time, and use the method above to decide if $OPT(P) \leq F$ in polynomial time, then further decide if the MCF problem has a solution, i.e., solve the decision version of the MCF problem in polynomial time. This is only possible if P=NP.

Case (b). If F exists in N_1 or N_3, we can make K duplicates of the node set N_2, that is, add node sets $N_2^1, \cdots, N_2^K$, each with 3 nodes. Then duplicate the traffic demands accordingly, and have the new goal $F' = (K+1)F$ as in Case (a). Using a similar proof technique, we can get the same result.

Case (c). If F exists in both N_2 and N_1 or N_3, use either Case (a) or Case (b) for the proof. ■

A similar proof can be done to show that the same conclusion can be made for the Overall objective as well. By combining the methods in Corollary 4.4 and making duplicates in the proof of Theorem 4.3, we get the following corollary:

Corollary 4.5 *If P≠NP, then no polynomial time approximation algorithm A for the Traffic Grooming problem in unidirectional path networks with the Overall objective (bifurcated* *routing allowed) can guarantee $A(Instance) - OPT(Instance) \leq K$ for a fixed constant integer K.*

4.2.4 Extending the Results to the Bidirectional and Ring Cases

Let us now consider the implications of these results for related topologies, *bidirectional path* networks, and *ring* networks (unidirectional or bidirectional). For the bidirectional case, we don't differentiate lightpaths according to their directions, but only their contributions to the in/out degrees. This is because the LTE cost at each node is dependent on the in/out lightpaths, regardless of their directions. The implications for ring networks are of practical importance, even though the NP-hard nature of traffic grooming for ring networks has already been demonstrated. In particular, it is known that the RWA problem in rings is NP-hard [55]. However, the following corollaries show that *even if all ring nodes are equipped with wavelength converters* (in which case wavelength assignment is trivial), traffic grooming with the Min-Max objective remains a difficult problem. We state below three corollaries that settle the question for these topologies.

Corollary 4.6 *The decision version of the grooming problem in bidirectional paths with the Min-Max or the Overall objective (bifurcated routing of traffic allowed or not allowed) is NP-complete.*

Proof. Based on our respective proof construction for *unidirectional* paths, we can add traffic demands between adjacent nodes in the opposite direction, each demand equal to the full capacity of the link in that direction, and add W to our Min-Max objective, that is, $F^1 = F + W$. A traffic component is allowed to be carried from source node s to destination node d on a sequence of lightpaths some of which are in one direction and some in the reverse direction; thus a traffic component may traverse the same link multiple times in either direction. It is clear, however, that a traffic component must traverse the outgoing links from node s in the direction in which node d lies at least once, and the incoming links to node d from the direction in which node s lies at least once. Since the traffic demands we added for the opposite direction have reached the capacities for the links in that direction, the above case will become infeasible without shortest-path routing.

Similarly, $F^2 = F' + (n-1) \times W$ will lead to the same NP-Complete results for the Overall objective. ■

Corollary 4.7 *The decision version of the grooming problem in unidirectional rings with the Min-Max or the Overall objective (bifurcated routing of traffic allowed or not allowed) is NP-complete, even when every node has full wavelength conversion capability.*

Proof. For an instance of the unidirectional path problem, if we merge the two nodes at both ends into one node, it will form a unidirectional ring. Thus, whichever problems we use for the NP-Completeness proof to reduce from in the path topology, we can reduce it into a unidirectional ring problem as well. ■

Corollary 4.8 *The decision version of the grooming problem in bidirectional rings with the Min-Max or the Overall objective (bifurcated routing of traffic allowed or not allowed) is NP-complete, even when every node has full wavelength conversion capability.*

Proof. Like in Corollary 4.6, we can add full traffic demands between adjacent nodes in the opposite direction of the ring, then adjust the corresponding goal values, and make similar assertions. ■

The non-approximability results we get in Section 4.2.3 can also be extended in similar ways. To avoid repetition, we will not include the results here.

Note that the proofs do not show constant factor approximability. In fact, we can always use only short lightpaths between adjacent nodes to pack and groom all the traffic demands, which will give us a simple W-approximation algorithm. We do not have results for better theoretical approximation rates for the set of problems.

4.3 Algorithm for Traffic Grooming with the Min-Max Objective in Unidirectional Rings

The complexity results presented in the previous section imply that it is very unlikely that a polynomial-time algorithm exists to get the optimal solution for the traffic grooming problem in the ring topology. Therefore, we need to find polynomial-time algorithms that give results close to the optimal.

We now present such an algorithm for unidirectional ring networks with the goal of minimizing the maximum nodal degree (indegree or outdegree) in the logical topology, consistent with the objective function (4.10). As we mentioned earlier, the degree of a node is a reflection of the LTE cost needed at that node. Rather than solving all three sub-problems of the traffic grooming problem simultaneously (refer to Section 2.2), we decouple the logical topology and traffic routing sub-problems from the RWA sub-problem and tackle them independently. Specifically, our algorithm consists of the following steps:

- **Step 1.** Solve the logical topology and traffic routing sub-problems on the ring network using the algorithm in Section 4.3.1. The result of this step is a set R of lightpaths (logical topology) and a routing of the traffic demands over the lightpaths in R that minimize the maximum amount of LTE at any node.

- **Step 2.** Use the algorithm in Section 4.3.2 to color the lightpaths of set R. The result of this step is a wavelength assignment that does not use more than W wavelengths. However, at the end of this step, the amount of LTE at one node, say, node i, of the ring may increase beyond the corresponding value after Step 1, by an amount equal to some value Δ.

- **Step 3.** Use the algorithm presented by Chen and Modiano [11] to distribute the additional Δ LTE at node i to other nodes in the ring network.

The following subsections explain the steps of our Min-Max traffic grooming algorithm in more detail.

4.3.1 Algorithm for Min-Max Grooming without WLA

In this subsection, we give details of a polynomial-time algorithm for the logical topology and traffic routing sub-problems of the traffic grooming problem. Note that unlike previous studies, our algorithm attempts to minimize the maximum amount of LTE at any ring node by creating long lightpaths that bypass intermediate nodes whenever possible. Because of the results given in Section 4.2, the decision version of the traffic routing sub-problem without RWA is itself NP-Complete, and hence our polynomial-time algorithm may terminate without necessarily finding an optimal, or even a feasible solution. However, numerical results to be presented later indicate that the solutions obtained using our algorithm are close to the optimal and/or the lower bound on the objective function (4.10).

Before we proceed, we introduce the concept of *reduction* of a traffic matrix. Specifically, we reduce the matrix T so that all elements are less than the capacity C of a single wavelength, by assigning a whole lightpath to traffic between a given source-destination pair that can fill it up completely. The available wavelengths on the links of the path segment from the source to the destination node are also decremented by the number of lightpaths thus assigned. Since breaking such lightpaths would increase the amount of LTE at some intermediate nodes of the path, this procedure does not preclude us from reaching an optimal solution, nor does it make the problem inherently easier or more difficult. We continue using the same notation for the traffic matrix and traffic components, but in what follows, they stand for the same quantities *after* the reduction process.

After the reduction, we initialize the logical topology to one in which a sufficient number of single-hop lightpaths is formed on each link of the ring network to carry the traffic using this link. We note that this initial solution is a feasible solution to the logical topology and traffic routing sub-problems, in that it does not use more than W lightpaths on any link. However, this initial topology yields a large value for the objective F in expression (4.10), which is equal to the number of single-hop lightpaths in the most congested link. Our approach, then, is to improve on this initial solution by joining short lightpaths to form longer ones, thus lowering the degrees at intermediate nodes. In the following, we summarize our algorithm for joining short lightpaths.

Let us define the relationship $i \prec j$ between ring nodes to denote that node i "precedes" node j in the direction of traffic flow; similarly, we will use the notation $i \preceq j$ to denote that node i precedes, or may be the same as, node j. The main idea of our algorithm is to consider the node with the maximum degree, and to attempt to decrease its degree by one at each iteration; this process repeats until no more improvement is possible. Let m be the node with the maximum degree. The algorithm searches for a pair of nodes $(i, j), i \prec m \prec j$, such that there exist lightpaths (i, m) and (m, j). The objective is to shift all the traffic from lightpaths (i, m) and/or (m, j) to either an existing or a new lightpath (i, j) in order to decrease the maximum of the indegree and outdegree of node m by one. If the traffic can be shifted entirely to an existing lightpath (i, j), then this procedure is always possible, since no wavelength limit constraints are violated, and also the degree of nodes i and/or j may also decrease in the process. However, if a new lightpath (i, j) must be created, the above procedure is carried out only if the wavelength limit constraint (4.7) is not violated and the degrees of nodes i and j do not increase above the current maximum degree minus one (the minus one is necessary to ensure that the algorithm will not get into an infinite loop). A more detailed description of the algorithm is provided in Figure 4.5.

Step 9 decides how much traffic is required to be removed from (i, m) and (m, j), in order to achieve the desired degree change at m. That amount needs to be accumulated as traffic is shift in the loop 11-14. If there is enough traffic that can be shifted, we will actually do the removal at Steps 16-18; otherwise, we try the next pair in the (i, j) loop. Step 15 requires a check of violation of the wavelength constraints and the min-max objective. If $I_m = O_m$, in order to lower the max degrees at m, we always need to take out both (i, m) and (m, j). In this case, the accumulation of 'sum' in Step 8 and 16 needs to be $max(C - r_{im}, C - r_{mj})$. If $I_m > O_m$, various cases need to be considered to ensure the wavelength constraints, and make progress at each iteration. Figure 4.6 shows the cases with examples. For $I_m < O_m$, since the cases are symmetric to $I_m > O_m$, we can also deal with the various cases symmetrically.

Specifically, in Case 1, reducing I_m requires removal of at least $C - r_{im} = 7$ units of traffic from lightpaths im and mj; since the remaining capacity on ij is enough to hold such amount, we can reduce the degree at m without increasing the degrees at i or j. In case 2, to reduce the indegree at m, 7 units need to be removed again. But in this case, a lightpath mj will also be removed and routed on lightpath ij. This time, since both the outdegree of i and indegree of j have been reduced by the removals, even if a new lightpath is required to reroute the traffic on ij, the wavelength constraints will not be violated. The above two cases will not result in new max degrees or wavelength violations.

However, if we encounter Case 3, which does not belong to either Case 1 or 2, and

Min-Max Traffic Grooming Algorithm for Unidirectional Rings
Input: A ring network with N nodes, W wavelengths, capacity C of each wavelength, and traffic matrix $T = [t^{(sd)}]$ (we assume that the traffic matrix has been reduced as explained in the text).
Output: The number of lightpaths b_{ij} from node i to node j of the ring, and the traffic routing quantities $t_{ij}^{(sd)}$ (which indicate the amount of the traffic component $t^{(sd)}$ routed over a lightpath from node i to j), so that the solution does not violate the wavelength limit constraints (4.7) and has a small value of the objective F in (4.10).

```
begin
    Initialize the logical topology to one with only single-hop lightpaths
    and initialize all $b_{ij}$ and $t_{ij}^{(sd)}$ accordingly
    for all $i, j$ do $r_{ij} \leftarrow$ capacity unused on the $(i, j)$ direct lightpath in
    current topology (residual capacity)
    for all $j$ do $I_j \leftarrow$ indegree of $j$ in the current topology
    for all $j$ do $O_j \leftarrow$ outdegree of $j$ in the current topology
    repeat                    // Main iteration
        $m \leftarrow$ some node s. t. $\max\{I_m, O_m\}$ is maximum in the ring
        //Our objective is to replace a lightpath $(i, m)$ and/or $(m, j)$ with
        //direct lightpath $(i, j)$ in order to reduce $\max\{I_m, O_m\}$ by one
        for each pair $(i, j)$ such that $i \prec m \prec j$ do
            if $I_m == O_m$ then $TotalToShift \leftarrow \max\{C - r_{im}, C - r_{mj}\}$
            else if $I_m > O_m$ then $TotalToShift \leftarrow C - r_{im}$
            else $I_m < O_m$ then $TotalToShift \leftarrow C - r_{mj}$
            // $TotalToShift$ is the amount of traffic to be shifted to other
            // wavelengths in order to reduce $\max\{I_m, O_m\}$ by one
            $TrafficToShift \leftarrow 0$
            for each pair $(s, d)$, $s \preceq i, j \preceq d$ do
                $TrafficToShift \leftarrow TrafficToShift + \min(t_{im}^{(sd)}, t_{mj}^{(sd)})$
                if $TrafficToShift > TotalToShift$ then break
                // No more $(s, d)$ pairs needed
            endfor              // of the $(s, d)$ loop
            if $TrafficToShift < TotalToShift$ then break
            // It is not possible to replace lightpaths $(i, m)$ or $(m, j)$
            // Continue with the next $(i, j)$ pair
            else if replacing lightpaths $(i, m)$ or $(m, j)$ would violate any
            wavelength limit constraints or would create new maximum
            nodal degrees at $i$ or $j$ then break
            else
                Remove lightpaths $(i, m)$ and/or $(m, j)$
                // $\max\{I_m, O_m\}$ decreases by one
                for all pairs $(s, d)$ contributing to $TrafficToShift$ do
                Reduce $t_{im}^{(sd)}$ and $t_{mj}^{(sd)}$ by the contributing amount
                and add an equal amount to $t_{ij}^{(sd)}$
                // This step reflects the new routing over lightpath $(i, j)$
            endif
        endfor        // of the $(i, j)$ loop
    until no decrease in the max degree at any node is possible
end        // of the algorithm
```

Figure 4.5: Unidirectional ring algorithm for logical topology and traffic routing

When $C = 12, I_m > O_m$, one of the following cases may happen.

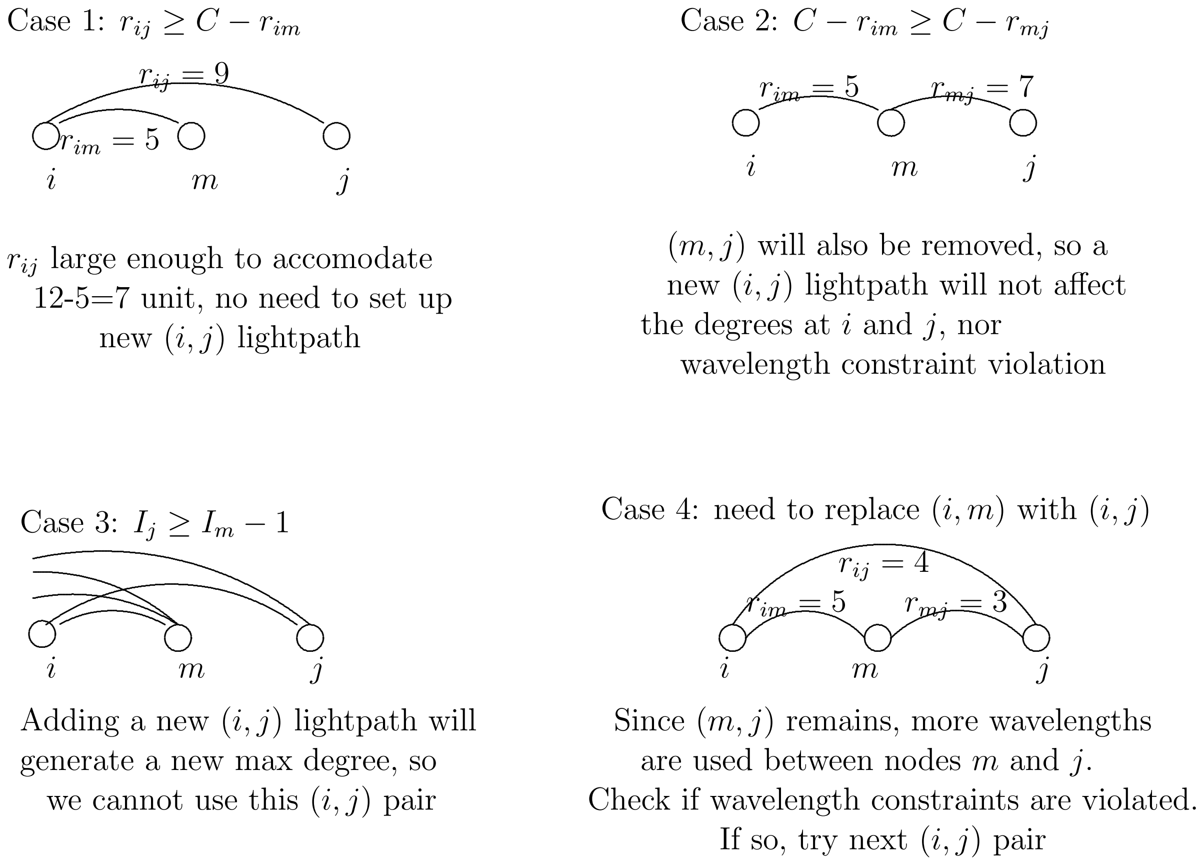

Case 1: $r_{ij} \geq C - r_{im}$
$r_{ij} = 9$
$r_{im} = 5$
i
m
j
r_{ij} large enough to accomodate 12-5=7 unit, no need to set up new (i, j) lightpath
Case 2: $C - r_{im} \geq C - r_{mj}$
$r_{im} = 5$
$r_{mj} = 7$
i
m
j
(m, j) will also be removed, so a new (i, j) lightpath will not affect the degrees at i and j, nor wavelength constraint violation
Case 3: $I_j \geq I_m - 1$
i
m
j
Adding a new (i, j) lightpath will generate a new max degree, so we cannot use this (i, j) pair
Case 4: need to replace (i, m) with (i, j)
$r_{ij} = 4$
$r_{im} = 5$
$r_{mj} = 3$
i
m
j
Since (m, j) remains, more wavelengths are used between nodes m and j. Check if wavelength constraints are violated. If so, try next (i, j) pair

Figure 4.6: Various cases for $I_m > O_m$ in the Ring Grooming Algorithm

furthermore, the indegree of node j is already very high, making the rerouting will generate a new maximum degree at node j, making the Min-Max objective unimproved. Rerouting in such a case is not desirable as it does not lead to any progress. Case 4 is the case in which the rerouting will result in more lightpaths passing through link mj. In this case, we need to check for wavelength availability to decide if the rerouting is valid.

The above four cases, examined in sequence, contain all the possibilities and corresponding options. The other symmetric cases are easy to identify and can be coped with in similar ways.

We now argue that at the end of each iteration of the **repeat** loop, the algorithm produces a solution to the logical topology and traffic routing sub-problems that is feasible (i.e., no link carries more than W lightpaths – recall that we are not concerned with wavelength assignment at this point), and the maximum nodal degree is no larger (but possibly smaller) than that of the logical topology at the beginning of the iteration. At each iteration of the **repeat** loop, the algorithm tries to replace the lightpaths (i, m) and (m, j) for some nodes $i \prec m \prec j$ with a (possibly) new and longer lightpath (i, j) so as to decrease the nodal degree of node m. However, no action is taken if replacing the two lightpaths with the longer one would violate any wavelength constraints or would increase the degrees of i or j to more than the maximum at the start of the iteration, as we can see at Step 15. Since the initial topology at Step 2 of the algorithm is feasible, we conclude that the topology at the end of each iteration will be feasible and will not increase the maximum nodal degree.

The running time complexity of our algorithm is determined by the main iteration between Steps 6 and 21. In turn, the complexity of the iteration is determined by the two **for** loops, the inner **for** loop from Step 11 to 14, and the outer loop from Step 8 to 20. Each of these loops takes time $O(N^2)$ in the worst case, where N is the number of nodes in the ring. Therefore, each iteration through the **repeat** loop from Step 6 to 21 takes $O(N^4)$ time in the worst case. The main iteration of the algorithm (i.e., the **repeat** loop) will be executed at most $N\delta$ times, where δ is the maximum decrease in the degree of any node. Since the value of δ is always less than the number W of wavelengths, the worst-case complexity of the algorithm in Figure 4.5 is $O(WN^5)$. However, as we will discuss in the next subsection, in practice, our algorithm runs much faster than the above worst-case analysis indicates; in fact it has never taken more than a few tens of milliseconds for any problem instance with $N = 16$ nodes and $W = 128$ wavelengths.

4.3.2 An Algorithm for Wavelength Assignment on Unidirectional Rings

The output of the algorithm we presented in the previous subsection is a set of lightpaths R between pairs of ring nodes (i.e., a logical topology), and a routing of the traffic elements $\{t^{(sd)}\}$ over these lightpaths. While the algorithm guarantees that the resulting logical topology is such that no link in the ring network carries more than W wavelengths, it may not be possible to color the lightpaths in R using no more than W wavelengths. In fact, the problem of deciding whether there exists a coloring of the set of lightpaths R that uses no more than W wavelengths is NP-Complete [52]. In this subsection, we present a polynomial-time algorithm to perform wavelength assignment with at most W colors; the tradeoff in ensuring that the number of wavelengths does not exceed W is a modification of the logical topology (i.e., the set R) which may result in an increase in the degree of some node in the ring. Consequently, the objective F of our optimization problem may increase. Therefore, we then refine the new logical topology to decrease the objective F.

Let us start by describing how to assign wavelengths to the lightpaths of set R. Our approach is based on the observation that, while the wavelength assignment problem is hard for ring networks, it is solvable in linear time in paths [31]. Consider some node m of the ring. Let R_1 denote the lightpaths in R that *optically bypass* node m, and let $R_2 = R - R_1$ be the set of remaining lightpaths. The lightpaths in set R_2 can be viewed as the logical topology on a path network, and thus, can be colored using no more than W wavelengths. Now consider all the lightpaths in set R_1. It may be possible to color some of them without violating any wavelength continuity constraints; in general, however, there may be some lightpaths in this set that cannot be colored without the need for additional wavelengths. In this case, we break such a lightpath $(x, y), x \prec m \prec y$ into two lightpaths (x, m) and (m, y). The new lightpaths do *not* bypass node m, and thus, can be colored along with the lightpaths in set R_2 using no more than W wavelengths. Remember that W is a static constraint given as input of the grooming problem. While breaking such a lightpath will increase the indegree and outdegree of node m by one, this approach guarantees a coloring of the new set of lightpaths that satisfies the wavelength constraints. The following steps describe our algorithm in more detail.

1. Let $R_1 \subset R$ be the set of lightpaths that optically bypass node m. Let $R_2 = R - R_1$ be the subset of remaining lightpaths.

2. Sort the lightpaths in R_2 in increasing order of their length.

3. Use the first-fit policy to color the lightpaths in R_2. Note that this step is always possible since it corresponds to a first-fit wavelength assignment for a path network [23].

4. Sort the lightpaths in R_1 in decreasing order of their length.

5. Use the first-fit policy to color the lightpaths in R_1. If lightpath $l = (x, y), y \prec x$, cannot be colored, then: break l into two lightpaths, $l_1 = (x, m)$ and $l_2 = (m, y)$ which do not bypass node (m); increment the indegree and outdegree of node m to accommodate the new lightpaths; and repeat from Step 1 with $R_2 \leftarrow R_2 \cup \{l_1, l_2\}$ and $R_1 \leftarrow R_1 - \{l\}$.

Note that we arbitrarily select the node m in the above algorithm. One possible improvement would be to consider all N possible nodes, run the algorithm for each of them, and then select the solution with the least maximum nodal degree. We find in our experiments that this approach is only slightly better than the random choice of m.

Let Δ denote the increase in the nodal degree of node m after the termination of the above wavelength assignment algorithm. This increase is due to the fact that Δ lightpaths which optically bypassed node m under the initial logical topology defined by the set R, have now been broken into two lightpaths each. Of the 2Δ new lightpaths, Δ terminate at node m and Δ originate from it. Therefore, node m needs an additional Δ pairs of LTE, one for each of the original Δ lightpaths that used to bypass the node. Consequently, this increase of the objective F by Δ at node m increases the LTE cost in the ring network.

We now show how we can refine the new logical topology at the end of the wavelength assignment algorithm to improve on the objective F. Our approach is based on the observation that each additional pair of LTE at node m, one for an incoming and one for an outgoing lightpath, can be thought of as a *wavelength converter*. Indeed, consider, one of the Δ lightpaths that initially bypassed the node. This lightpath was broken by the algorithm into two shorter lightpaths that terminate at and originate from the node, respectively. This action was taken in Step 6 of the algorithm because it was not possible to assign the two shorter lightpaths on the two links in either side of node m the same color. Therefore, the additional pair of LTE at node m acts as a converter, changing the wavelength of the new incoming short lightpath to the wavelength of the new outgoing lightpath.

Consider a logical topology and corresponding feasible wavelength assignment on a ring network that requires Δ converters at some node m. The recent study in [11] showed that it is possible to modify the wavelength assignment such that 2Δ converters are uniformly distributed across all N ring nodes (i.e., each node has at most $\lceil 2\Delta/N \rceil$ converters). Therefore, we use this algorithm to distribute the Δ pairs of LTE (i.e., "converters") at node m to the other ring nodes. As a result, the maximum degree at all the ring nodes will increase, but the maximum nodal degree of the network (i.e., of node m) will decrease, resulting in a new logical topology with a smaller value for the objective F. For the details of this algorithm, the reader is referred to [11].

As we will present in Section 4.6.3, the wavelength assignment described above works well for various types of traffic patterns, and for link loads up to 95%.

4.4 Revised Algorithm for Bidirectional Rings

The WDM bidirectional ring case with the Min-Max objective is even more complicated than the unidirectional case. As was stated in Section 4.2, our objective is to minimize the maximum in/out nodal degree, regardless of the lightpath directions.

First, we *do not allow* a single lightpath to occupy both directions of a link, as is shown in Figure 4.7. That is because it uses up some wavelengths in both direction without contribution to our Min-Max objective, which this is globally undesirable. Moreover, since the lightpath takes fibers for both directions, it will make network protection more complicated.

Second, we *do allow* a traffic component to be carried from its source node to the destination node on a sequence of lightpaths some of which are in one direction and some in the reverse direction.; thus a traffic component may traverse the same link multiple times in either direction. Such routing of traffic components may offer advantages in terms of the Min-Max objective we consider, if it can make use of the remaining capacities in these lightpaths; in other words, not allowing a traffic component to be routed in such a manner might require the setup of additional lightpaths, causing an increase in the degree of some node.

We now present a traffic grooming heuristic for bidirectional ring networks. The heuristic first applies shortest path routing to decompose the bidirectional problem instance

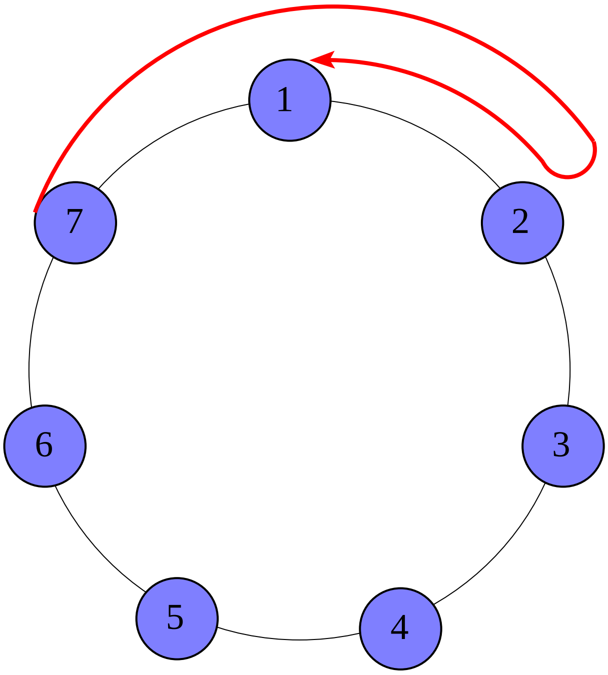


Figure 4.7: An example of a lightpath not allowed in bidirectional rings

into two unidirectional problem instances, as shown in Figure 4.8. Shortest path routing forces traffic demands to travel over the minimum number of links, and makes efficient use of the available network capacity (wavelengths).

The heuristic then solves each unidirectional sub-problem using the algorithm we presented in Figure 4.5, and combines the individual solutions into a solution for the original problem by adding the in/out nodal degrees determined by each solution for each node. Note that, if the sub-problems are solved independently of each other, it may happen that some node x be the maximum degree node in both solutions. In this case, the final combined solution may not be a good one, since node x may end up with a large overall degree. Therefore, we modify the algorithm to take into account the overall objective (for the original bidirectional problem). In this approach, the solution to each unidirectional sub-problem takes into account the solution to the other sub-problem, and vice versa. Results to be presented in the next section indicate that this shortest path decomposition produces results close to the optimal solution in which the routing of traffic components is determined by the solution to the ILP.

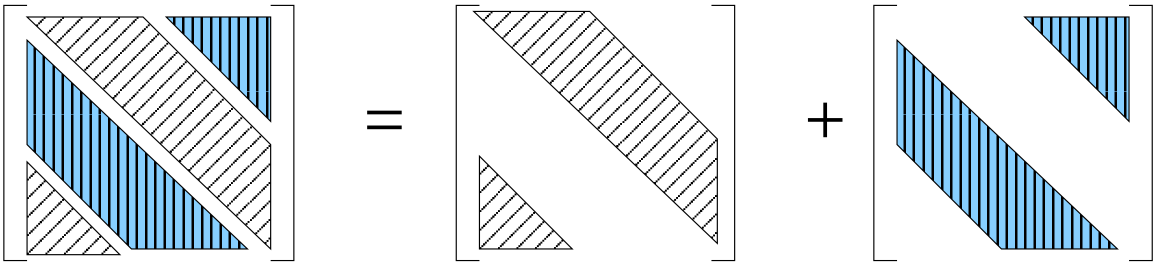


Figure 4.8: Decomposition of a bidirectional ring using shortest-path routing

Specifically, our bidirectional traffic grooming algorithm consists of the following steps.

- **Step 1.** Use shortest path routing to determine the direction in which each traffic component will be routed. Decompose the original traffic matrix T into two traffic matrices T_r and T_l containing the components routed in the clockwise and counterclockwise direction, respectively, and such that $T = T_r + T_l$. This decomposition creates two unidirectional ring sub-problems with respective traffic matrices T_r and T_l.

- **Step 2.** Use the heuristic we developed in the previous section (refer to Figure 4.5 in Section 4.3) to solve the clockwise ring, and record the resulting in/out degrees at each node. Then, solve the counterclockwise ring, with the following minor change in the heuristic: when selecting the node m with the maximum degree (refer to Step 7 in Figure 4.5), we add the current degrees for both directions, so that m is the node with the maximum *aggregate* degree. At the end of this step, we obtain a solution S_1.

- **Step 3.** Repeat Step 2, but this time solve the counterclockwise ring first, then solve the clockwise ring accordingly, to obtain a solution S_2. Return the best solution among S_1 and S_2.

It is clear that the asymptotic complexity of the above algorithm is the same as that of the unidirectional ring algorithm, which is polynomial.

Note that if the above decomposition approach cannot find a feasible solution, it doesn't mean the original bidirectional ring has no feasible solution. In this case, we need to try decomposition methods other than the shortest-path approach. However, our experiments show that the shortest-path decomposition already works well for many practical traffic patterns, as we will describe in Section 4.6.4.

4.5 Modified Algorithm for the Overall Objective

In [58], Zhang and Qiao gave an effective algorithm for grooming in WDM rings to minimize the number of LTEs in the whole network, which we referred to as the *Overall* objective. Unlike their studies, we assume that a *digital cross-connect (DXC)* is equipped at each network node of the ring, which allow for traffic demands to be carried on a different wavelength after stopping at an intermediate node. Therefore, in our solutions, each node can be equipped with different number of incoming devices that drop/demultiplex lightpaths than outgoing devices that add/multiplex lightpaths. This nodal asymmetry will potentially reduce the cost on LTE in the system, while in [58], the authors didn't use DXC at any node, and the solutions required that incoming/outgoing LTE equipment at each node is symmetric. The relationship between the devices at a network node is shown in Figure 1.4 in Chapter 1 with detailed explanations.

To accommodate our algorithm to take care of both Min-Max and Overall objectives, we make the following changes: after finding the min-max solution without wavelength assignment, we continue reducing the overall degree using a greedy approach. At each node, we may be able to make an additional iteration (see Figure 4.5, Steps 7 to 20) to reduce the overall in/out degrees by 0, 2 or 4 without increasing the Min-Max objective. We randomly choose a node from those that could reduce the largest number of degrees without changing the Min-Max result, and run an iteration to improve the Overall objective. Repeat this procedure until no decrease can be made, then the same procedure can be made for wavelength assignment as in Section 4.3.2 to get the final solution.

To find the nodes whose selection as m (please refer to Figure 4.5, Step 7) could reduce the largest number of degrees, we need to study a variety of cases as in Figure 4.6 with minor changes to take care of the Overall objective. We skip the details to avoid repetition. Note that the adapted algorithm above will not change the original Min-Max objective. A comparison of our algorithm and the one given by Zhang and Qiao in [58] can be found in Section 4.6.5

4.6 Numerical Results from Experiments

In this section, we present experiments to demonstrate the performance of the traffic grooming algorithm we described in the previous sections. The experiments are characterized by the following parameters: the traffic pattern, the number N of nodes in the ring, the number W of wavelengths per link, the capacity C of each wavelength, and/or the load L on the link carrying the most traffic. The maximum amount of traffic that can flow through a link is WC, hence we express the load L as a percentage of WC. It is hard to define *all-electronic routing* in bidirectional rings, since there are two ways to route each lightpath. We simply assume the shortest-path routing and calculate the traffic loads using 4.14.

In order to demonstrate the performance and generality of our algorithm, we consider three traffic patterns in our study. For each traffic pattern, the traffic matrix $T = [t^{(sd)}]$ of each of the 50 problem instances is generated by drawing $N(N-1)$ random numbers (rounded to the nearest integer) from a Gaussian distribution with a given mean t and standard deviation σ that depend on the traffic pattern. The three traffic patterns are:

1. *Uniform pattern.* To generate this traffic pattern, we first determine the mean value t of the Gaussian distribution according to the desired link load L, and we let the standard deviation be 10% of the mean t. As a result, most elements of the traffic matrix $T = [t^{(sd)}]$ take values close to the mean t. The traffic is also balanced across the links of the network, with each link carrying a load close to L.

2. *Random pattern.* The traffic matrix for this pattern is generated in a manner very similar to that of the uniform pattern. The main difference is that we let the standard deviation of the Gaussian distribution be 150% of the mean t. Consequently, the traffic elements $t^{(sd)}$ take values in a wide range around the mean, and the loads of individual links also vary widely. With such a high standard deviation, the random number generator may return a negative value for some traffic element; in this case, we set the corresponding $t^{(sd)}$ value to zero. Also, if a traffic matrix generated in this manner is infeasible (i.e., the load on some link exceeds the value WC), then we discard it and we generate a new matrix for the corresponding problem instance. Note that with the high variation, the link load L is also not as balanced, and we can only give a range of link loads for the instances. We believe that this ‘random’

pattern with normal distribution resembles the real cases we might encounter better than the plain uniformly random traffic pattern.

3. *Locality pattern.* This traffic pattern is designed to capture the traffic locality property that has been observed in some networks. Specifically, the traffic elements $t^{(sd)}$ are generated such that, on average, 50% of the traffic sourced by any node s is destined to node $s \oplus 1$, 30% is for node $s \oplus 2$, and 10% is for node $s \oplus 3$. The remaining 10% of the traffic generated by node s is distributed equally among the other $N - 4$ nodes of the ring.

For bidirectional rings, we simply double the mean values for the unidirectional ring case for each traffic pattern. Since we use shortest-path routing in our algorithm, there is no routing of distance over $N/2$, thus the values of traffic load and all-electronic routing results are not doubled accordingly.

We also test the performance of our modified algorithm for the *Overall* objective, which is to minimize the total number of degrees in the virtual topology. Comparisons are made with existing grooming algorithm on both the Min-Max and traditional aggregate objectives.

4.6.1 Min-Max Results for Small Unidirectional Rings

For small unidirectional ring networks with up to 8 nodes, we were able to calculate the optimal values using existing ILP solver tool, and make comprehensive comparisons on the related set of values. For problem instances with $N = 10$ nodes, CPLEX run on a Sun UltraSparc workstation for several days without terminating.

For each experiment (i.e., each set of values for the above parameters), we generate 50 problem instances, and we compare the following values for each instance:

- The *lower bound* F^l on the Min-Max objective F given by:

$$F^l = \max_s \left(\max \left(\left\lceil \frac{\sum_d t^{(sd)}}{C} \right\rceil , \left\lceil \frac{\sum_d t^{(ds)}}{C} \right\rceil \right) \right) \tag{4.13}$$

  To obtain the lower bound, we note that each node must source and terminate a sufficient number of lightpaths to carry the traffic demands from and to this node, respectively. In the above expression, F^l is the maximum of this value over all ring nodes, and can be determined directly from the traffic matrix.

- The *optimal value* $F^\star$ of the objective F, obtained by using CPLEX [1] to solve the ILP we presented in Section 2.2. We note that we can solve the ILP with CPLEX only for small problem instances, as we explain later. The CPLEX software can automatically remove redundant constraints and variables before searching the solution space, so it actually works on a set of ILP formulations optimized from the one we have given.
- The value of the objective F returned by our algorithm.
- The value F^e of the objective for a network using *all-electronic routing.* This is the value of the objective for a logical topology that consists of only single-hop lightpaths, i.e., one in which no optical switching of wavelengths takes place. The value of F^e corresponds to the number of wavelengths needed to carry the traffic on the link with the heaviest traffic load, and can be determined from the traffic matrix as:

$$F^e = \max_l \left\lceil \frac{\sum_{(x,y)\in B(l)} t^{(sd)}}{C} \right\rceil \tag{4.14}$$

Figures 4.9-4.11 plot the lower bound F^l, the optimal value $F^\star$, the value of the goal F returned by our algorithm, and the all-electronic value F^e, for 50 instances of each experiment, and for the stated values of parameters W, C, and L. Figure 4.9 shows results for the uniform pattern, Figure 4.10 for the locality pattern, and Figure 4.11 for the random pattern.

For each figure, we present the instances with the sequence they are randomly generated, instead of sorting them in a specific sequence. We do this to show the character of randomness for all the traffic patterns we consider.

There are several observations that we can make regarding the relative behavior of the four curves in the three figures. We first observe that the value of the objective F returned by our algorithm is close to the optimal, and that the optimal is close to the lower bound. On the other hand, the value of the all-electronic solution is significantly higher than the other three values, anywhere from twice to four times larger. This relative behavior of the four curves is consistent across all three traffic patterns, and has been observed for a wide range of values for the system parameters. These results demonstrate the effectiveness of our algorithm, which can find a solution close to the optimal in a tiny fraction of the time required for CPLEX to terminate; in fact, for the experiments presented in this section, the running time of our algorithm never exceeds a few milliseconds, whereas CPLEX, depending on the values of the parameters, may take anywhere from a few minutes to a few hours.

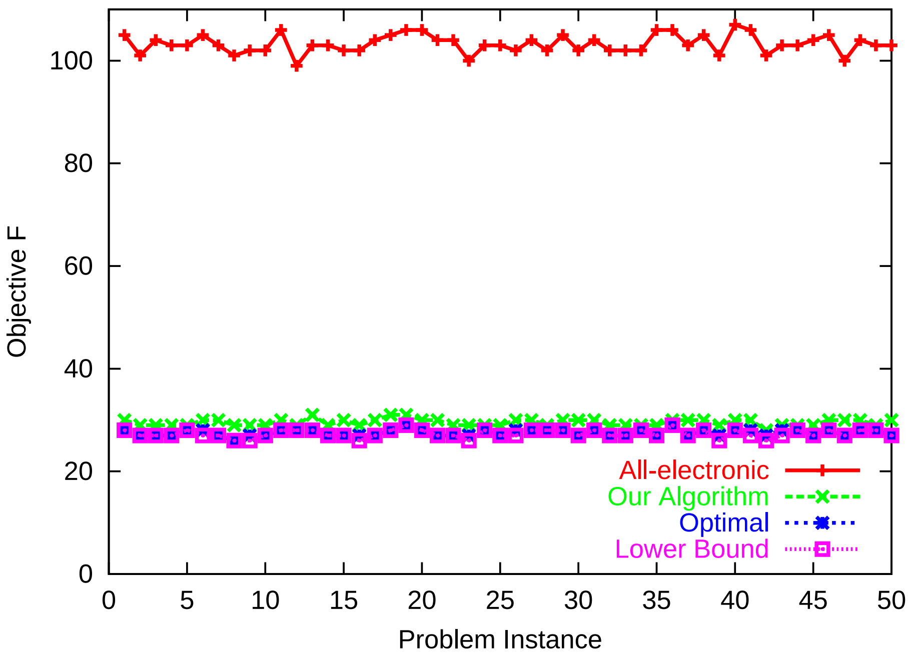


Figure 4.9: Uniform pattern, $N = 8$, $W = 128$, $C = 12$, $L = 80\%$

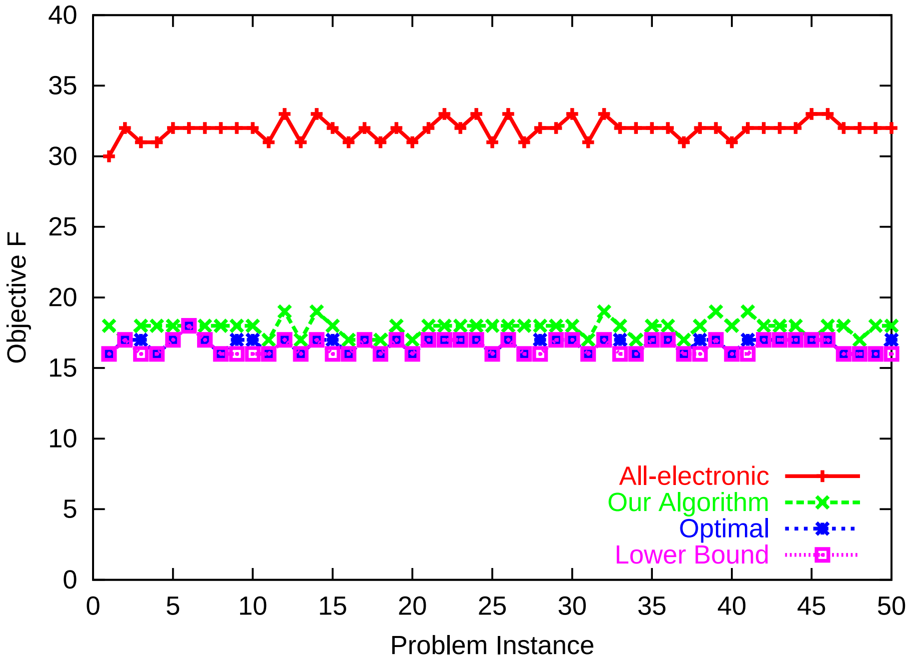


Figure 4.10: Locality pattern, $N = 8$, $W = 64$, $C = 12$, $L = 50\%$

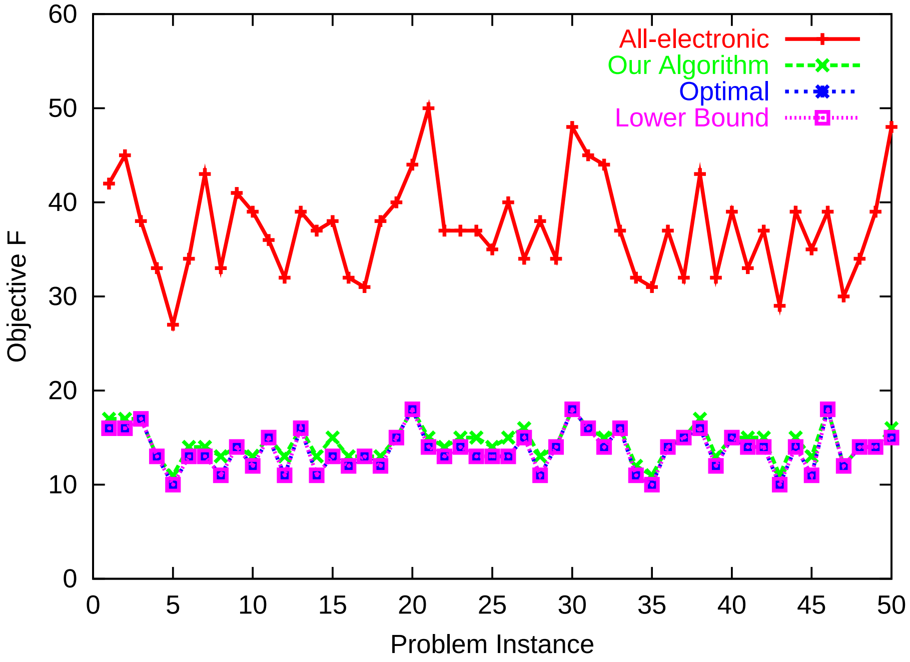


Figure 4.11: Random pattern, $N = 8$, $W = 64$, $C = 12$, $L = 50 - 80\%$

From the figures, we can also see that the value F^e for the all-electronic solution is always close to LW. This result is exactly what is predicted from expression (4.14): since the traffic load on the most congested link is LWC, the number of lightpaths needed to carry this traffic is $\lceil LW \rceil$, and these lightpaths must terminate at the nodes at the two ends of this link, requiring an equal amount of LTE at each node. On the other hand, the solution returned by our algorithm, as well as the optimal one, are significantly lower than the all-electronic value. This result indicates that our approach of minimizing the maximum LTE cost in the network can produce significant cost savings. Another related observation is that, by using a Min-Max objective, we can ensure that the cost of any individual network node (as well as the overall network cost) is determined by the traffic demands of the node, and will not scale with the number of wavelengths. Specifically, the values of the objective F in Figures 4.9-4.11 are small compared to the number of wavelengths we used for the experiments ($W = 64, 128$), and are close to the lower bound, which represents the minimum LTE cost to accommodate the traffic demands for any ring node (see (4.13). This result demonstrates that, with an appropriate network design approach, a network with optical switching capabilities can scale to large numbers of wavelengths without a corresponding increase in cost; we reached a similar conclusion in a different context in another recent study [56].

Finally, from the three figures we see that the optimal value of the objective depends on both the traffic pattern and the maximum link load L. This is more evident in Figure 4.11, where the high degree of randomness in the traffic matrices produces a variety of patterns and maximum link loads across the 50 problem instances, resulting in more jagged curves. (Note, however, that even in this case, the curve produced by our algorithm tracks the optimal curve very closely, attesting to the quality of our approach.) This observation suggests that, for the purposes of dimensioning the network for unknown future demands, the network designer should run our algorithm for a variety of traffic patterns and loads, and then equip each node with the amount of LTE determined by the highest value of F.

4.6.2 Min-Max Results for Large Unidirectional Ring Networks

In this subsection, we present results for unidirectional ring networks with $N = 16$ nodes and $W = 128$ wavelengths; such networks are of practical interest because the maximum size of a SONET ring is 16 nodes, and also because WDM links supporting 128 wavelengths are becoming commercially available. Since we were not able to use CPLEX to obtain the optimal solution, the figures in this section only contain three curves: one for the all-electronic value F^e, one for the value of F returned by our algorithm, and one for the lower bound F^l. Our algorithm, on the other hand, needed less than a second to find a solution for each problem instance that we present in this section.

Figures 4.12, 4.13, and 4.14 present the results of 50 different problem instances for each of the uniform, locality, and random traffic patterns, respectively. The relative behavior of the three curves in these figures is very similar to that we observed in Figures 4.9-4.11. In particular, the values of the objective F returned by our algorithm are close to the lower bound, and track it well across the different traffic patterns and problem instances within each pattern. The all-electronic solutions, on the other hand, are again quite high with values at around LW. Therefore, all the conclusions regarding the scalability and cost-effectiveness of our traffic grooming approach are valid for large WDM ring networks as well.

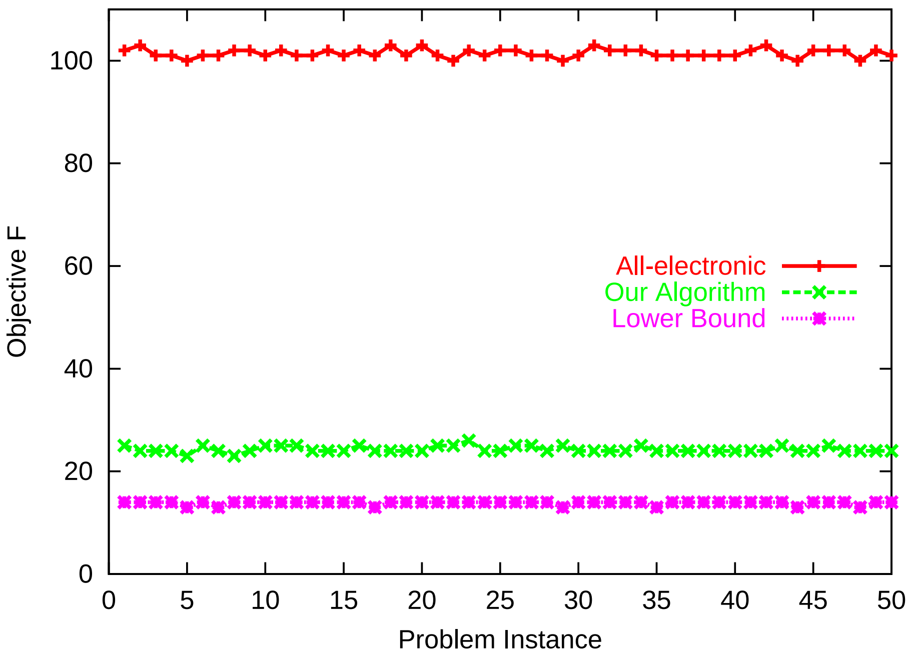


Figure 4.12: Uniform pattern, $N = 16$, $W = 128$, $C = 12$, $L = 80\%$

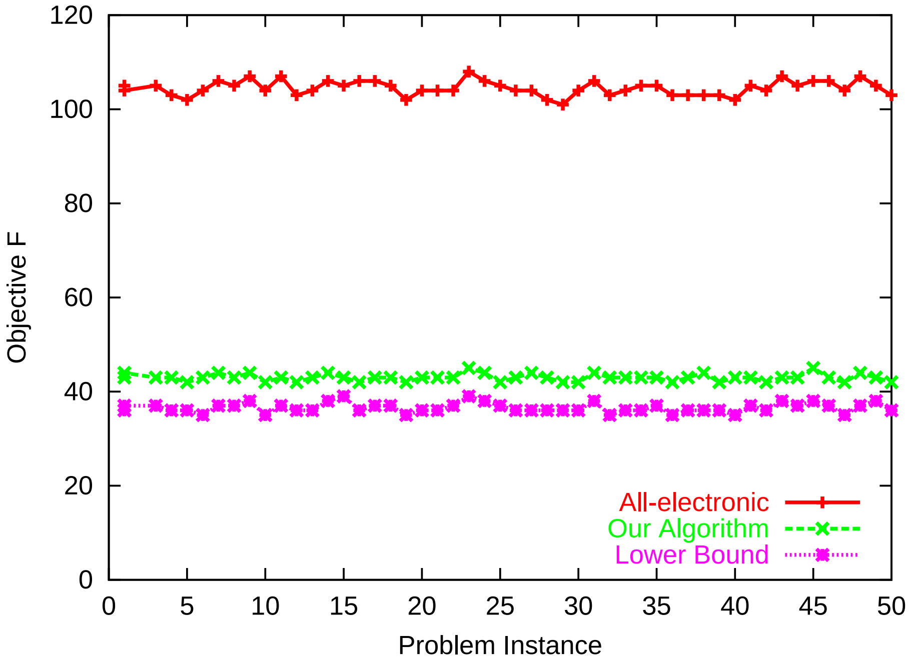


Figure 4.13: Locality pattern, $N = 16$, $W = 128$, $C = 12$, $L = 80\%$

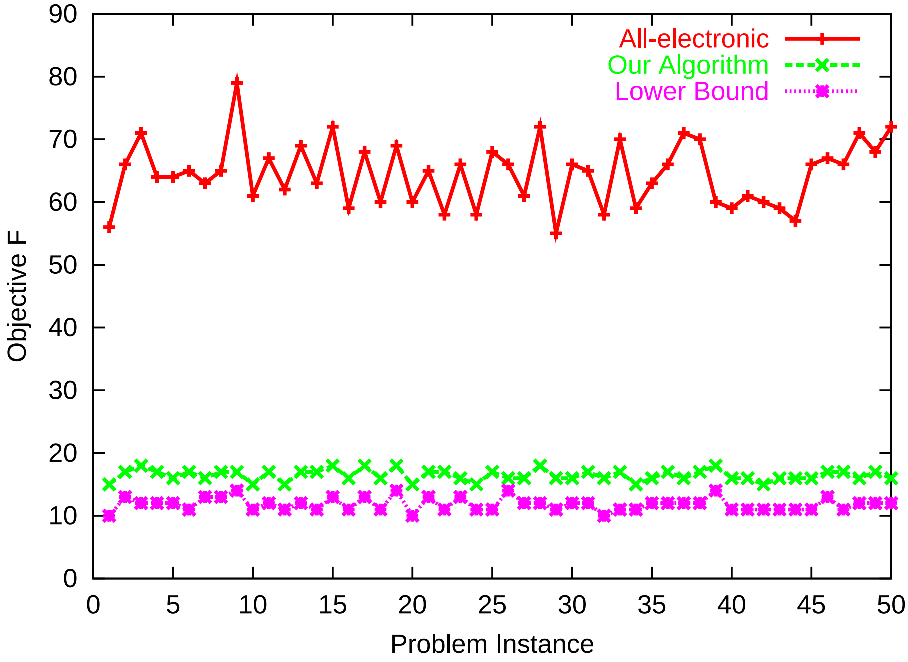


Figure 4.14: Random pattern, $N = 16$, $W = 128$, $C = 12$, $L = 40 - 60\%$

4.6.3 The Effect of the Traffic Load on Min-Max Objective

Let us now investigate how our solutions scale with the load L on the most congested link of the network. In Figures 4.15 and 4.16, we plot the values of F^e, F^l, and F against the load L for a uniform traffic pattern; each point in the figures is the average over 50 problem instances for the given value of L (the optimal values $F^\star$ are also plotted for the small network in Figure 4.15). The experiments presented in the two figures are identical in all respects except that Figure 4.15 is for 6-node rings, while Figure 4.16 is for 16-node rings. We have obtained similar results for the other two patterns as well.

As we can see, F^e, F^l, and F all increase linearly with the load L. However, the curve corresponding to the all-electronic solution has a slope much steeper than that of the curves corresponding to the lower bound and our solution. This behavior demonstrates that the cost benefit of our optimization approach increases with the load of the network. This property is an important one, and it implies that network operators will be able to operate the network at high loads with only an incremental increase in cost.

Another important property of our Min-Max optimization approach becomes ev-

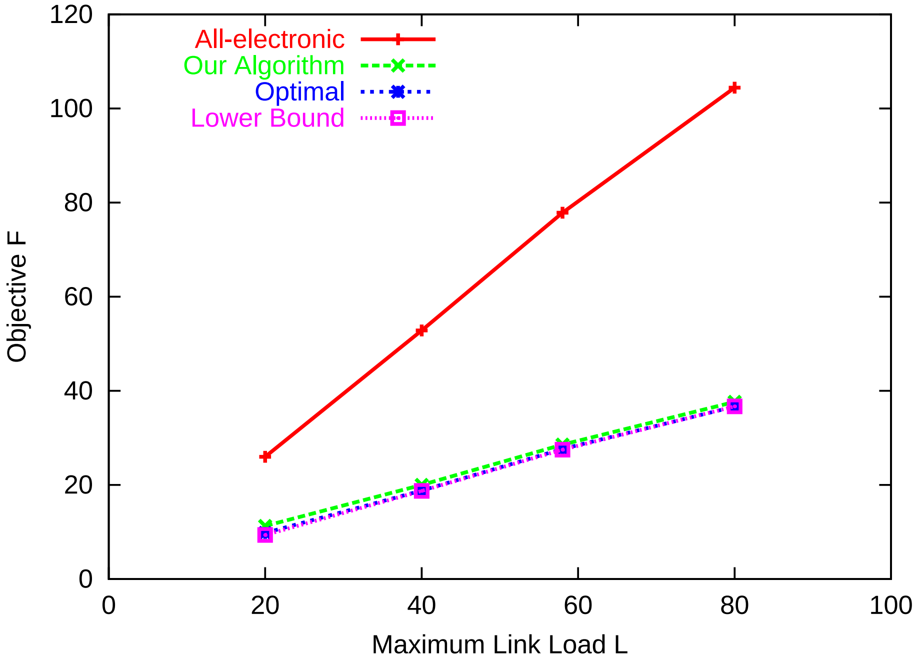


Figure 4.15: Uniform pattern, $N = 6$, $W = 128$, $C = 12$, various loads

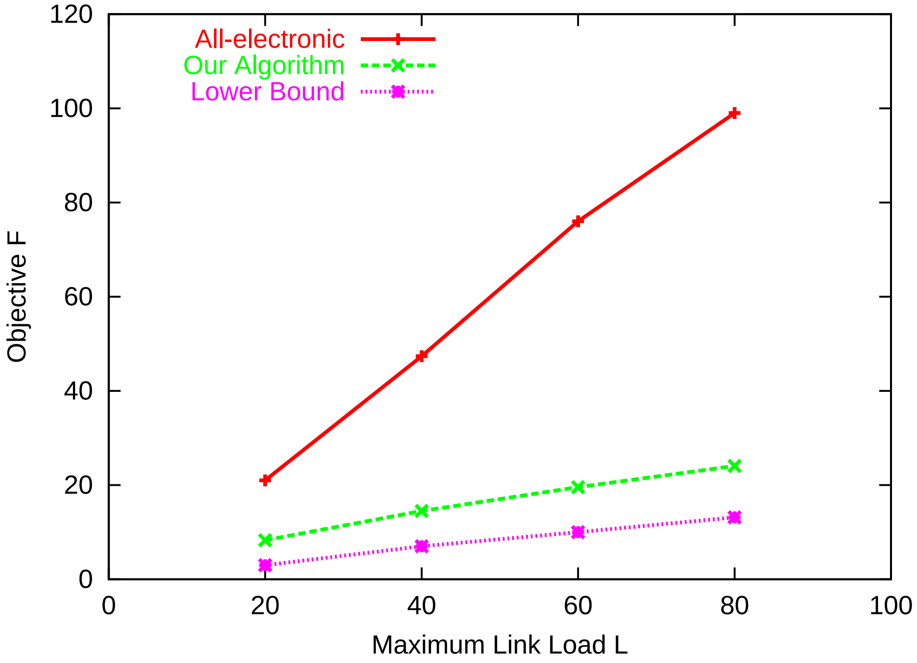


Figure 4.16: Uniform pattern, $N = 16$, $W = 128$, $C = 12$, various loads

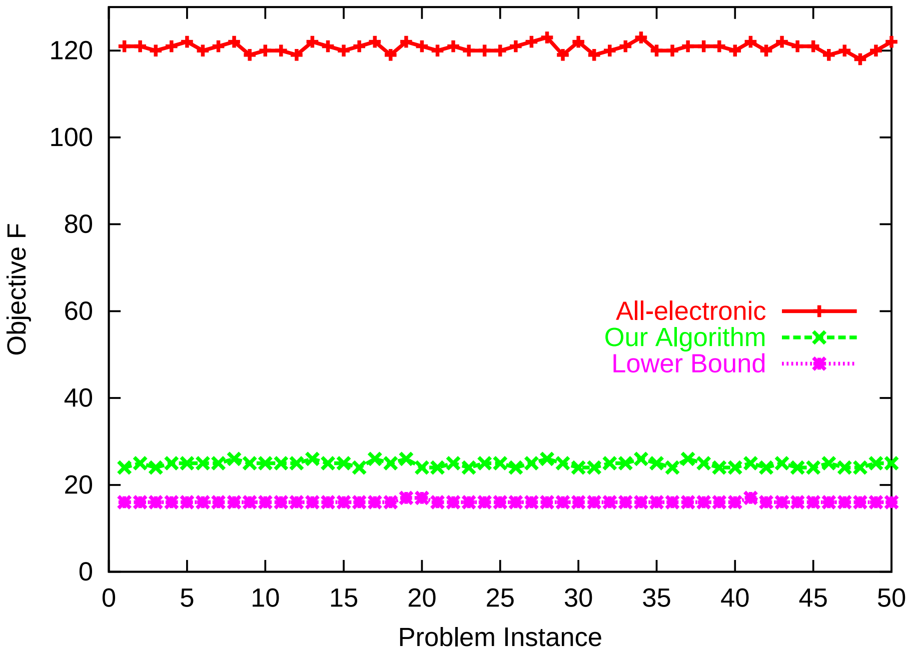


Figure 4.17: Uniform pattern, $N = 16$, $W = 128$, $C = 12$, $L = 95\%$

ident once we compare the curves of Figure 4.15 to those of Figure 4.16. Specifically, we note that the curves corresponding to the all-electronic solutions have similar slopes in both figures. On the other hand, the curves corresponding to the lower bound and our solution have a steeper slope in Figure 4.15 (which corresponds to a 6-node network) than in Figure 4.16 (for a 16-node network). Equally important, for a given load L, the actual values of the lower bound and our solution are *lower* for the larger network than the respective values for the smaller network; this is true despite the fact that link loads of both networks are similar and that the 16-node network carries a much larger aggregate amount of traffic than the 6-node network. Based on this observation, we conclude that our solution approach scales well not only with the load and number of wavelengths, as we explained above, but also with the size of the network.

Finally, in Figure 4.17, we present results for problem instances with $N = 16$ nodes, a uniform traffic pattern, and a high load $L = 95\%$. Let us compare Figure 4.17 to Figure 4.12 which plots results for the same values of the system parameters, except that the load $L = 80\%$ in Figure 4.12. As we can see, the all-electronic solution F^e is about 20%

higher in Figure 4.17, as expected, and the lower bound is also somewhat higher. However, there is little difference in the values of F returned by our algorithm when the load increases from 80% to 95%. We have observed similar behavior for the other two traffic patterns, reinforcing our earlier conclusion that the benefit of our Min-Max optimization approach increases with the traffic load, enabling cost-effective network operation at very high loads.

4.6.4 Results for Bidirectional Ring Networks

We now present our results for bidirectional rings with $N = 5$ nodes. We found that for rings of more than 6 nodes, CPLEX will take too long to get the optimal solutions. This is because the ILP formulation for bidirectional rings are more complicated than in the unidirectional case, as is shown in Section 4.1.2.

As we have discussed, the traffic load cannot be decided in the bidirectional case because of indefinite routing options. Therefore, we use *grooming effectiveness* to characterize the results. The *grooming effectiveness* is defined here as *the ratio of* the maximum degree of a solution *and* the one obtained with all-electronic shortest-path routing. By definition, the smaller the grooming effectiveness, the better the solution with regard to our Min-Max objective. For each case, we calculate and compare four values:

- The *optimal* solution. This is the solution given by CPLEX using the ILP formulation for bidirectional rings.

- The *decompose optimal* solution. In this case, we use the shortest-path decomposition we described in Section 4.4, and solve the two unidirectional ring instances independently using CPLEX on the corresponding unidirectional ILPs. We then combine the two solutions, and compute the overall Min-Max result (and corresponding grooming effectiveness value).

- The *decompose heuristic* solution. This is similar to the 'decompose optimal' method, except that we use the unidirectional heuristic in Figure 4.5 and the corresponding wavelength assignment algorithm in Section 4.3.2 instead of CPLEX to solve each unidirectional ring instance.

- The *bidirectional algorithm* solution, which is obtained by running the algorithm we presented in Section 4.4. Recall that this algorithm focuses on the aggregate degrees

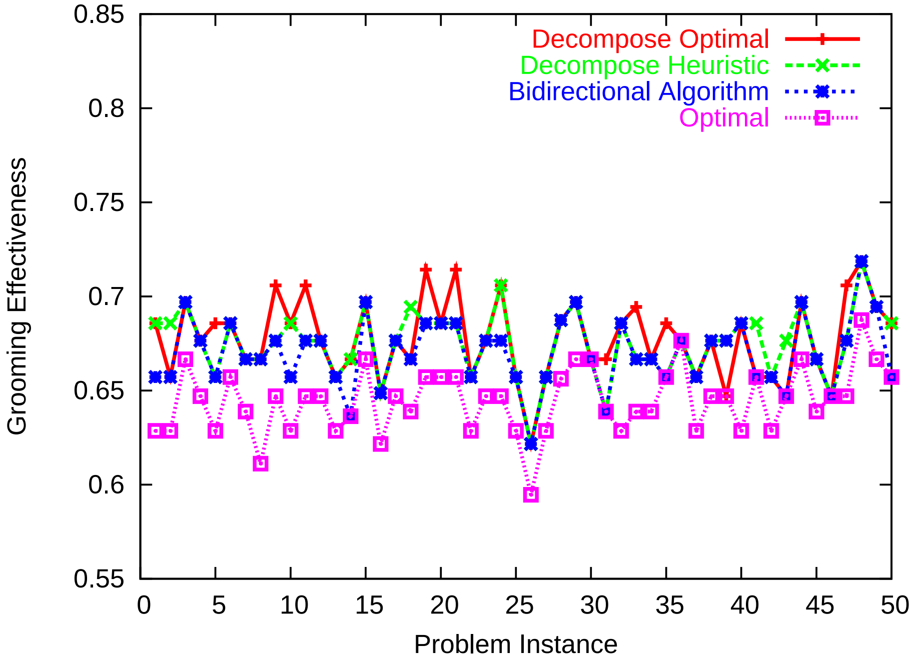


Figure 4.18: Bidirectional Uniform, $N = 5$, $W = 64$, $C = 12$, $F^e = 33 - 37$

for the bidirectional ring as a whole, and is such that the solution to each unidirectional ring instance takes into account the solution to the other.

Figures 4.18, 4.19, and 4.20 present the results of 50 different problem instances for each of the uniform, locality, and random traffic patterns, respectively. It is not surprising to see that the *grooming effectiveness* is the best in the locality pattern, since the pattern itself favors shortest-path routing. In both the uniform and locality patterns, all the four solutions give very close results. However, in the random pattern, we can see that the *decompose optimal* method gives worst results, while *decompose heuristic* approach is much better. The reason may be that CPLEX stops while finding the first optimal solution for each of the unidirectional ring, which may have the Min-Max degree existing on the same node (*e.g.*, node 1) for the two unidirectional rings; while our algorithm returns more balanced degrees by taking the first ring into consideration while working on the second.

Finally, the *bidirectional algorithm* is closest to the optimal, as expected, since it takes into consideration the aggregate objective.

From the three figures, we observe that our shortest-path decomposition always

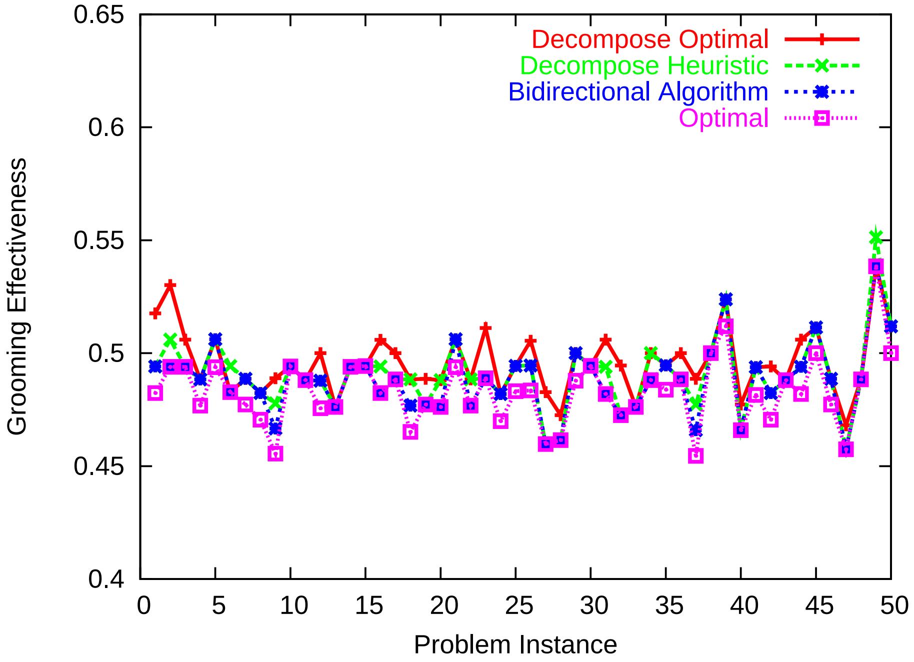


Figure 4.19: Bidirectional Locality, $N = 5$, $W = 64$, $C = 12$, $F^e = 78 - 94$

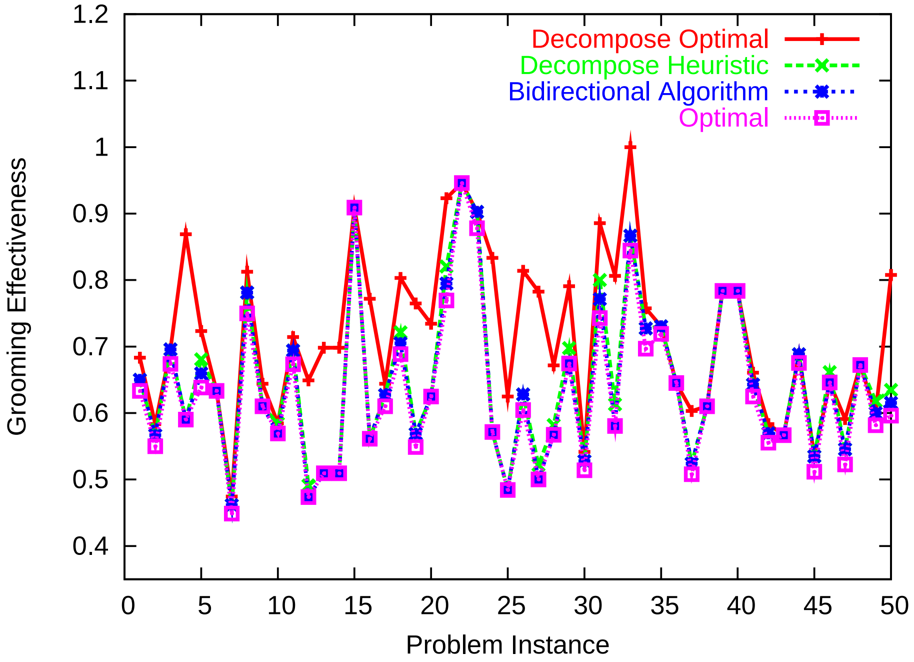


Figure 4.20: Bidirectional Random, $N = 5$, $W = 64$, $C = 12$, $F^e = 31 - 78$

works well for the five-node ring networks we have considered. To further evaluate the effectiveness of the decomposition, we conducted additional experiments on five-node bidirectional rings with the random traffic pattern. We randomly generated 100,000 problem instances and obtained the optimal solutions using CPLEX. The resulting maximum nodal degrees range from 9 to 74, with an average around 33. We then run the bidirectional heuristic and recorded the difference in maximum nodal degree from the optimal solution. Of the 100,000 instances, our algorithm found the optimal solution in 38,895 cases (39%); in 55,519 cases (55.5%) our algorithm found a solution whose maximum degree is one more than the optimal; in 5,550 cases (5.5%) our solution are two more than the optimal; and in 36 cases, our solution are three more than the optimal. These results indicate that using shortest path routing in bidirectional rings works well, at least for small rings and for a wide range of traffic patterns.

Although we cannot use CPLEX to solve larger problem instances, we believe that shortest path decomposition combined with the bidirectional algorithm we presented in Section 4.4 will work well for moderate size rings (i.e., rings with up to 16 nodes, the maximum size of SONET networks). To see why this is so, recall that the results in Figures 4.12-4.14 indicate that the solution to each unidirectional problem is close to the lower bound. In addition, the sum of the lower bounds for each unidirectional sub-problem is a lower bound for the original bidirectional problem (since the lower bound, Equation 4.13 only accounts for the total amount of traffic originating or terminating at each node, which is independent of the routing of traffic). Therefore, the maximum nodal degree resulting from combining (adding) the solutions to the individual unidirectional problems will also be close to the lower bound.

4.6.5 Compare with Existing Algorithm for the Overall Objective

As has been mentioned in Section 4.5, Zhang and Qiao in [58] gave an effective algorithm for the Overall objective. In this subsection, we will compare our adapted algorithm described in Section 4.5 with this existing one for both objectives.

The lower bound for the Overall objective can be obtained in a way similar to the Min-Max lower bound we get in Equation 4.13 of Section 4.6.1. To obtain the overall lower bound, we need to sum up the nodal in/out degrees. So the new equation becomes:

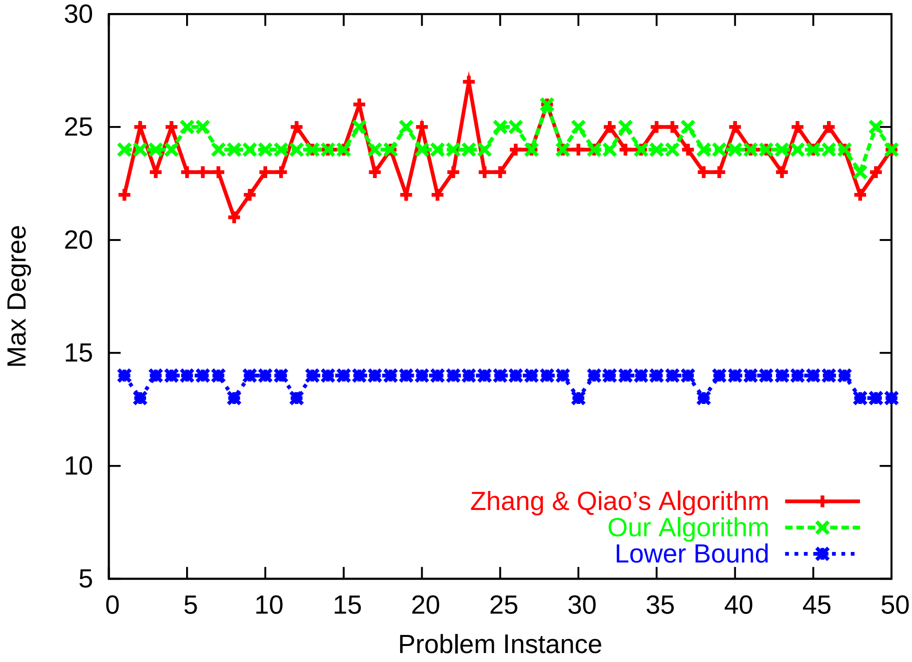


Figure 4.21: Uniform Min-Max, $N = 16$, $W = 128$, $C = 12$, $L = 80\%$

$$F_2^l \quad = \quad \max\left(\sum_s \left\lceil \frac{\sum_d t^{(sd)}}{C} \right\rceil, \sum_d \left\lceil \frac{\sum_s t^{(sd)}}{C} \right\rceil\right) \tag{4.15}$$

Note that both Equation 4.13 and 4.15 are topology-independent and approach-independent. That is, the lower bounds can be calculated regardless of the network topology and grooming methodology. Later in the following chapters, we will use Equation 4.15 again to calculate lightpath overall lower bounds on general topology networks, and compare them with results from a specific grooming approach.

We experiment with the same patterns and loads as in Section 4.6.2 for 16-node unidirectional rings. Three sets of data are compared: the Lower Bound; results from Our Algorithm with the above changes; and results from Zhang and Qiao's algorithm. For each set of parameters, 50 instances are generated, and comparisons on both the Min-Max and Overall objectives are made.

From Figures 4.21, 4.22 and 4.23, we can see that with respect to the Min-Max objective, our algorithm clearly outperforms Zhang and Qiao's algorithm for the locality and random patterns, while the results for the uniform pattern are mixed, with each algorithm

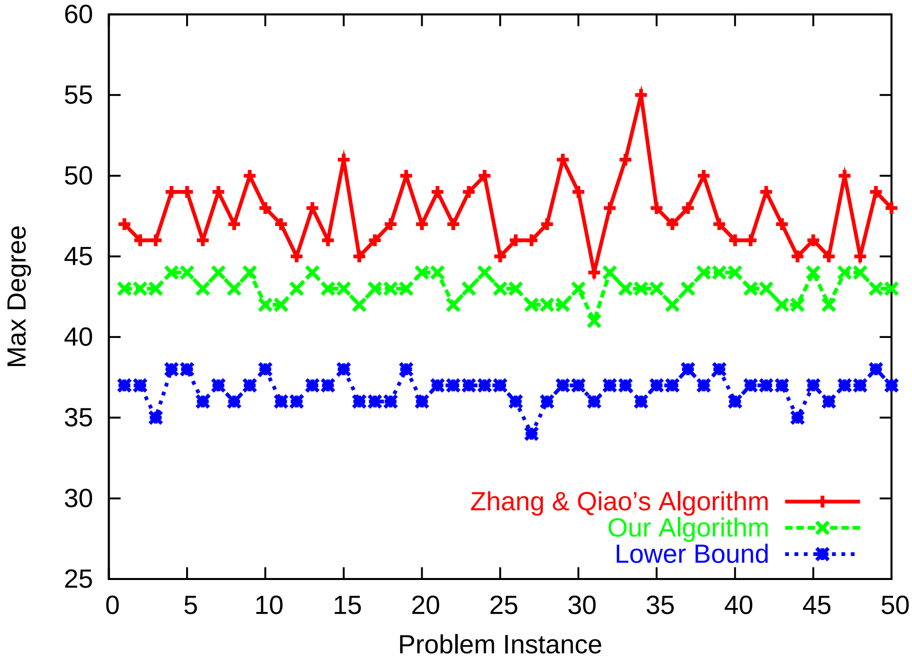


Figure 4.22: Locality Min-Max, $N = 16$, $W = 128$, $C = 12$, $L = 80\%$

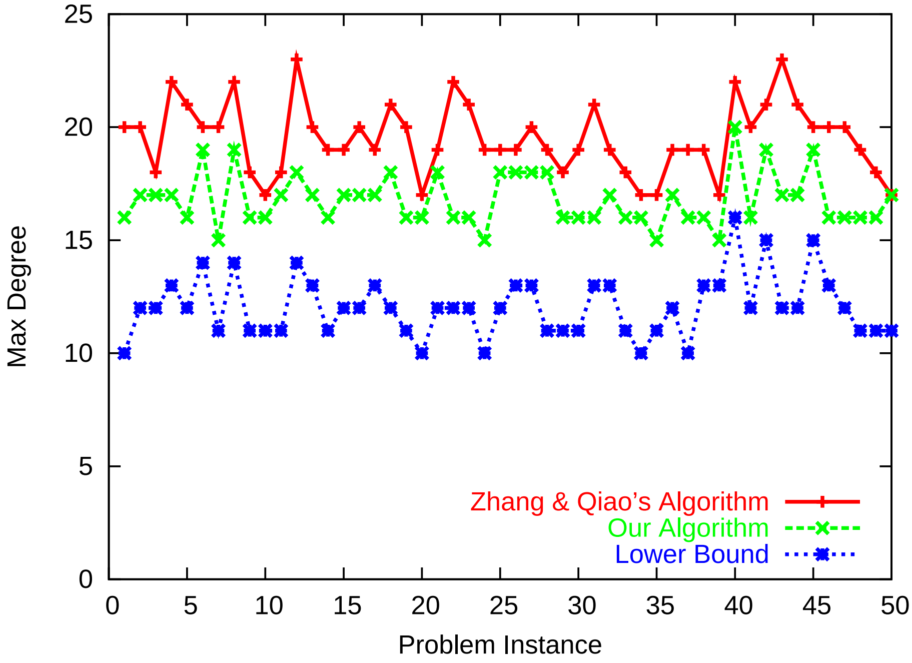


Figure 4.23: Random Min-Max, $N = 16$, $W = 128$, $C = 12$, $L = 40 - 60\%$

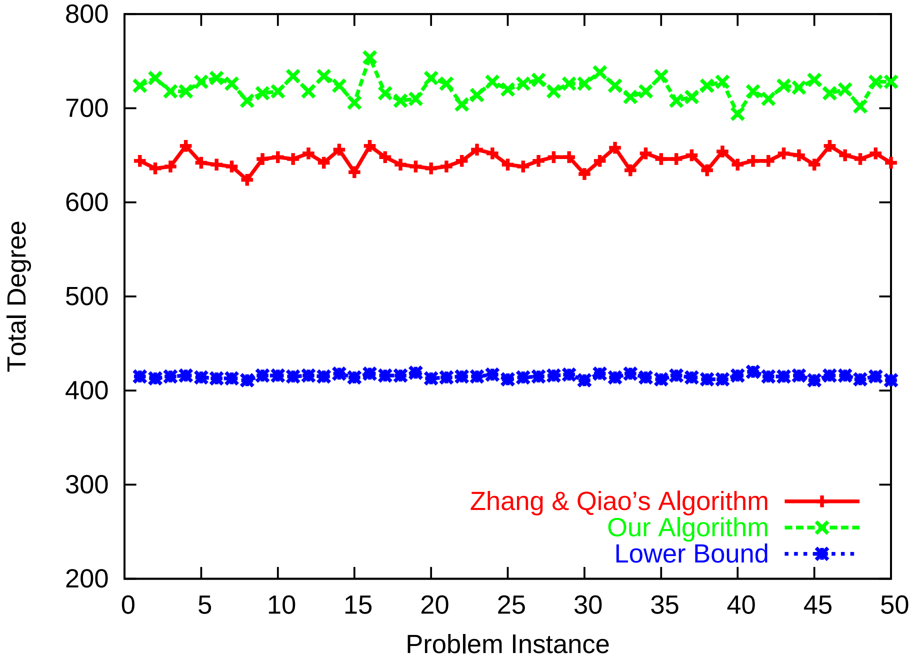


Figure 4.24: Uniform Overall, $N = 16$, $W = 128$, $C = 12$, $L = 80\%$

performing better than the other over some of the problem instances.

Although our algorithm does not aim purely at reducing the Overall degrees in the ring network, results from Figures 4.25 and 4.26 show that for the Locality and Random traffic patterns, our revised algorithm outperforms Zhang and Qiao's algorithm again for the Overall objective. By aiming at both the Min-Max and Overall objectives, our solutions give more options for network designers, provided that a digital cross-connect is equipped at each network node. Considering the savings for the more expensive LTEs needed for all the additional lightpaths (each of which counts as two more degrees), the cost of 16 DXCs in the whole network is almost negligible.

The fact that Zhang and Qiao's algorithm performs well when traffic is uniform is not surprising, due to the nature of the algorithm which attempts to form full circles of unit traffic components which are then groomed into wavelengths; this operation is most successful when traffic demands are symmetric. However, the important observation is that, for the asymmetric traffic scenarios that are more likely to be encountered in practice, our algorithm performs better, not only in terms of the maximum nodal degree, but also in

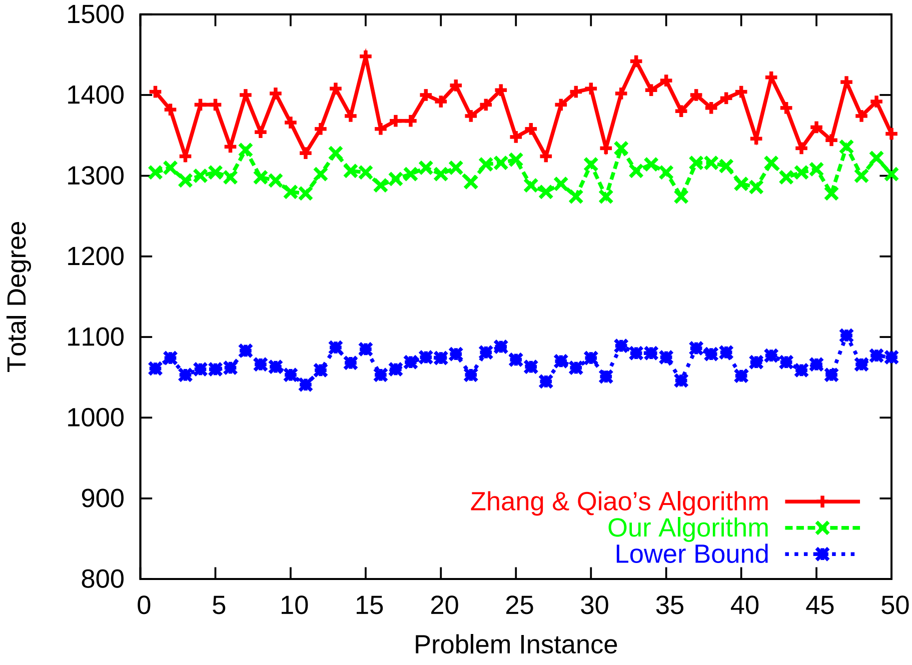


Figure 4.25: Locality Overall, $N = 16$, $W = 128$, $C = 12$, $L = 80\%$

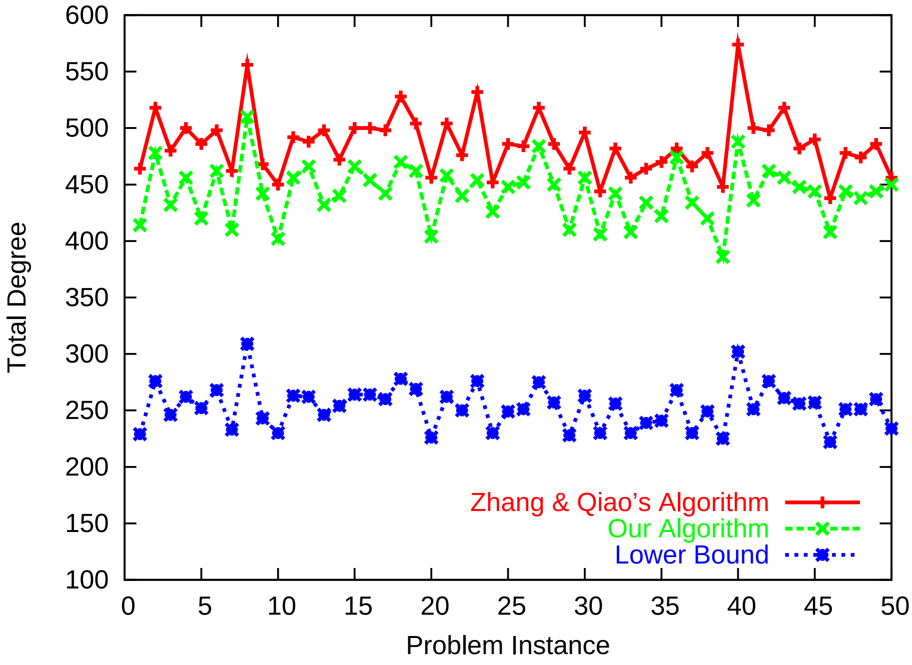


Figure 4.26: Random Overall, $N = 16$, $W = 128$, $C = 12$, $L = 40 - 60\%$

terms of the total degree over all network nodes.

Another advantage of our algorithm is that it *always* returns a feasible solution, if a feasible solution exists. In Zhang and Qiao's algorithm, when the link load exceeds 80% for large rings, there is no guarantee on getting feasible solutions. Moreover, although both algorithms run very fast during the experiments, our algorithm is much faster, because of the smaller stochastic complexity and less requirement on calculating, maintaining and sorting lists of large size.

We will not present comparisons on results from bi-directional rings here. That is because Zhang and Qiao used the same shortest-path routing strategy in their algorithm. Thus, the decomposition of any traffic demand matrix by the two algorithms will be the same, and the results for the Overall objective will simply be the addition of results from two unidirectional rings in both cases. Moreover, since our algorithm for bidirectional rings (see Section 4.4) is adapted especially for the Min-Max objective, it is expected to outperform Zhang and Qiao's algorithm even more if Min-Max optimization is the main concern.

To sum up, the numerical results we presented in this section demonstrate that our min-max optimization approach for traffic grooming in the ring topology is successful in obtaining solutions that keep the maximum nodal degree low. Moreover, our solutions also tend to keep the overall network cost (in terms of the total degree over all network nodes) low, and compare favorably to solutions obtained by algorithms whose main objective is network-wide cost minimization.

Chapter 5

Grooming in the Star Topology

In this chapter, we will study networks with the physical topology of a star, in which several nodes are connected to a single hub directly, but are not connected to each other.

Star networks arise in the interconnection of LANs or MANs with a wide area backbone. Cable TV networks and passive optical networks (PONs) are based on a tree topology, which can be decomposed into stars as well. Also, consider a relatively small optical WDM network with a general topology. If we require that only one of the network nodes has switching ability, the virtual topology formed by a traffic grooming solution will be in the form of a star network. Although direct lightpaths that "pass through" the hub node may not actually pass the OXC at the hub, the virtual topology (by ignoring the physical links) will look just like that in the star topology. Therefore, if we use the star virtual topology as building block, it may be possible to solve larger and more general network topologies by employing a decomposition method. For these reasons, it is worthwhile to study grooming in the star topology in detail.

Similar to what we present in the previous chapter for the ring topology, we first give an ILP formulation for the grooming problem, then give proofs of various complexity results for the star grooming problem.

Heuristic algorithms are given for both the Min-Max and the Overall objectives we consider. Numerical results given by experiments are shown to show the effectiveness of our algorithm. Comparison between the Min-Max and the Overall objective is done to

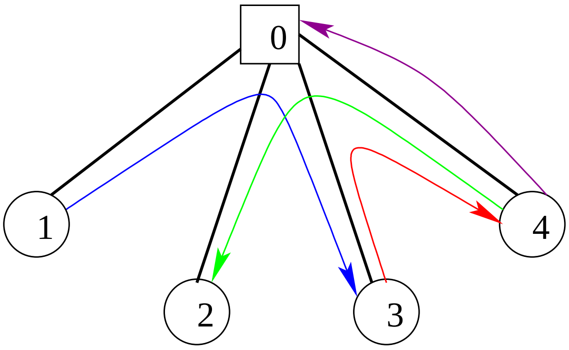


Figure 5.1: A 5-node star with 4 lightpaths

explore the relationship of the two objectives.

5.1 Problem Definition and ILP Formulation for Stars

Again, we first include the ILP formulation for traffic grooming problem in star networks, in order to better understand the problem in this specific topology by showing the differences from the ring or mesh networks. We also use the formulation to get optimal solutions via CPLEX.

Figure 5.1 illustrates a star network with 5 nodes. We always label the central hub node with tag 0, then label the non-hub nodes in sequence. We assume that every physical link is bi-directional, and no traffic demand can traverse the same physical link more than once. This assumption makes the problem simpler, and can ensure efficient use of wavelength capacity. Thus, only the hub node can be the intermediate stop for a traffic component, that is, non-hub nodes are not allowed to switch traffic. This assumption will affect the nonzero variables allowed in the ILP formulation, as can be shown later.

Under the assumptions above, there will be only two kinds of lightpaths in the logical topology. The first kind consists of single-hop lightpaths that either originate at a non-hub node and terminate at the hub node, or vice versa. The second kind consists of two-hop lightpaths that originate and terminate at non-hub nodes, and are switched optically at the hub node.

The ILP formulation for grooming in Star networks is below:

Given:

The physical topology, a star with a hub node 0 and N non-hub nodes numbered $1, \ldots, n$. Each physical link consists of a fiber in each direction.

The traffic demand matrix,

$T = [t^{(sd)}], s, d \in \{0, \cdots, (N-1)\}, t^{(sd)} \in \{0, 1, 2, \cdots\}, t^{(ss)} = 0, \forall s.$

The wavelength limit W, which is the number of distinct wavelengths each link can carry, and **wavelength capacity** C, the number of unit traffic rates each wavelength can carry.

Find:

A grooming solution, in terms of lightpath indicators b_{ij}, lightpath wavelength indicators $c_{ij}^{(k)}$, and traffic routing variables $t_{ij}^{(sd)}$.The 'meaningful' superscrips and subscripts only take one of the following forms:

$$\begin{cases} t_{sd}^{(sd)}, & \text{if } s = 0 \text{ or } d = 0, s \neq d; \\ t_{sd}^{(sd)}, t_{s0}^{(sd)}, t_{0d}^{(sd)}, & \text{if } s \neq 0, d \neq 0 \text{ and } s \neq d. \end{cases}$$

That is, if a variable $t_{ij}^{(sd)}$ doesn't fit into any form above, then its value is 0.

To Minimize: the maximum number of initiating/terminating lightpaths at a node.

$$\max_i \left(\max(\sum_j b_{ji}, \sum_j b_{ij}) \right). \tag{5.1}$$

or, the overall number of degrees in the virtual topology.

$$\sum_{i,j} b_{ij} \tag{5.2}$$

Subject to:

Wavelength Constraint at Each Link:

$$0 \leq \sum_j b_{ij} \leq W, \text{ for } i = 1, \ldots, n \tag{5.3}$$

$$0 \leq \sum_i b_{ij} \leq W, \text{ for } j = 1, \ldots, n \tag{5.4}$$

Traffic Routing Constraints:

$$t_{ij} = \sum_{\forall s,\, d} t_{ij}^{(sd)},\ \forall i, j. \tag{5.5}$$

$$t_{ij} \leq b_{ij}C,\ \forall i, j. \tag{5.6}$$

$$\sum_j t_{ij}^{(sd)} - \sum_j t_{ji}^{(sd)} = \left\{ \begin{array}{ll} t^{(sd)}, & i = s \\ -t^{(sd)}, & i = d \\ 0, & \text{otherwise} \end{array} \right\},\ \forall i, s, d. \tag{5.7}$$

Most of the constraints above are self-explanatory. Because we've already made constraints in $t_{ij}^{(sd)}$, many other constraints are implied from this variable restriction. Moreover, we exclude wavelength assignment variables c and make the ILP formulation look even less complicated, since wavelength assignment can always be done in polynomial time, as long as other wavelength constraints are followed [23].

Despite the simplicity of the ILP formulation, we prove in the next section that the the grooming problem remains intractable.

5.2 Complexity Results for Stars

We present complexity results with proofs for star networks in this section. Please refer to Section 4.2 for definition of the bifurcated and the non-bifurcated cases.

It has been known that the wavelength assignment sub-problem can be solved in polynomial time for star networks. Readers can refer to [23] for related results and algorithms. However, we are able to prove that the whole grooming problem in stars, no matter if bifurcated routing is allowed or not, still remains NP-Complete.

5.2.1 NP-Completeness in the Non-bifurcated Case

Theorem 5.1 *The decision version of the grooming problem in star networks with the Min-Max objective (bifurcated routing of traffic NOT allowed) is NP-complete.*

Proof. We reduce the decision version of the BIN PACKING problem to the non-bifurcated Star problem. An instance of the BIN PACKING problem has a set U of n elements, $u_1, \ldots, u_n$, having weights $w_1, \ldots, w_n \in Z^+$, respectively; a positive integer bin capacity B, and a positive integer K as the decision goal. The problem asks whether the set U can be partitioned into disjoint sets $U_1, U_2, \ldots, U_k$, such that $\sum_{u_i \in U_j} w_i \leq B, j = 1, \ldots, K$. Since the problem is trivial if there exists $w_i > B$, or $\sum_j u_j > B \times K$, or $K > n$, we can eliminate those cases, and the problem remains NP-Complete.

Given such an instance, we construct a star network using the following transformation: the star has $n+1$ non-hub nodes, in which nodes $1, 2, \ldots, n$ are *destination nodes*, and node $n+1$ is called the *source node*; wavelength capacity $C = B$, $W = n + K$, and the decision goal $F = n + K$.

The traffic matrix $t^{(sd)}$ is:

$$t^{(sd)} = \begin{cases} w_d, & s = n+1, d = 1, 2, \ldots, n; \\ nC, & s = n+1, d = 0. \end{cases}$$

Notice that the traffic demands from the source node to the hub node already requires the set up of n lightpaths, making the outdegree of the source node to be n. In order to reach the decision goal, the remaining traffic from $n+1$ to the destination nodes must be 'packed' in no more than K lightpaths, each having capacity $C = B$. Since bifurcated routing is not allowed, and we've assumed $w_i \leq B$, that means each demand w_i can be packed only in one lightpath. Now, whether the degree $O_{n+1} \leq n + K$ depends on whether the BIN PACKING problem can be solved. The objective is big enough to handle degrees elsewhere, so the max degree must be O_{n+1}. Now we can say that the star problem is feasible under $F = n + K$ *if and only if* the corresponding BIN PACKING problem is decidable. Since the BIN PACKING problem is known to be NP-Complete (Please refer to [26]), so is the star grooming problem. ■

The problem restriction allows us to extend the result to the Overall objective as well. Since we require that $W = n + K$, if we use $F' = 6n + 4K$ as the overall goal, we are still forced to use the virtual topology design above, and the same reduction from the BIN PACKING problem can be made.

Like in the proofs from the previous chapter, because of the construction in the proof, we can get the following aggregated result for the non-bifurcated case in the star topology:

Corollary 5.1 *The decision version of the grooming problem in star networks with the Min-Max or the Overall objective (bifurcated routing of traffic NOT allowed) is NP-complete, even when a logical topology is provided.*

Since the Bin Packing problem we reduce from is NP-Complete in the strong sense, we can conclude that the corresponding theory and corollaries also show strong NP-Completeness, meaning that even a pseudo polynomial-time algorithm is not likely to exist.

5.2.2 NP-Completeness in the Bifurcated Case

Now we want to show the same results in the bifurcated case. In Star networks, once a virtual topology is given, Routing and Wavelength Assignment can be done in polynomial time; and the routing of traffic onto the lightpaths is also in P: first, just make best use of the two-hop 'long' lightpaths to carry non-hub demands, then stuff the single-hop demands with the rest of the traffic. Since bifurcated routing is allowed, we can 'groom' the rest traffic in any combination, as long as we make sure to keep using existing capacities in the single-hop wavelength before starting to use a new one.

Therefore, the NP-Completeness exists in the Virtual Topology design sub-problem. We have the following conclusion:

Theorem 5.2 *The decision version of the grooming problem in star networks with the Min-Max objective (bifurcated routing of traffic allowed) is NP-complete.*

Proof. We reduce the decision version of the constrained PARTITION problem to the grooming problem. An instance of the constrained PARTITION problem is given by a finite set A of $2n$ elements. Each element a_i has a weight $w_i \in Z^+$. The problem asks whether there exists a subset $A' \subset A$, such that $|A'| = n$ and $\sum_{a_i \in A'} w_i = \sum_{a_i \in A-A'} w_i$. That is, whether we can partition set A into two parts, each with n elements, and each having exactly half of the total weight of A.

Given such an instance, we construct a star network using the following transformation: $N = 2n+2, W = 2n, C = \sum_{a_i \in A} w_i$, and objective min-max degree $F = 2n$. Nodes

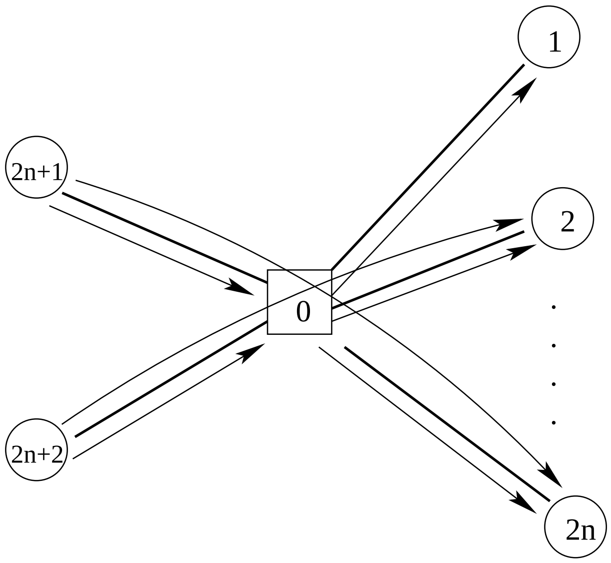


Figure 5.2: The Star Network with a Hub Node 0, Source Nodes $n+1$ and $n+2$, and Destination Nodes $1, 2, \ldots, 2n$. Used for Proof of Theorem 5.2

$2n+1$ and $2n+2$ are *source nodes*, nodes $1 \ldots 2n$ are *destination nodes.* Hub node 0 is the center of the star, has both arriving and departing traffic, and is the only node that does switching, optically or electronically.

The traffic matrix $t^{(sd)}$ is:

$$t^{(sd)} = \begin{cases} C - w_d, & s = 0, d = 1, 2, \ldots, 2n; \\ w_d, & s = 2n+1 \text{ or } 2n+2, d = 1, 2, \ldots, 2n; \\ (n - 1/2)C, & s = 2n+1 \text{ or } 2n+2, d = 0. \end{cases}$$

Figure 5.2 is the graph that shows the star network.

Notice that since the objective degree $F = 2n$, the amount of traffic demand from the hub node 0 to the destination nodes $1 \ldots 2n$ already requires outdegree $O_0 \geq 2n$. Therefore, the traffic demand from the two source nodes $2n+1, 2n+2$ to the destination nodes must *either* use the capacities left on the existing $2n$ lightpaths from node 0 to $1 \ldots 2n$, *or* set up two-hop lightpaths that bypass node 0. The capacities left on each of the $2n$ Lightpaths are so set up that any one, but not both, of the traffic components from the source nodes *has to* be routed optically for each destination node.

Now we examine the indegree of node 0 and the outdegrees of source nodes $2n+1$ and $2n+2$. The demand from source node $2n+1$ to node 0 requires set up of n single-hop lightpaths, so is the demand from source node $2n+2$ to node 0. This already makes the indegree of node 0, $I_0 = 2n$, and $O_{2n+1} = O_{2n+2} = n$. As we saw from previous analysis, we still need to set up $2n$ lightpaths that bypass node 0, either from node $2n+1$ or $2n+2$, for the $2n$ destination nodes accordingly. With $F = 2n$, this means that n of the lightpaths need to be from node $2n+1$ to some n destination nodes, and another n lightpaths should be from node $2n+2$ to the remaining n destinations. This is the same as *partitioning* set A of $2n$ elements into two sets, A' and $A - A'$, each with n elements.

Therefore, we need only to consider candidate solutions described above, which there is a two-hop lightpath from exactly one of nodes $2n+1, 2n+2$ to each destination node $1, 2, \ldots, 2n$. Then, whether the virtual topology is feasible depends on whether the remaining capacities of the single-hop lightpaths, $(2n+1, 0)$ and $(2n+2, 0)$, can carry the traffic that needs to be electronically switched at the hub node. We have set up the traffic demands so that the capacity left for both $(2n+1, 0)$ and $(2n+2, 0)$ is $C/2 = (\sum_{a_i \in A} w_i)/2$. As a consequence, grooming can be done *if and only if* there is a PARTITION of the $2n$ destination nodes each with n elements, and their total weights are equal. Since this constrained PARTITION problem is known to be NP-Complete [26], so is the star grooming problem. ■

Again, if we use the above construction and set the Overall goal $F' = 12n$, we are still forced to use the above virtual topology, thus the same reduction can be used for proof on the Overall objective. Now we can conclude results on the star bifurcated case with the following corollary:

Corollary 5.2 *The decision version of the grooming problem in star networks with the Overall objective (bifurcated routing of traffic allowed) is NP-complete.*

Note that we cannot make similar conclusions on the NP-completeness for the logical topology design part as in the non-bifurcated case, since the trick in the proof is to find the appropriate virtual topology. As we have analyzed at the beginning of this subsection, if a logical topology is given, optimal grooming could be done in polynomial time.

We do not further study on the approximability of the star grooming problems, nor on whether the bifurcated version is strong NP-Complete. Instead, for practical purposes, we concentrate on studying the bifurcated version of grooming in stars, by designing polynomial-time algorithms that give very good results in the following section.

5.3 Heuristic Grooming Algorithm for Stars

As in the ring grooming problems, for practical purposes a polynomial-time algorithm with good performance is needed for star networks of large size, preferably one that provides near-optimal grooming solutions. In this section, we present polynomial-time algorithms for star networks with either the Min-Max or the Overall objective. We focus only on the case where bifurcated routing is allowed, the reason similar to that in the ring case: bifurcated routing is more flexible, allows better utilization of network resources and may be preferred in some situations because of security concerns.

In this section, we give star grooming heuristic algorithms for the Min-Max and the Overall objectives, respectively.

5.3.1 Star Grooming for the Min-Max Objective

As we have mentioned before, wavelength assignment can be done in polynomial time, so we will concentrate on the virtual topology and routing problem in our heuristic algorithm.

The detailed algorithm is described in Figure 5.3.

Please refer to Section 4.3.1 for the definition and procedure of the *Reduction* and *Initialization* in the description of algorithm. After reduction, we obtain an initial solution by applying the all-electronic approach, similar to what we discuss in the ring grooming algorithm. In this manner, traffic is packed as tightly as possible onto lightpaths that traverse only one physical link. If, after this step, the most congested link (the physical fiber that carries the largest number of lightpaths) has more than W lightpaths, we can conclude that no feasible solution exists.

The following step, checking feasibility, is straightforward: determine if the all-electronic solution violates any wavelength constraint. Note that after getting the all-electronic solution, we have reached the lower bounds for the degrees at non-hub nodes. Since we are considering the Min-Max objective, if the hub degree is even smaller than one of the non-hub degrees, we have already reached the optimal; otherwise, we need to lower the hub degree without increasing our Min-Max objective. To this end, we use a greedy heuristic which, at each step picks the largest traffic component that has not been considered yet, and attempts to optically route it, if doing so will reduce the hub degree

Min-Max Traffic Grooming Algorithm for Star Networks
Input: A star network with N non-hub nodes and a hub 0, W wavelengths, capacity C of each wavelength, and traffic matrix $T = [t^{(sd)}]$.
Output: The number of lightpaths b_{ij} from node i to node j of the star, and traffic routing quantities $t_{ij}^{(sd)}$, so that the solution is feasible and has a small value of the min-max degree objective.

```
procedure StarGrooming
  1. Reduce the traffic matrix by assigning direct lightpaths of capacity $C$ fully used
  2. Use single-hop lightpaths to carry the remaining two-hop traffic
  3. Check feasibility. If infeasible, exit
  4. Initialize $b_{ij}, t_{ij}^{(sd)}$, record indegree $I_j$, outdegree $O_j$ and remaining capacity
     $r_{ij}$ on lightpath $(i, j)$ in the current topology
  5. $u \leftarrow$max degree of the non-hub nodes
  6. while $max\ hub\ degree > u$ do
  7.     Sort all the two hop residual $t^{(sd)}$ in non-incresing order
  8.     for each of the sorted $t^{(sd)}$ do
  9.         if carrying the traffic directly doesn't increase $u$ then
                 re-route the traffic on direct lightpaths, update all variables accordingly
  10.    endfor
  11.    if $u < w$ and $max\ hub\ degree > u$ then $u++$
  12.    else break; endif
  13. endwhile
  14. Use the polynomial-time WLA algorithm in [23] to assign wavelengths to the lightpaths
end Procedure;
```

Figure 5.3: Traffic Grooming Algorithm for Star Networks with the Min-Max Objective

without increasing the maximum degree. Following this procedure, we may reach a point where the maximum degree is at one or more non-hub nodes, possibly jointly with the hub node, in which case the algorithm terminates. Otherwise the hub node still has a degree greater than that of any non-hub node. If it would not violate the wavelength limit at any non-hub node, we then decrease our Min-Max target by one and repeat the procedure.

It is also straightforward to obtain the complexity of this algorithm: the reduction and initial feasible solution takes time $O(N^2W)$; the *while* loop between Steps 6-13 is executed no more than W times, since a feasible solution requires $u \leq W$; and there are no more than $N(N-1)$ traffic elements to consider within the loop. Thus, the overall complexity of the algorithm is $O(N^2W)$.

5.3.2 Star Grooming for the Overall Objective

One of the main goals of this study was to understand the relationship between the Overall and the Min-Max objectives. Accordingly, we were interested in knowing whether the Min-Max heuristic presented above could be used with little or no modification to perform well for the Overall objective as well. We found that a different approach is needed to address the Overall objective, though there is some commonality. In this section, we present our development of the algorithm for and insight into the Overall objective; later in Section 5.4, we present data regarding the performance of these algorithms not only for the objective each was designed for, but also for the other objective.

First, we argue that the Overall results given by our Min-Max heuristic in Figure 5.3 can hardly be improved without violating the Min-Max objective. This is because we aimed at the Min-Max objective alone in the algorithm. As we will see in Section 5.4, the Min-Max algorithm performs very close to the optimal: for stars with 10 non-hub nodes, only 1 out of the 50 cases in our experiments results in a non-optimal solution. However, this is accomplished at the expense of neglecting the Overall objective. Let us consider the following approaches to improve the value of the Overall objective.

(1) After the Min-Max heuristic terminates, continue rerouting two-hop traffic onto direct lightpaths. This might improve the nodal degree at the hub, but cannot be done without violating the existing Min-Max solution, because at the last iteration of the *while* loop at Step 6-13, we have rerouted all such traffic that will not violate the current objective u in non-increasing order (refer to the *if* statement at Step 9).

(2) Break some direct lightpaths back into two at the hub node. This can only be done by examining the last iteration of the *while* loop step by step, considering the candidate traffic in non-decreasing order (because smaller traffic are more likely to fit into remaining capacities). Moreover, we need to ensure that (a) the hub degree cannot increase beyond u, and (b) the non-hub degree cannot increase beyond u. Note that at the previous iteration, we have reached the limit that the non-hub degrees cannot increase beyond $u-1$, so there is little room left for improvement.

Finding a good combination of the above two approaches leads to combinatorial explosion. Therefore, if we want to balance the two objectives, new approaches are needed. To find a good algorithm, we use a Integer Linear Programming (ILP) formulation for the star grooming problem. A more general (and more straightforward) ILP can be found in [22], but here we give a simplified version that has only binary (0-1) variables. The formulation allows us to relax the constraints to allow for traffic demands between each source/destination pair that exceed the wavelength capacity C.

In this context, we need to distinguish between the problem parameters as given by the instance, and after some preprocessing. First, we perform the traffic matrix reduction (as described above in Section 5.3.1), then we set up the minimum number of single hop lightpaths to (and from) the hub node from (and to) each non-hub node which are required to carry traffic components that terminate (and originate) at the hub (that is, traffic components of the form $t^{(s0)}$ and $t^{(0d)}$). We define the following notation for quantities after this preprocessing:

$t_r^{(sd)}$: the traffic demand ($0 \leq t_r^{(sd)} < C$) from s to d, $s \neq d \neq 0$. We further define $t_{out}^s = \sum_d t_r^{(sd)}, t_{in}^d = \sum_s t_r^{(sd)}$.

r_{out}^s: the remaining capacity left on the (possibly) underutilized lightpath $(s, 0)$. r_{in}^d is defined similarly.

w_{out}^s: the full wavelengths available on link $(s, 0)$. w_{in}^d is defined similarly.

We need to find $x^{(sd)} \in \{0, 1\}$, in which 0 denotes electronic routing of remaining demand $t_r^{(sd)}$, while 1 denotes optical routing (setting up a two-hop lightpath dedicated to it).

Therefore, our goal to minimize the total number of lightpaths can be expressed as follows; note that we do not count the lightpaths set up during reduction, because they are necessary and we have no choice but to keep them.

Minimize:

$$\begin{aligned} & \sum_s \left\lceil \frac{t_{out}^s - r_{out}^s - \sum_d t_r^{(sd)} x^{(sd)}}{C} \right\rceil \\ + & \sum_d \left\lceil \frac{t_{in}^d - r_{in}^d - \sum_s t_r^{(sd)} x^{(sd)}}{C} \right\rceil + \sum_{s,d} x^{(sd)} \end{aligned} \tag{5.8}$$

Subject to (link capacity constraints):

$$t_{out}^s - r_{out}^s - \sum_d t_r^{(sd)} x^{(sd)} + C \sum_d x^{(sd)} \leq C w_{out}^s, \forall s \tag{5.9}$$

$$t_{in}^d - r_{in}^d - \sum_s t_r^{(sd)} x^{(sd)} + C \sum_s x^{(sd)} \leq C w_{in}^d, \forall d \tag{5.10}$$

Note that, except for $x^{(sd)} \in \{0, 1\}$, all other values can be calculated from the traffic matrix very easily, and really count as parameters. Therefore, the total number of variables is at most $N(N-1)$. Each variable is binary, thus the solution space has $2^{N(N-1)}$ combinations (or less if some traffic components are zero).

If we relax the ceiling operations in the objective, we can get a new formulation that inspires a greedy algorithm, as we show below. The new constraints are (link capacity constraints):

$$\sum_d (C - t_r^{(sd)})x^{(sd)} \leq Cw_{out}^s - t_{out}^s + r_{out}^s, \forall s \tag{5.11}$$

$$\sum_s (C - t_r^{(sd)})x^{(sd)} \leq Cw_{in}^d - t_{in}^d + r_{in}^d, \forall d \tag{5.12}$$

The goal is now to minimize:

$$\begin{aligned} &\sum_{s,d} (C - 2t_r^{(sd)})x^{(sd)} \\ &+ \left(\sum_s (t_{out}^s - r_{out}^s) + \sum_d (t_{in}^d - r_{in}^d) \right) \end{aligned} \tag{5.13}$$

where the latter part is a constant. The ILP formulation resembles the *Multi-constraint 0-1 Knapsack Problem* (MKP), also called Multi-Dimensional 0-1 Knapsack Problem (MDKP). However, it has special forms that characterize the star grooming problem, so better approaches can be taken to get near-optimal solutions.

We can now get insight into the development of a good heuristic by using the 0-1 ILP formulation for the problem. From the goal, we know that we should try to route all traffic demands that are greater than $C/2$ optically, so that the values $(C - 2t_r^{(sd)})x^{(sd)}$ are negative, and will decrease the goal. However, we should guard against the constraints as well. Therefore, our intention is to greedily route the traffic demands that are more than $C/2$ optically, while making sure not to violate the constraints.

Note that minimizing the quantity $\sum_{s,d} (C - 2t_r^{(sd)})x^{(sd)}$ is the same as maximizing $\sum_{s,d} (2t_r^{(sd)} - C)x^{(sd)}$. Hence, for the relaxed objective, at each iteration, we are closer to our goal by $2t_r^{(sd)} - C$ by routing $t_r^{(sd)}$ directly onto one lightpath. However, for the original objective with the ceiling operations, the performance of each iteration will depend on the remaining capacities on the two corresponding lightpaths $(s, 0)$ and $(0, d)$. For instance, even if a $t_r^{(sd)} < C/2$, when both lightpaths have remaining capacity $\geq C - t_r^{(sd)}$, routing the

Logical Topology Algorithm for Star Networks
Input: A star network with N nodes, W wavelengths, capacity C of each wavelength, and traffic matrix $T = [t^{(sd)}]$.
Output: The set of lightpaths R in the logical topology such that $|R|$ is minimized; **or** failure if no feasible solution exists.

```
procedure StarTopology
begin
 1.  Reduce the traffic matrix $T$, and record the residual
     traffic matrix $T_r = [t_r^{(sd)}], t_r^{(sd)} < C \; \forall \; s, d$
 2.  Create single-hop lightpaths to carry the residual traffic
     by electronically switching (grooming) it at the hub
 3.  Check feasibility; if infeasible, exit with failure
 4.  $U_0 \leftarrow$ number of lightpaths in current logical topology
 5.  Sort all the residual traffic demands $t_r^{(sd)}$ between non-hub
     nodes $s$ and $d$ in non-increasing order, and label them as
     $t_1, t_2, \ldots, t_k, k = (N-1)^2$
 6.  $i \leftarrow 1$; // iteration index
 7.  while $t_i > 0$ do
 8.      Create a new two-hop lightpath to route $t_i$ directly
         from source to destination, if doing so does not
         violate any wavelength constraints
 9.      $U_i \leftarrow$ number of lightpaths in new logical topology
 10.     $i \leftarrow i + 1$
 11. end while
 12. Find the smallest of $U_0, U_1, \ldots, U_m$, and return the
     corresponding logical topology $R$ as the solution
end
```

Figure 5.4: Traffic Grooming Algorithm for Star Networks with the Overall Objective

traffic directly on a single lightpath will decrease the total degree by 2; accordingly, some traffic greater than $C/2$ may actually introduce a new lightpath without eliminating either of the two one-hop lightpaths, adding penalty to the objective. Therefore, we can make potential improvement by trying to route more traffic directly which does not exceed $C/2$, and if the results are better, we should accept the better solution.

This observation leads to the essentially greedy algorithm presented in Figure 5.4. The algorithm is similar to the one presented in [18], and that in [13] as well, although those algorithms were designed to minimize an electronic routing (not nodal degree) objective. Note the important difference that in our algorithm, while the order of considering traffic elements is essentially greedy, we continue until all traffic elements have been considered, and then pick the best one. This means that in case a succession of greedy steps produces first an increase, followed by a larger decrease, we shall be able to pick the best case; further, if several steps produce the same objective value, we shall be able to pick any of them. The importance of this ability will become clear in the discussion of Section 5.4.3.

After the routine reduction step, the residual traffic demands to be groomed are less than the wavelength capacity C, for each source-destination pair. Applying the method we use on ring networks, we obtain an initial solution by first carrying all such demands on single-hop lightpaths to the hub, electronically grooming them there, and then carrying them on single-hop paths to their respective destinations. The same approach has been used on the Min-Max algorithm.

Like in the Min-Max case, the all-electronic solution for the reduced traffic matrix is not expected to be optimal with respect to minimizing the number of lightpaths. Intuitively, it would be possible to re-route traffic demands between non-hub nodes onto direct lightpaths that bypass the hub node, to create longer (two-hop) lightpaths; doing so is desirable if the creation of a two-hop lightpath leads to the elimination of two single-hop lightpaths, decreasing the total number of lightpaths. However, if such direct lightpaths carry only a small amount of traffic compared with the wavelength capacity C, this approach may not lead to a better solution. Although finding the optimal set of non-hub demands for which to set up direct lightpaths is NP-hard (since the star grooming problem is NP-hard), intuition suggests that a greedy approach of assigning lightpaths to the largest traffic demands will work well in practice.

Steps 5-11 of the algorithm perform the greedy assignment of lightpaths. At each iteration, we check whether creating a direct two-hop lightpath for the largest traffic component currently routed over two single-hop lightpaths would violate the wavelength constraint W. If so, we do nothing; otherwise, we create the new two-hop lightpath and remove any single-hop lightpaths for which this was the only traffic component they carried. We continue in this manner, recording the total number of lightpaths after every iteration, until no additional two-hop lightpaths can be created. Among all the logical topologies created at the end of each iteration, the algorithm returns the one with the smallest number of lightpaths as the solution.

It is straightforward to see that the time complexity of the star grooming algorithm is $O(N^2 \log N)$, since there are no more than N^2 residual elements to be considered after the reduction step, whose sorting takes $O(N^2 \log N)$ time, to be examined sequentially in the algorithm.

5.4 Numerical Results for Stars

In this section, we present our experimental results with stars for the Min-Max objective. Since the star topology is symmetric over the central hub, and the traffic demands are at most two-hop, it is hard to define complicated traffic patterns as in the ring topology. Instead, we generate the traffic demand matrix using the Random and Uniform traffic patterns as in the ring topology. Please refer to Section 4.6 for a detailed description of such traffic patterns.

We present our experimental results for the Min-Max and the Overall objectives respectively, and then examine the relationships of the two objectives, as well as how they evolve in the iteration steps.

5.4.1 Results for the Min-Max Objective

Using CPLEX with the ILP formulation described in Section 5.1, we are able to obtain results for star grooming instances from the Random traffic pattern with the Min-Max objective. Using Sun UltraSparc system, we can get optimal results For stars with $N = 16$ non-hub nodes. For stars with $N = 24$ non-hub nodes, in about one out of ten cases we cannot get the optimal after several hours. Since stars with 16 non-hub nodes are already considered large, we will study stars with the two sizes, and discard the cases that we cannot get optimal solutions in a few hours.

After getting the initial all-electronic feasible solution, we can calculate a lower bound (the max degree at non-hub nodes) and an upper bound (usually the max degree at the hub node) for our Min-Max objective. The upper bounds are too big to be plotted, so only a range is given in general.

We collect data for 50 cases with $N = 16$ and 24, respectively. Similar to what we have done in Section 4.6.4 for bidirectional rings, a *grooming effectiveness* is used as evaluation scale, which is defined as the objective divided by the all-electronic result (the hub node degree for this case) . It turns out that the heuristic algorithm works even better than we have predicted. For $N = 16$, 4 of the 50 cases give non-optimal result; for $N = 24$, 18 of them are not optimal.

Figures 5.5 and 5.6 show the results respectively. We find that all non-optimal results are very close to the optimal. We also notice that the grooming effectiveness is

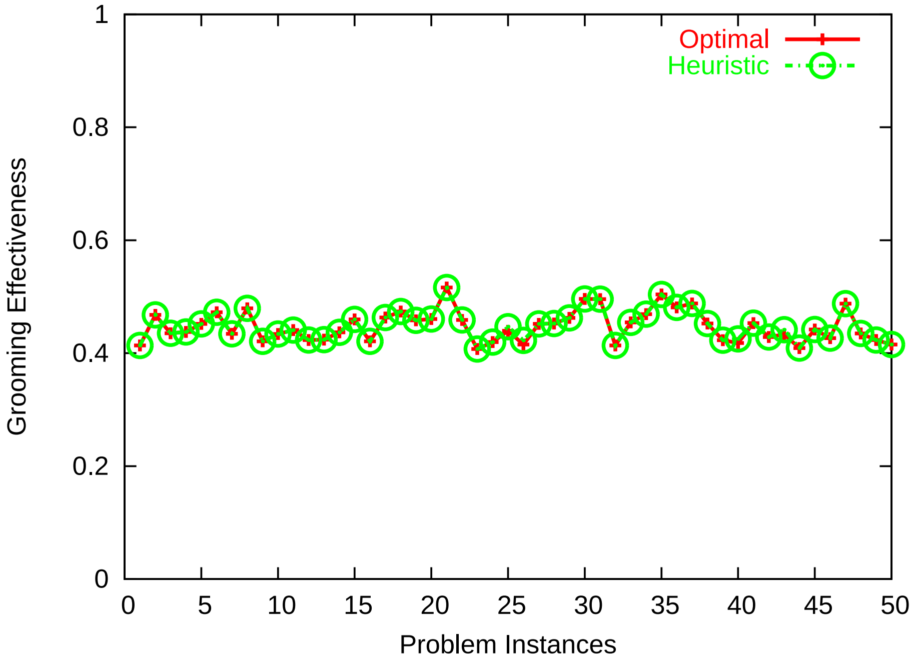


Figure 5.5: Star Random, $N = 16, C = 12, W = 64$, All-electronic = 119-150

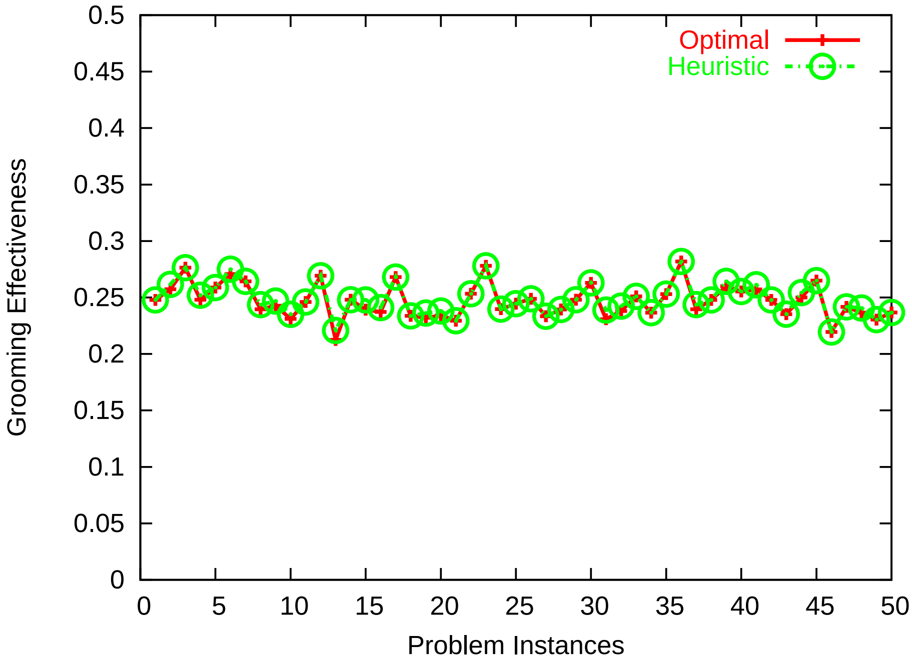


Figure 5.6: Star Random, $N = 24, C = 12, W = 64$, All-electronic = 225-259

smaller(better) when the star size grows. This result suggests that grooming techniques are more helpful in larger networks, even if we restrict the network to the simple topology of stars.

5.4.2 Results for the Overall Objective

The Overall objective algorithm, after applying all the constraints, also works quite well for stars with 10 non-hub nodes. When the star size grows to 16, many instances we generate take more than a few hours to solve with CPLEX, and we are unable to obtain bulk results for the optimal in these cases. The plot in Figure 5.7 shows results for 10-node stars. The typical grooming effectiveness is just under 60%, which translates to an average of 446 lightpaths for each case, and our algorithm gives results with only 2 to 4 more lightpaths. The average difference is 2.96, which is less than one percent from the optimal values.

5.4.3 Comparison between the Min-Max and Overall Objectives

From the nature of strong symmetry in stars, we have the intuition that for the star network topology, the Min-Max and the Overall objective are closely related to each other. That is, by getting the Min-Max solution of the problem, the resulting overall degrees are also close to the Overall objective, and vice versa. This is the purpose of our study in this section.

We define *symmetric degree* solution as the one in which the LTEs used at each node to add/drop traffic are designed symmetrically, with each node having the same capability to add traffic as that to drop traffic. Correspondingly, we define the *Overall Symmetric* objective as $\sum_j Max(b_{ij}, b_{ji})$.

For $N = 10$, we are able to get the optimal solution for the minimum *overall* degrees (defined as $\sum_{i,j} b_{ij}$ in the ILP in Section 5.1), as well as the minimum *symmetric* degrees.

Again, for $N = 16$, we cannot use CPLEX to get the exact solutions. However, CPLEX has an option of calculating approximate solutions more quickly, and we can obtain some results within a few hours for many of the instances. For the minimum *overall* degree, we obtain solutions that are no more than 3% higher than the optimal; for the minimum *symmetric* degree, the solutions are within 6% tolerance. With those numbers, we can

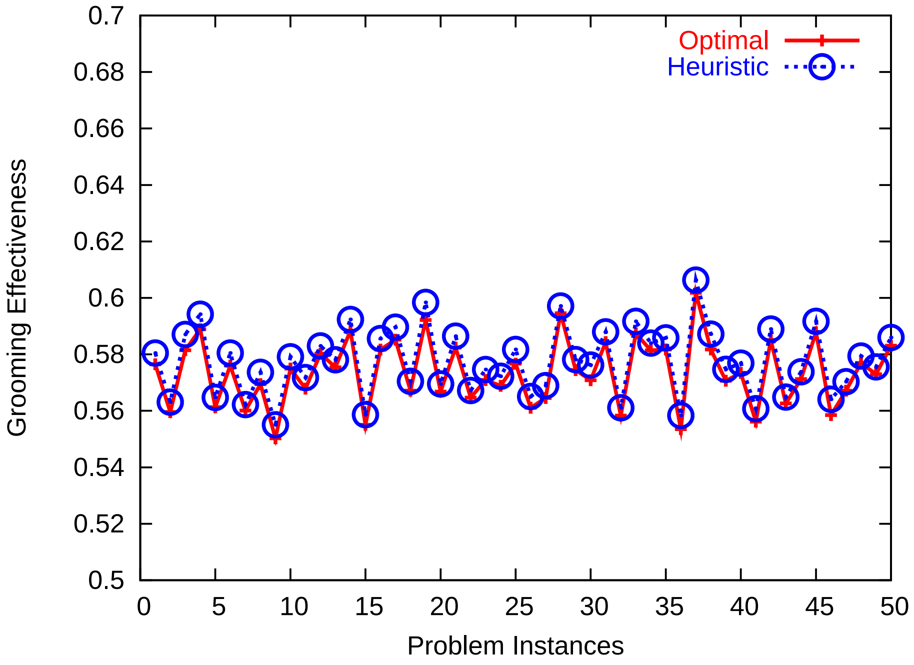


Figure 5.7: Performance of Star Overall Algorithm, $N = 10$

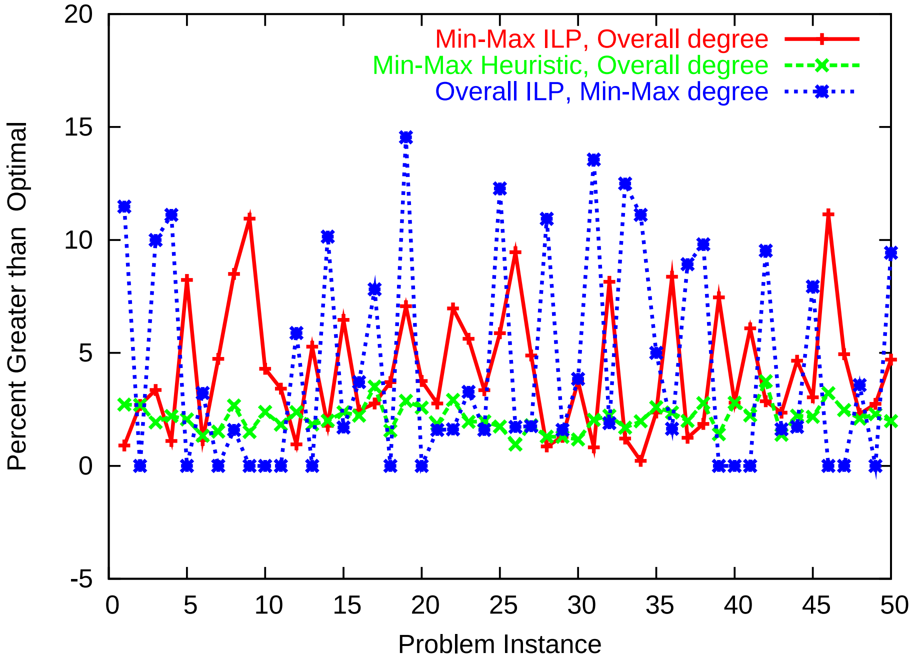


Figure 5.8: Star Random, $N = 10$, Min-Max vs. Overall Degrees (403-528)

calculate a lower bound for the optimal, and plot the percentage of degrees more than the *overall lower bound* and/or the *symmetric lower bound*.

Figures 5.8 and 5.9 show how our solutions for the Min-Max objective perform with respect to the Overall objective. We can see that for stars of size $N = 10$, our Min-Max heuristic algorithm gives 5% or less higher number of degrees than the overall optimal solutions. For $N = 16$, although we can only get the lower bounds for the optimal overall degrees, we can see that there is only around 10% increase of the lower bound for the solutions from our Min-Max objective. These results suggest that, although our objective is the Min-Max degree, the performance of our results on the overall and symmetric degree is also acceptable.

In Figure 5.8, there are three separate curves, and they all compare the result obtained for some particular objective by a given algorithm versus the optimal value of *that objective*, for individual problem instances. In all cases, the optimal values of the two objectives were obtained by solving the ILPs exactly. Thus, the vertical axis represents the percentage by which the output of a particular algorithm exceeds the optimal value. For example, considering the first three points on the dotted blue line with data points marked by asterisks ('*'), we see that the maximum degree at a node in the solution obtained by solving the ILP for the Overall objective was 12% more than the minimum possible maximum degree at a node (which was obtained by solving the ILP for the Min-Max objective) for the first problem instance; for the second instance, the maximum degree in the optimal solution of the Overall ILP was the same as the optimal maximum in/out degree, and for the third instance it was 10% greater.

The first line (with plus '+' signs) plots the value of the overall degree in the optimal solution to the Min-Max ILP and the second line (with cross 'X' signs) plots the value of the overall degree in the solution returned by the Min-Max heuristic algorithm of Figure 5.3. The first obvious observation is that the Min-Max heuristic algorithm has very good performance with respect to the Overall objective, exceeding the optimal by only 3-4%. It is interesting to notice that with respect to the Overall objective, the Min-Max heuristic algorithm gives better results than the Min-Max optimal solutions given by CPLEX. This is not surprising because CPLEX simply returns the first optimal solution from its branch-and-bound search of the solution space, and the results may have many high-degree nodes due to the search sequence. For the same reason, we also note the important fact that the *variability* is much less for the Min-Max heuristic, thus the Min-Max heuristic minimizes the overall degree much more *consistently*.

Now we do a reverse comparison for the 10-node case. The third line in Figure 5.8 (with asterisk '*' signs) plots the value of the maximum degree at a node in the optimal solution to the Overall ILP. It shows that while we try to minimize the overall degree, the maximum degree from the solution is also not far from the optimal Min-Max objective in star networks. However, it is quite inconsistent: many of the values are close to optimal while many are between 10% and 15% more. We can understand this by recalling that often a reduction in the overall degree can be obtained at the cost of increasing the degree of the hub node and reducing the degrees at the non-hub nodes. In the next section, we further investigate the behavior of the Min-Max objective as the overall degree is sought to be minimized.

For the 16-node case, the results are similar for the reverse comparison. To make the figure easier to read, we do not include the third line in Figure 5.9, but separately in Figure 5.10.

It is interesting to notice that with respect to the Overall objective, the Min-Max

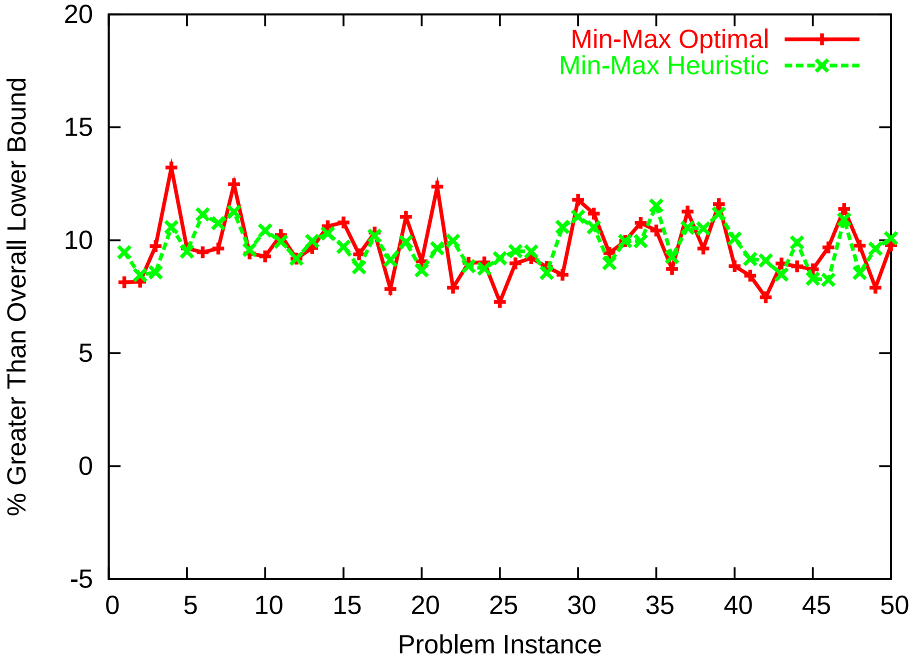


Figure 5.9: Star Random, $N = 16$, Min-Max vs. Overall Degrees (613-778)

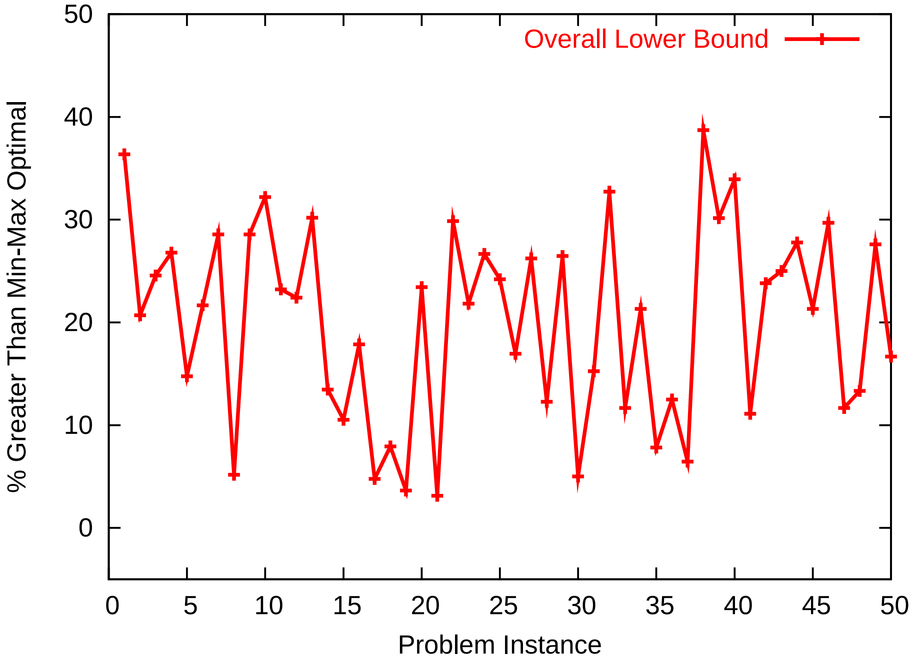


Figure 5.10: Star Random, $N = 16$, Overall Degrees vs. Min-Max (52-68)

heuristic algorithm gives even better results than the Min-Max optimal solutions given by CPLEX. This might be because CPLEX just return the first optimal solution from its branching-and-bound search of the solution space, and the results may have many high-degree nodes due to the search sequence.

Figures 5.11 and 5.12 give the similar comparison over the Symmetric Overall degree objective. For this new objective, the results from Min-Max solutions seem to be even closer to the optimal.

5.4.4 Evolution of the Two Objectives

In our heuristic algorithm for the Overall objective, a simple greedy approach is used without considering the Min-Max objective. In this section, we study the behavior of the two objectives as the algorithm proceeds from the all-electronic to the direction of all-optical solutions, and present the results in Figures 5.13-5.15.

We record the value for both objectives at each iteration, and find the trends from all-electronic to all-optical solutions to study the trend on both objectives. If our goal is to find good solutions that are not far away from either of the two objectives, we can analyze the two figures showing the steps from the Overall algorithm, and find a point close to the *trough* of both.

Initially, to allow us to focus on this evolution, we ignore the constraint of the wavelength limit on a link. A star of size $N = 10$ means that the number of outgoing lightpaths from a node s has a variation of at most 9 between the all-electronic and all all-optical solutions. Therefore, the wavelength limit constraint does not come into play for instances with low traffic load and uniform pattern. For the Random traffic pattern, statistically, a few cases may happen to reach the limits, but the chances are low as long as the link loads are low. On the other hand, when many links have very high loads concurrently (that is when the load is uniformly high), the wavelength limit realistically constrains feasible solutions.

From our experiments, we find that the Min-Max objective will generally go down and then later up during the transformation from all-electronic to all-optical solutions. The Overall objective has the same general trend, but may "thrash" up and down less smoothly. This justifies our strategy of picking the best value from the results of all iterations, rather than stopping when the Overall objective goes up for the first time.

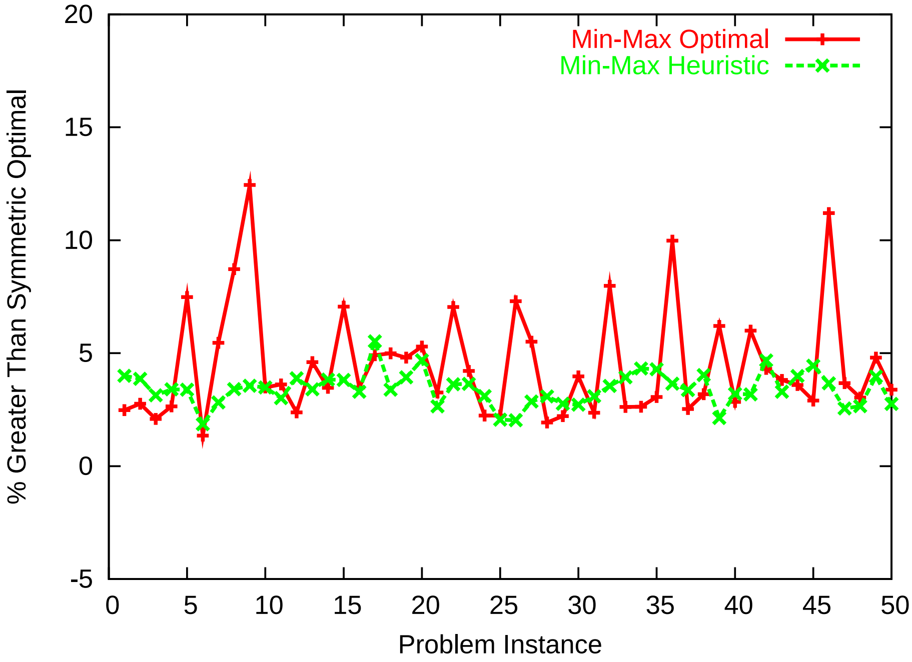


Figure 5.11: Star Random, $N = 10$, Min-Max vs. Symmetric Degrees (451-592)

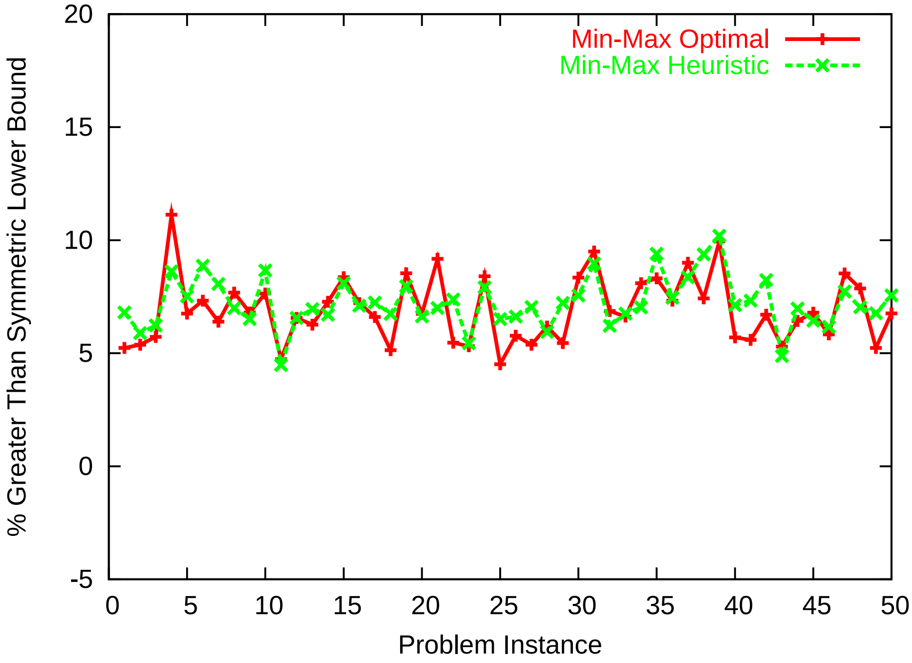


Figure 5.12: Star Random, $N = 16$, Min-Max vs. Symmetric Degrees (691-850)

Figures 5.13 and 5.14 present results for instances with *random* traffic pattern and low loads (in the range of 50%) and *uniform* traffic pattern with high load (95%) respectively. The horizontal axis represents the number of traffic elements that are routed optically instead of electronically, and the vertical axes represent the actual objective values. For the first instance, both objectives simultaneously reach their best values at around the 40^{th} iteration. However, for the instance of Figure 5.14, the Overall objective goes up well before the Min-Max objective reaches its best value.

Finally, Figure 5.15 shows the effect of the wavelength limit. It is the wavelength limit constrained version of Figure 5.14. (For the case of Figure 5.13, the constraints do not make a difference in the output; this is expected at low loads, as we mentioned above.) In this figure, if rerouting the traffic optically at certain iteration would result in wavelength violation, we skip the element and leave the results for corresponding iteration blank. For instance, there is a big gap between Iterations 78 and 84, which means that the elements considered in those iterations will generate infeasible solution if we attempt to route them optically. We find that the effect of the constraints is a dampening of the objective curves, but the general behavior remains the same. On the whole, the removal of the constraints (at high traffic loads) produce effects similar to those produced by reducing the traffic load somewhat. We have obtained more numerical results that confirm these observations. Interestingly, the constraints do not in general affect the best value of the objectives by much.

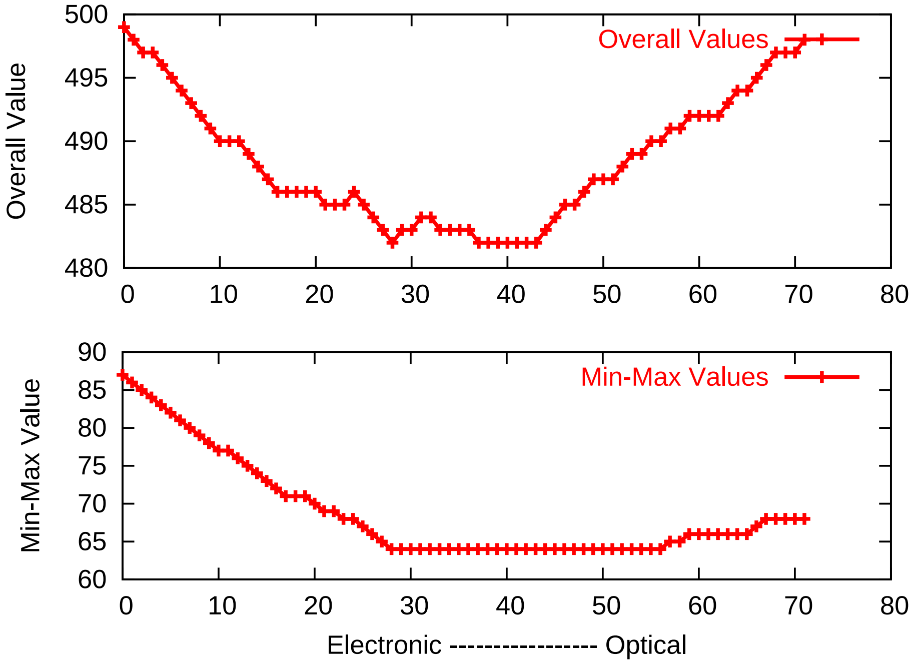


Figure 5.13: Evolution of Objectives, Random Traffic

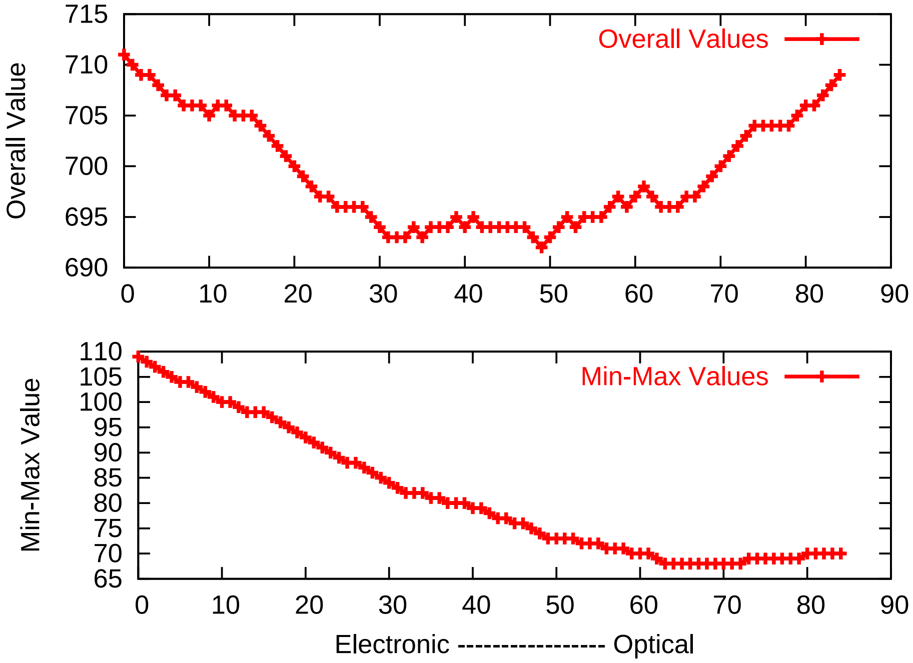


Figure 5.14: Evolution, Uniform, High Load

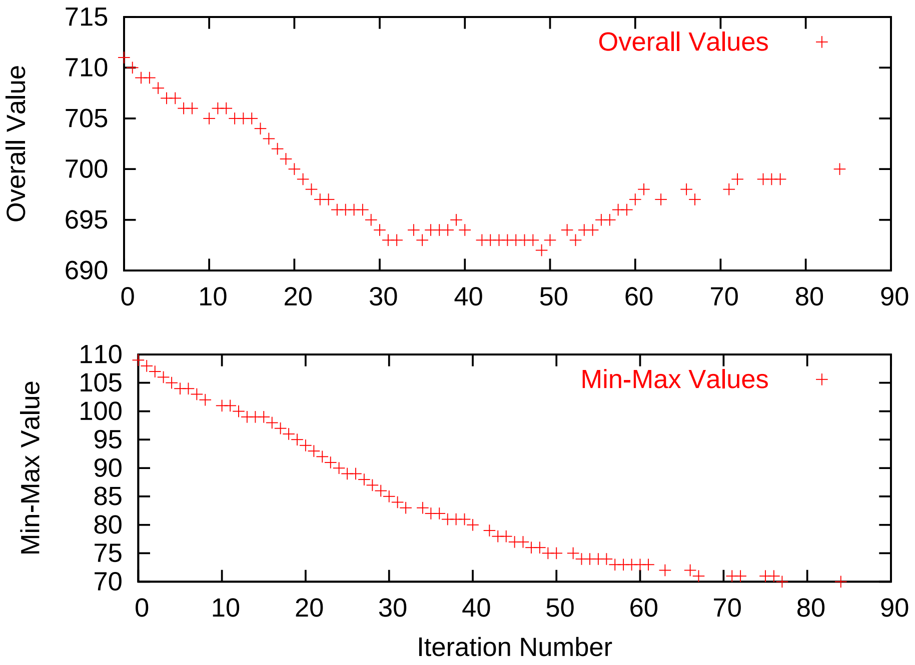


Figure 5.15: Evolution, with Constraints

Chapter 6

Hierarchical Grooming in General Topology Networks

Traffic grooming in simple, regular WDM networks has its own importance and value, as we have discussed in the previous two chapters. However, for large optical backbones with complex requirements and constraints, the limitations for the elemental topologies make them not suitable for the tasks. Therefore, research is moving in the direction of traffic grooming in general Mesh networks, which is the case for most large-scale backbone networks.

As we show in Chapter 3, there are lots of work on traffic grooming in general topology networks that take the Integer Linear Programming formulation as the starting point. In essence, solution approaches based on an ILP formulation regard the network as a flat entity for the purposes of lightpath routing, wavelength assignment, and traffic grooming. It is well known, however, that in existing networks, resources are typically managed and controlled in a hierarchical manner. The levels of the hierarchy either reflect the underlying organizational structure of the network or are designed in order to ensure scalability of the control and management functions.

Based on this observation, in this work we develop a framework for hierarchical traffic grooming in mesh networks with the objective of minimizing the total number of electronic ports in the network. Note that a lightpath requires exactly two electronic ports, one at the source and one at the destination. Hence, minimizing the number of electronic ports is equivalent to minimizing the number of lightpaths in the logical topology. To this end, we emulate the hub-and-spoke model used by the airline industry to "groom" passenger traffic onto connecting flights. At the first level of the hierarchy, the network is partitioned into clusters, and one node in each cluster (referred to as the *hub*) is responsible for grooming intra-cluster traffic as well as inter-cluster traffic originating or terminating locally. At the second level of the hierarchy, the first-level hubs form another cluster for grooming and routing inter-cluster traffic. The logical topology within a (first- or second-level) cluster is formed by viewing it as a *virtual star*, and applying a customized algorithm for stars that we develop in Chapter 5. Finally, a routing and wavelength assignment (RWA) algorithm is used on the underlying topology to route and color the lightpaths.

The hierarchical approach has the following desirable characteristics:

- it is hierarchical, facilitating control, management, and security functions;
- it decouples the grooming of traffic components into lightpaths from the routing and wavelength assignment for these lightpaths: grooming is performed on a logical hierarchy of clusters by abstracting each cluster as a virtual star, and applying efficient and near-optimal algorithms; while RWA is performed directly on the underlying physical topology, ensuring efficient use of network resources;
- it provisions only a few nodes (the hubs) for grooming traffic they do not originate or terminate;
- it handles efficiently small traffic demands: at the first level of hierarchy, nodes pack their traffic on lightpaths to the local hub; at the second level, demands among remote clusters are packed onto lightpaths between the corresponding hubs; and
- it routes large components on direct lightpaths, eliminating the cost of terminating and switching them at intermediate nodes.

6.1 A Hierarchical Approach to General Topology Grooming

The hierarchical approach we devise borrows ideas from the hub-and-spoke paradigm that is widely used within the airline industry. Specifically, we assume that the network is partitioned into clusters (or islands) of nodes, where each cluster consists of nodes in a contiguous region of the network. The clusters may correspond to independent administrative entities (e.g., autonomous systems), or may be created solely for the purpose of simplifying resource management and control functions (e.g., as in partitioning a single OSPF administrative domain into multiple areas).

For the purposes of traffic grooming, we view each cluster as a *virtual star*, and we designate one node as the *hub* of the cluster. We refer to each cluster as a *virtual star* because, even though the physical topology of the cluster may take any form (and in fact may be quite different from a *physical star* topology), the hub is the only node responsible for grooming intra- and inter-cluster traffic. Consequently, hub nodes are expected to be provisioned with more resources (e.g., larger number of electronic ports and higher switching capacity for grooming traffic) than non-hub nodes. Returning to the airline analogy, a hub node is similar in function to airports that serve as major hubs; these airports are typically larger than non-hub airports, in terms of both the number of gates ("electronic ports") and physical space (for "switching" passengers between gates).

The main idea behind our hierarchical grooming strategy is to solve the first and third sub-problems of the traffic grooming problem (i.e., construct the logical topology and determine the routing of traffic components on it) in two steps. In the first step, we apply the StarTopology algorithm introduced in Section 5.3.2 to each cluster; the result of this step is a set of lightpaths within each cluster to route local (intra-cluster) traffic, as well as inter-cluster traffic to and from the local hub. In the second step, we view all the hub nodes as forming a second-level virtual star, and we apply the StarTopology algorithm once more to determine the lightpaths and corresponding routing for inter-cluster traffic. Finally, given the above collection of inter- and intra-cluster lightpaths, we solve the RWA problem on the underlying physical topology of the network. We provide a detailed description of this hierarchical grooming algorithm in the following sections.

To illustrate our approach, let us consider the 32-node network in Figure 6.1. The bottom part of the figure shows a partition of the network into eight clusters, $B_1, \cdots, B_8$, each cluster consisting of four nodes. These clusters represent the first level of the hierarchy.

Within each cluster, one node is the hub; for instance, node 2 is the hub for cluster B_1. The top part of the figure shows the second-level cluster, consisting of the hub nodes of the eight first-level clusters; one of these nodes, say, node 13, is selected as the hub node for the second-level cluster. We emphasize that, while we view each cluster as a virtual star, the actual physical topology of the cluster is determined by the physical topology of the part of the original network where the cluster nodes lie; for example, the four nodes of cluster B_8 form a ring. Since the RWA algorithm is performed on the underlying physical topology *after* the logical topology has been determined, the lightpaths will follow the most efficient paths in the network, despite the fact that the StarTopology algorithm was developed for physical stars (see the next section). Consider, for example, cluster B_8 with node 32 as its hub. Suppose that the logical topology obtained by running the StarTopology algorithm on the corresponding *virtual star* with node 32 as the hub, includes the "one-hop" lightpath $(28, 32)$ and the "two-hop" lightpath $(31, 28)$. After running the RWA algorithm, the "one-hop" lightpath may be routed over the path $28 - 30 - 32$ (since node 28 is not directly connected to the hub node 32 of the virtual star), while the "two-hop" lightpath may in fact be routed over the direct link $31 - 28$, completely bypassing the hub node 32 (unlike a physical star where a two-hop lightpath is optically switched at the hub). Similar observations apply to all clusters at both levels of the hierarchy.

We now present the details of our hierarchical grooming approach for networks with a general topology. Our primary objective is to minimize the number of lightpaths in the logical topology; however, we are also interested in keeping the number of required wavelengths low.

The hierarchical grooming algorithm consists of three phases:

1. **Clustering and hub selection.** Partition the network into m clusters and designate one node in each cluster as the hub.

2. **Logical topology design and traffic routing.** During this phase, the first and third sub-problems of the traffic grooming problem are solved in an integrated manner. This phase is further subdivided into three parts:

   (a) setup of direct lightpaths for large traffic demands;

   (b) intra-cluster traffic grooming; and

   (c) inter-cluster traffic grooming.

The outcome of this phase is a set R of lightpaths for carrying the traffic demand matrix T, and a routing of individual traffic components $t^{(sd)}$ over these lightpaths.

3. **Routing and wavelength assignment.** Each of the lightpaths in R is assigned a wavelength and path on the underlying physical topology of the original mesh network.

The following subsections will discuss each of the three phases of the algorithm in depth. But before we proceed, a theoretical analysis will help us better understand the effect of clustering size on the grooming results.

6.2 Theoretical Analyses on Cluster Size

In this section, we present some theoretical studies on the effect of cluster size on the results of traffic grooming. Suppose the physical topology has sufficient wavelengths available, then we can ignore RWA and concentrate on the goal of minimizing total number of lightpaths in the network.

To illustrate this point, consider the tradeoffs involved in determining the number K of clusters with average m nodes per cluster. If K is small, the amount of inter-cluster traffic will likely be large. Hence, the K hubs may become bottlenecks, resulting in a large number of electronic ports at each hub and possibly a large number of wavelengths (since many lightpaths may have to be carried over the fixed number of links to/from each hub).

On the other hand, a large value for K implies a small number of nodes within each cluster. In this case, the amount of intra-cluster traffic will be small, resulting in inefficient grooming (i.e., a large number of lightpaths); similarly, at the second-level cluster, $O(K^2)$ lightpaths will have to be set up to carry small amounts of inter-cluster traffic. Therefore, the network designer must select the number and size of clusters to strike a balance between capacity utilization and number of lightpaths for both intra- and inter-cluster traffic.

We assume that the whole network consists of N nodes, and the traffic demand matrix has been reduced, so that each source/destination demand is less than C, the capacity of each wavelength. For simplicity, we assume the *uniform* traffic pattern with $t^{(sd)} = t, \forall s, d$, and assume that any virtual topology can fit into the physical topology.

Assuming there are m nodes per cluster, which gives N/m clusters overall. Two cases are studied, including very high demand ($t = C/2$) and very low demand ($t = C/N$), rounded to the nearest integers if necessary.

We study the two objectives with a variety of cluster sizes. The cluster size $m = 4, 8, 16, 32, 64, 128$.

For high loads $t = 64$, within each cluster, we use direct lightpaths to route ever traffic element. This is because each wavelength can groom only up to two traffic demands, and if we do so for non-hub demands, electronic switching will be forced at the hub nodes, breaking the lightpaths into two hops. The number of lightpaths in this solution will be roughly the same as routing the traffic non-stop, but the burden for the hub node on electronic switching, consequently the hub degrees, will be much higher. Under this assumption, we have:

$$LP = N^2 - 2N - N^2/2m + 3N/2m \tag{6.1}$$

$$HubDeg = Nm - m^2 + N/2 + m - 1 \tag{6.2}$$

Figure 6.2 shows the trend of total LPs and hub degrees in this high load case. From the results, we can see that the clustering approach favors smaller cluster size for minimizing number of lightpaths and hub degrees. The one-cluster case (last row) is special.

For very low loads $t = 1$, unlike the high-load case, if our primary goal is to use fewer number of lightpaths, it is always beneficial to do grooming. We have the following equations:

$$LP = 2N^2/m - 2N + N/m + 2m - 2 \tag{6.3}$$

$$HubDeg = N + m + N/m - 3 \tag{6.4}$$

Figure 6.3 shows the low load trends. In this case, larger cluster size gives fewer number of lightpaths, and the hub degrees don't have much difference.

From the analysis above, we find that the grooming approach behaves very differently for very low and very high traffic loads. If we assume $t = C/4$, we have to choose between decreasing the hub degree and decreasing the total number of lightpaths more carefully, which will make the analysis much more complicated. In general, if t were between C/N and $C/2$, and the traffic elements have a broad range, it will not be easy to predict the optimal cluster size, not to say that the actual physical topology will add more constraints in the problem. Therefore, experimentation with different clustering (*e.g.*, different cluster size) is required for getting a better solution.

Now, imagine a two-cluster 'dumb bell' network, with each cluster having the same size and connected with only one fiber link, it is clear that if there are limited physical links between the two clusters, clustering is helpful, because inter-cluster traffic can then be groomed together, using fewer number of lightpaths connecting the two clusters. Such design will give virtual topology more likely to succeed in RWA. Thus, physical constraints are also important in selecting the appropriate cluster size.

The theoretical analyses we make here are simplistic. In the real world, the traffic demands are generally much more complicated than uniform, and the underlying physical topology may simply prevent some cluster size to be applied in reality. We leave the task of studying the effects of clustering in more detail to Chapter 7, and just apply a manual clustering in this chapter, to illustrate how the entire hierarchical approach works.

6.3 Phase I: The Clustering and Hub Selection

The objective of this phase is twofold. First, we partition the network nodes into some number m of clusters, denoted $B_1, \cdots, B_m$. Second, we select one node in each cluster to serve as the hub where grooming of intra- and inter-cluster traffic is performed. Let n_i denote the number of nodes in cluster B_i, $n_1 + n_2 + \cdots + n_m = N$, and h_i denote the hub of cluster B_i.

The development of good clustering algorithms that lead to a logical topology with a small number of lightpaths, and which will not require a large number of wavelengths when superimposed on the underlying physical topology, is the subject of Chapter 7. In this chapter, we manually partition the network in Figure 6.1, and we experiment with clusters of various sizes in Section 6.6. We leave the study of automatic clustering methods to the next chapter.

6.4 Phase II: Logical Topology Design and Traffic Routing

Suppose we already obtain a clustering of the network from Phase I that includes m clusters and corresponding hub nodes $h_i, i = 1, \ldots, m$. We set up the logical topology of

#Clusters	Cluster composition and hub nodes
1	$\{ B_1 \cup B_2 \cup B_3 \cup B_4 \cup B_5 \cup B_6 \cup B_7 \cup B_8(\mathbf{13})\}$
2	$\{ B_1 \cup B_2 \cup B_3 \cup B_4$ $(\mathbf{13})\}$,$\{B_5 \cup B_6 \cup B_7 \cup B_8$ $(\mathbf{15})\}$
4	$\{ B_1 \cup B_2$ $(\mathbf{11})\}$, $\{ B_3 \cup B_4$ $(\mathbf{5})\}$,$\{B_5 \cup B_6$ $(\mathbf{21})\}$, $\{ B_7 \cup B_8$ $(\mathbf{32})\}$
8	$\{B_1, \mathbf{2}\}$, $\{B_2, \mathbf{11}\}$, $\{B_3, \mathbf{5}\}$, $\{B_4, \mathbf{13}\}$, $\{B_5, \mathbf{21}\}$, $\{B_6, \mathbf{25}\}$, $\{B_7, \mathbf{15}\}$, $\{B_8, \mathbf{32}\}$

Table 6.1: Cluster and hub selection for the 32-node network in Figure 6.1

lightpaths in a sequence of three steps, each addressing a category of traffic demands.

6.4.1 Setup of direct lightpaths for large traffic demands

During this step, we first reduce the traffic matrix T by assigning direct lightpaths to all traffic demands $t^{(sd)}$ that are greater than the wavelength capacity C, even if nodes s and d belong to different clusters. Since carrying C units of traffic from source s to the local hub, then to the remote hub (if different), and finally to the destination d, would require two or three lightpaths, setting up direct lightpaths for such demands is preferable given our goal of minimizing the total number of lightpaths in the logical topology.

Following the reduction step, we also apply a "direct to the destination hub" rule to set up lightpaths between some node s and a remote hub h, if the total amount of traffic from s to nodes d in h's cluster $\sum_d t^{(sd)} \geq p \times C$, where $p \in (0.5, 1)$ is a parameter determined by the network designer; in our work, we let $p = 0.8$. Setting up such lightpaths for large demands to bypass the local hub node (i.e., the hub of in the cluster of node s), has several benefits: the number of lightpaths in the logical topology is reduced, the number of electronic ports and switching capacity required at hub nodes is reduced (leading to higher scalability), and the RWA algorithm may require fewer wavelengths (since hubs will be less of a bottleneck).

Let R_{init} be the set of direct lightpaths created in this step. Let $T_r = [t_r^{(sd)}]$ denote the matrix of residual traffic demands (i.e., excluding those carried by the lightpaths in R_{init}) that need to be groomed. Obviously, $t_r^{(sd)} < C$ for all s, d. Next, we concentrate on setting up lightpaths to groom the demands $\{t_r^{(sd)}\}$.

6.4.2 Intra-cluster traffic grooming

Consider the i-th cluster B_i with n_i nodes, one of which, say, node h_i, is designated as the hub. We view cluster B_i as a *virtual star* with a $n_i \times n_i$ traffic matrix $T_i = [t_i^{(sd)}]$, defined as:

$$t_i^{(sd)} = \begin{cases} t_r^{(sd)}, & s \neq h_i, d \neq h_i \\ t_r^{(sd)} + \sum_{x \notin B_i} t_r^{(sx)}, & d = h_i \\ t_r^{(sd)} + \sum_{x \notin B_i} t_r^{(xd)}, & s = h_i \end{cases} \tag{6.5}$$

In other words, if s and d are non-hub nodes, then $t_i^{(sd)}$ represents the intra-cluster traffic from s to d. If, on the other hand, node d (respectively, node s) is the hub node, then $t_i^{(sd)}$ includes not only the intra-cluster traffic component $t_r^{(sd)}$, but also the aggregate inter-cluster traffic originating at node s (respectively, terminating at node d). This definition of $t_i^{(sd)}$ when either s or d are the hub node, implements the hierarchical grooming of traffic: all inter-cluster traffic, other than that carried by direct lightpaths set up earlier, is first carried to the local hub, groomed there with inter-cluster traffic from other local nodes, carried on lightpaths to the destination hub (as we discuss shortly), groomed there with other local and non-local traffic, and finally carried to the destination node.

Given traffic matrix $T_i = [t_i^{(sd)}]$, we view cluster B_i as a *virtual star* with hub h_i and $n_i - 1$ non-hub nodes. We apply the StarTopology algorithm in Figure 5.4 to obtain the set of lightpaths R_i for carrying the demands $\{t_i^{(sd)}\}$. Recall that the lightpaths in R_i are either "single-hop" (i.e., from a non-hub node to the hub, or vice versa), or "two-hop" (i.e., from one non-hub node to another). Hence, the routing of the traffic components $t_i^{(sd)}$ is implicit in the logical topology R_i, as we explained in Section 5.1.

We emphasize that, at this stage, we only identify the lightpaths to be created; the routing of these lightpaths over the physical topology is performed later. Depending on the actual topology of the cluster B_i, which may be quite different than that of a physical star, once routed, the lightpaths in R_i may follow paths that do not resemble at all the paths of a physical star. For instance, a "one-hop" lightpath from a non-hub node of the cluster to the hub h_i is routed on the unique link from the node to the hub in a physical star; in our case, however, the path followed by the lightpaths may consist of several links, depending on the physical topology of the network and the RWA algorithm (which we discuss in a moment). Similarly, a "two-hop" lightpath is always switched optically at the hub of a physical star; in a *virtual star* cluster, on the other hand, a "two-hop" lightpath will be routed by the

RWA algorithm on the actual underlying topology, and its path may not even pass through the hub h_i at all, if doing so is more efficient in terms of resource usage (e.g., if the two non-hub nodes are connected by a direct link).

We perform intra-cluster grooming in this manner, by applying the StarTopology algorithm to each cluster $B_i, \cdots, B_m$, in isolation. As a result, at the end of this step, we identify a set of lightpaths $R_{intra} = R_1 \cup R_2 \cup \cdots \cup R_m$ for carrying all intra-cluster traffic.

6.4.3 Inter-cluster traffic grooming

At the end of intra-cluster grooming, all traffic (other than that carried by the initial direct lightpaths) from the nodes of a cluster B_i with destination outside the cluster, is carried to the hub h_i for grooming and transport to the destination hub. In order to groom this traffic, we consider a new cluster B that forms the second-level hierarchy in our approach. Cluster B consists of the m hub nodes $h_1, \cdots, h_m$, of the first-level clusters. Let $h \in \{h_1, \cdots, h_m\}$ be the node designated as the second-level hub. We view cluster B as a *virtual star* with a $m \times m$ traffic matrix $T_{inter} = [t_{inter}^{(h_i h_j)}]$ representing the inter-cluster traffic demands. This inter-cluster matrix is defined as:

$$t_{inter}^{(h_i h_j)} \quad = \quad \sum_{s \in B_i, d \in B_j} t_r^{(s,d)}, \quad i, j = 1, \cdots, m, \ i \neq j \tag{6.6}$$

We now apply the StarTopology algorithm in Figure 5.4 to the *virtual star B* with hub h, and we obtain the set of lightpaths R_{inter} to carry the traffic demands $\{t_{inter}^{(h_i h_j)}\}$. Again, we emphasize that the routing of these lightpaths is performed on the underlying physical topology, thus, the same observations regarding the routing of the intra-cluster lightpaths above also apply to the lightpaths in R_{inter}.

Figure 6.4 provides a pseudocode description of the hierarchical logical topology algorithm. The time complexity of the algorithm is determined by the application of the StarTopology algorithm for intra- and inter-cluster grooming in Steps 5-8 and 13, respectively. The **for** loop in Steps 5-8 is executed m times, where m is the number of first-level clusters. During the i-th iteration of the loop, the StarTopology algorithm is run on a cluster of size n_i, taking time $O(n_i^2)$. Since $n_i > 1$ and $n_1 + \cdots + n_m = N$, we have that $N \leq n_1^2 + \cdots + n_m^2 \leq N^2$, hence the **for** loop takes time $O(N^2)$. Step 13 calls the StarTopology algorithm on the second-level cluster with m nodes, taking time $O(m^2)$. Since $m < N$, the overall complexity of the algorithm is $O(N^2)$.

Finally, we note that we considered only two levels of clusters in our grooming algorithm. However, for networks of very large size, our approach can be extended to three or more levels of hierarchy in a straightforward manner.

6.5 Phase III: Lightpath Routing and Wavelength Assignment

The outcome of the logical topology design phase is a set of lightpaths $R = R_{init} + R_{intra} + R_{inter}$, and an implicit routing of the original traffic components $t^{(sd)}$ over these lightpaths. Our objective is to route the lightpaths in R over the underlying physical topology, and color them using the minimum number of wavelengths.

The RWA problem on arbitrary network topologies has been studied extensively in the literature, as the review of previous work has shown in Chapter 3. In this work, we adopt the LFAP algorithm [37], which is fast, conceptually simple, and has been shown to use a number of wavelengths that is close to the lower bound. For completeness, we now describe the main steps of the LFAP algorithm.

1. Calculate a shortest path for all source-destination pairs for which a direct lightpath must be set up. List the lightpaths in R in non-increasing order of the length of their shortest path. Let the current wavelength $w \leftarrow 1$.

2. Consider each lightpath in the ordered list, and assign wavelength w and the corresponding pre-computed shortest path to as many lightpaths as possible; remove these lightpaths from the list.

3. Remove from the network topology all the links carrying lightpaths assigned wavelength w in the previous step. Consider the lightpaths remaining in the ordered list and compute a new shortest path on the new topology. Assign wavelength w and the corresponding new shortest path to as many lightpaths as possible. Remove the lightpaths that have been assigned a path and wavelength from the ordered list, and restore the original network topology.

4. If the ordered list of lightpaths is empty, stop; otherwise, set $w \leftarrow w + 1$ and repeat from Step 2.

6.6 Numerical Results for General Topology Grooming

We first obtain lower bounds on both the number of lightpaths and the number of wavelengths required to carry the traffic matrix T. These bounds are obtained *independently* of the manner (e.g., hierarchical or otherwise) in which traffic grooming is performed. Therefore, the bounds are useful in characterizing the effectiveness of our algorithm for problem instances for which it is not possible to solve the ILP directly to find an optimal solution.

6.6.1 Calculating Lower Bound for the Number of Lightpaths

A simple lower bound F_2^l on the total number of lightpaths (our main objective) has been given in Equation 4.15. This bound can be determined directly from the traffic matrix T and regardless of the underlying physical topology. We introduced it on the ring topology in Section 4.6.5, and here we simply use the same method.

6.6.2 Lower Bound for Wavelength Requirement

Consider a cut of the network, and let t be the maximum amount of traffic that needs to be carried on either direction of the links in the cut set. Let k be the number of links in the cut set, and C the capacity of each wavelength. Then, the quantity $\lceil t/kC \rceil$ is a lower bound on the number of wavelengths for carrying the given traffic matrix. This bound does not require any information regarding the logical topology or the routing and wavelength assignment of lightpaths.

We use the METIS software [42] to obtain a bisection of the 32-node network in Figure 6.1 into two groups of roughly equal size with the minimum number of links in the cut set. We use this bisection (whose cut set consists of the five links (10,18), (11,19), (15,20), (14,26), and (16,31), and partitions the network into two groups of 15 and 17 nodes each) to obtain a lower bound on the number of wavelengths for all problem instances we consider.

We also note that, once a logical topology has been determined, a lower bound on the number of wavelengths *for this particular topology* can be obtained by determining the maximum number of lightpaths that travel across the cut in either direction, and dividing it by the cut size. We expect that this bound will be higher than the one that is independent of the logical topology, and the amount of increase is an indication of the performance of the logical topology design algorithm.

With the lower bounds we obtain, we can study the performance of our hierarchical algorithms on a variety of traffic patterns for the network topology in Figure 6.1. The definitions for those traffic patterns are similar to those described in Section 4.6, but since lightpath routing is not determined for general topology networks, the definitions for the Falling and Rising traffic patterns are revised accordingly. They can be found in the corresponding subsections.

6.6.3 Results for the Random Traffic Pattern

We have found that random patterns are often challenging in the context of traffic grooming, since the matrix does not have any particular structure that can be exploited by a grooming algorithm. To generate a traffic matrix for a problem instance, we let the standard deviation of the Gaussian distribution be 150% of the mean t. Consequently, the traffic elements $t^{(sd)}$ take values in a wide range around the mean, and the loads of individual links also vary widely. If the random number generator returns a negative value for some traffic element, we set the corresponding $t^{(sd)}$ value to zero.

Figures 6.5 and 6.6 and Table 6.2 present experimental results obtained by applying the MeshTopology algorithm to each of the four clustering of the 32-node network, as shown in Table 6.1. For each clustering, we generated thirty instances of the problem, and the traffic matrix of each instance was created according to the random traffic pattern. Figure 6.5 plots, for each problem instance, the number of lightpaths created by the MeshTopology algorithm for each clustering, as well as the lower bound on the number of lightpaths from expression (4.15). Figure 6.6 plots, for each problem instance, the number of wavelengths required to establish the logical topology for each clustering, as well as two lower bounds on the number of wavelengths. The bottom curve in the figure is the lower bound based on the bisection of the 32-node network (refer to Section 6.6.2); this lower bound is independent of how the traffic grooming problem is solved. The next higher curve is the best of four the topology-specific lower bounds, each corresponding to the logical topology of a clustering, as explained in Section 6.6.2. Finally, Table 6.2 presents aggregate statistics over all thirty problem instances regarding the average lightpath length, the average maximum hub degree, and the average number of wavelengths.

#Clusters	Avg LP Length	Avg Max Hub Degree	Wavelengths
1	3.17	266	60
2	3.07	228	60
4	2.93	183	59
8	2.84	143	56

Table 6.2: Aggregate statistics for the random traffic pattern

We observe that as the number of clusters into which the network is partitioned increases, the total number of lightpaths in the resulting topology increases gradually (Figure 6.5). On the other hand, the number of required wavelengths generally decreases as the number of clusters increases (Figure 6.6), and so do the average lightpath length and the maximum hub degree. These results can be explained by noting that, as the number of clusters increases, the size of each cluster decreases. With a smaller cluster size, more lightpaths are necessary for both intra-cluster traffic (since the amount of traffic within a cluster is relatively small and lightpaths are not utilized efficiently) and inter-cluster traffic (since each hub has to establish lightpaths to a larger number of hubs in other clusters). In addition, intra-cluster lightpaths are shorter when clusters are small, and these short lightpaths are less likely to share links, resulting in fewer wavelengths. At the same time, there is relatively less traffic to be groomed at each hub, hence hub degrees (and hub cost) decrease; the fact that hubs are less of a bottleneck also reduces the wavelength requirements.

Form Figure 6.5, we note that the number of lightpaths created by our hierarchical grooming approach are only about 25-30% above the lower bound, and this behavior is consistent across all problem instances. We believe, however, that this lower bound is rather loose since expression (4.15) does not take into consideration the underlying physical topology; we are currently working on obtaining a tighter bound. From Figure 6.6, we observe that, with appropriate clustering, the wavelength requirements of our approach are close to the lower bound obtained from the bisection. We also emphasize that the topology-specific lower bound (see Section 6.6.2) lies only slightly above the overall lower bound that is independent of the specific grooming approach. In other words, our hierarchical approach does not adversely affect the wavelength requirements for a given traffic matrix.

6.6.4 Results for the Falling Traffic Pattern

This traffic pattern is designed to capture the traffic locality property that has been observed in some networks, like the locality traffic pattern we discussed in the ring networks. Specifically, if the mean of the Gaussian distribution for node pairs that have shortest distance 1 is t, then the mean for node pairs with shortest distance 2 (respectively, 3) is set to $0.8t$ (respectively, $0.6t$); for all other pairs, the mean is set to $0.2t$. This is why we call the pattern 'falling'. We also let the standard deviation of the Gaussian distribution be 10% of the mean.

Figures 6.7 and 6.8, and Table 6.3 are similar to the ones above, except that they show results for the falling pattern. As we can see, the general trends in these results are very similar to the ones we observed with the random traffic pattern. In particular, as the number of clusters increases, the total number of lightpaths also increases moderately, while the number of wavelengths, the average lightpath length and the maximum hub degree all decrease. However, comparing the absolute values to the ones obtained with the random traffic pattern reveals the effect of the traffic pattern on the overall solution. For instance, the average lightpath length is significantly smaller under the falling pattern, due to the fact that most of the traffic is destined to nodes near by, therefore, it is more likely to be confined within a cluster. There is a similar effect on the number of required wavelengths: a large cluster size is likely to force longer, indirect lightpaths that may cause wavelength collisions and require a larger number of wavelengths. Consequently, there is a significant drop in the wavelength requirements as we move from one to eight clusters (Figure 6.8), that is more pronounced than the one in Figure 6.6. Also, the clustering affects the maximum hub degrees much more dramatically than under the random pattern. In particular, when there are few clusters, even traffic destined locally is forced to travel to a relatively remote hub, increasing the degree of the hub (and the required electronic switching capacity) significantly. Increasing the number of clusters allows most of the traffic to remain within a cluster; as a result, the maximum hub degrees decrease by 66% when there are eight clusters compared to one cluster, while the corresponding decrease for the random pattern is about 46%.

#Clusters	Avg LP Length	Avg Max Hub Degree	Wavelengths
1	2.49	484	67
2	2.47	318	62
4	2.34	217	62
8	2.28	164	44

Table 6.3: Aggregate statistics for the falling traffic pattern

6.6.5 Results for the Rising Traffic Pattern

The Rising traffic pattern is designed to be the opposite of the Falling traffic pattern. Specifically, if the mean of the Gaussian distribution for node pairs that have the longest shortest distance(6 in this case) is t, then the mean for node pairs with second longest shortest distance (respectively, third) is set to $0.8t$ (respectively, $0.6t$); for all other pairs with shorter shortest distance(≤ 3 in the given physical topology), the mean is set to $0.2t$.

Figures 6.9 and 6.10 show the lightpath and wavelength results for hierarchical grooming in the rising traffic pattern. We can see that the result trends are similar to those in the falling traffic pattern, which is desirable.

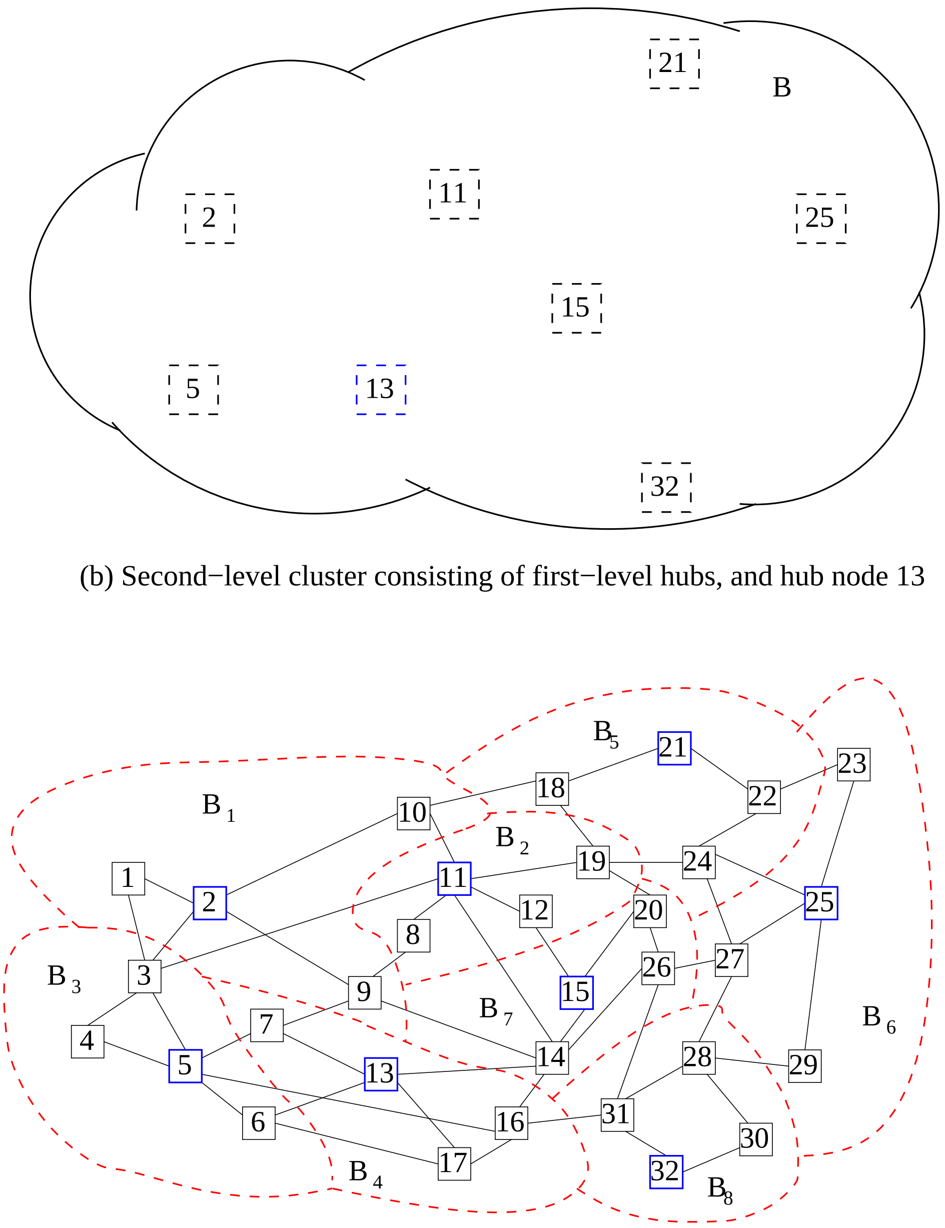
The aggregate results on hub degrees and lengths of lightpaths also show similar trends, which is not surprising given the rising traffic pattern and the balanced network topology.

6.6.6 Results for Uniform Traffic Patterns

More results from our experiments show that the performance of the hierarchical method on uniform traffic patterns is generally not desirable for instances with a large mean value. However, for instances with very small mean in the Gaussian distribution generation, the performance on the one-cluster solution is good, and is better than solutions with more clusters on both the number of lightpaths and wavelength requirements.

Several reasons account for the special characters in the Uniform traffic pattern. First, clustering is ‘grouping of similar objects’, so if every node has similar traffic demands to all other nodes, as in the Uniform pattern, clustering does not have clear advantage. Second, we calculate the lower bounds regardless of traffic patterns for the instances, thus for the Uniform traffic pattern, the bounds might be too loose to show the actual effectiveness of our algorithms. Finally, as our analysis in Section 6.2 shows, in strict uniform traffic pattern, the magnitude of each traffic demand element affects the potential solutions in different trends as the cluster size changes. In fact, our experiments do show curves similar to the theoretical results for large and very small uniform demands, respectively.

Overall, the results we have presented in the numerical results section demonstrate that our traffic grooming approach can be efficiently applied to large size networks and produce hierarchical logical topologies whose lightpaths and wavelength requirements are close to the corresponding lower bounds. We have also identified important tradeoffs between the number of clusters (or, equivalently, the cluster size) and pertinent performance metrics such as the total number of lightpaths and wavelengths, the average lightpath length, and the hub degrees. We have also shown that the tradeoffs depend on the traffic pattern, as well as on the underlying network topology. Therefore, our next goal is on developing appropriate clustering techniques for traffic grooming, which leads us into Chapter 7.



(b) Second−level cluster consisting of first−level hubs, and hub node 13

(a) First−level clusters

Figure 6.1: A 32-node WDM network, its partition into eight first-level clusters $B_1, \cdots, B_8$, and second-level cluster B consisting of the eight first-level hubs

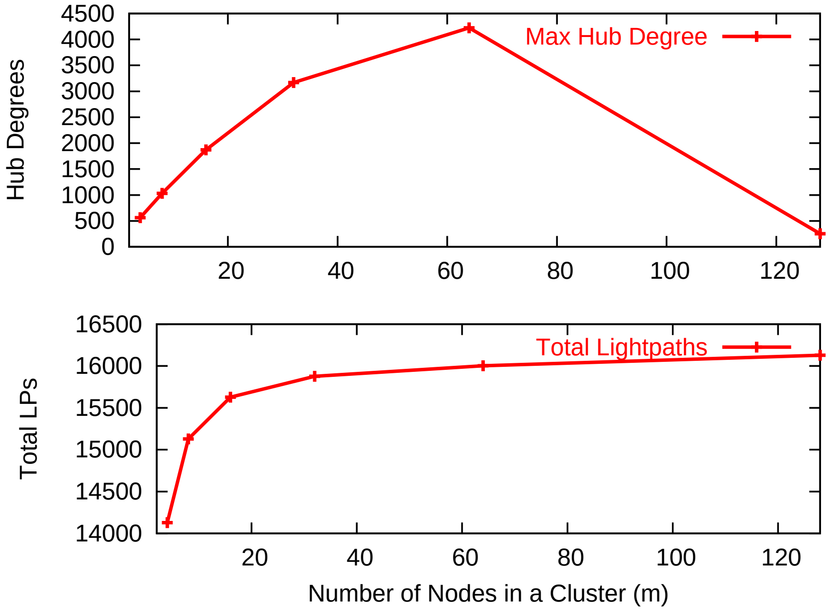


Figure 6.2: Uniform Traffic with $t = C/2 = 64, N = 128, C = 128$

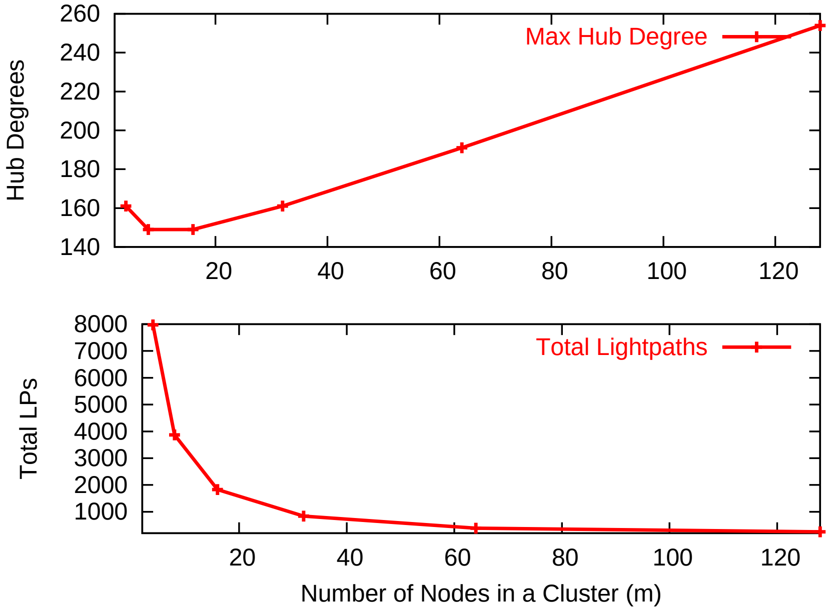


Figure 6.3: Uniform Traffic with $t = C/N = 1, N = 128, C = 128$

Logical Topology Algorithm for Mesh Networks
Input: A mesh WDM network with N nodes partitioned in m clusters $B_1, \cdots, B_m$, hub h_i of cluster B_i, W wavelengths per link, capacity C of each wavelength, and traffic matrix $T = [t^{(sd)}]$
Output: The set of lightpaths R in the logical topology, and the routing of the traffic components $t^{(sd)}$, such that $|R|$ is minimized

```
Procedure **MeshTopology**
begin // Set up direct lightpaths
  1.  Reduction: create direct lightpaths for demands $> C$
  2.  Direct to destination hub: create lightpaths to hub nodes when the aggregate traffic
      to a cluster is large ($\geq 0.8 \times C$)
  3.  $R_{init} \leftarrow$ initial set of direct lightpaths
      // Intra-cluster grooming
  4.  $T_r = [t_r^{(sd)}] \leftarrow$ residual traffic matrix
  5.  for $i = 1, \cdots, m$ do
  6.      $T_i = [t_i^{(sd)}] \leftarrow$ intra-cluster traffic matrix for cluster $B_i$, computed from expression (6.5)
  7.      $R_i \leftarrow$ set of lightpaths obtained by running the StarTopology algorithm on virtual star $B_i$
          with hub $h_i$
  8.  end for
  9.  $R_{intra} \leftarrow R_1 \cup R_2 \cup \cdots \cup R_m$
      // Inter-cluster grooming
  10. $B \leftarrow$ cluster consisting of $m$ hub nodes $h_1, \cdots, h_m$
  11. $h \leftarrow$ hub of cluster $B$
  12. $T_{inter} \leftarrow$ the $m \times m$ inter-cluster matrix from expression (6.6)
  13. $R_{inter} \leftarrow$ set of lightpaths obtained by running the StarTopology algorithm on virtual star $B$
      with hub $h$
  14. Return the set of lightpaths $R = R_{init} \cup R_{intra} \cup R_{inter}$
end
```

Figure 6.4: Logical topology algorithm for mesh networks

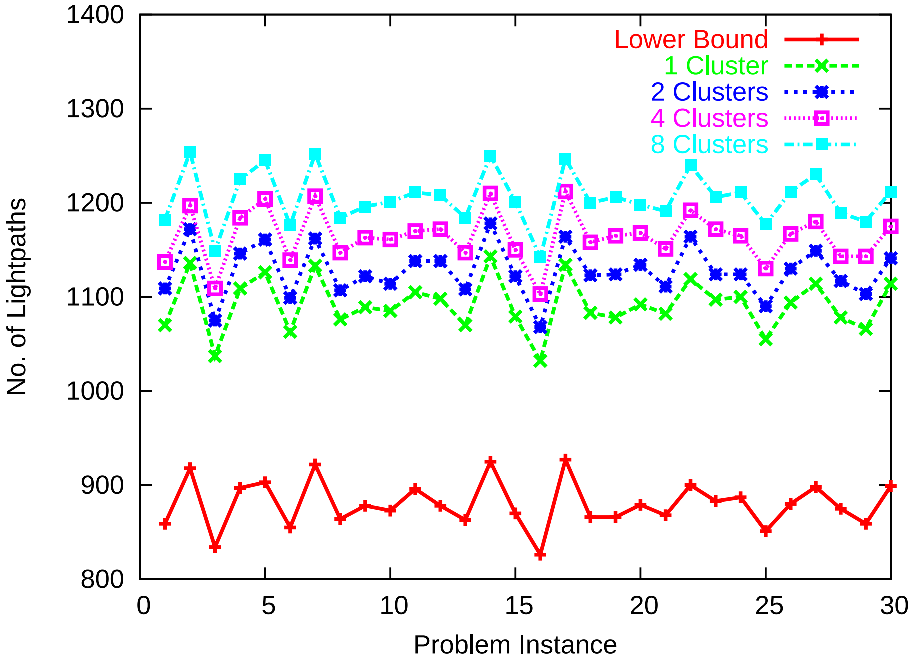


Figure 6.5: Number of lightpaths for various numbers of clusters, 32-node network of Figure 6.1 with random traffic pattern

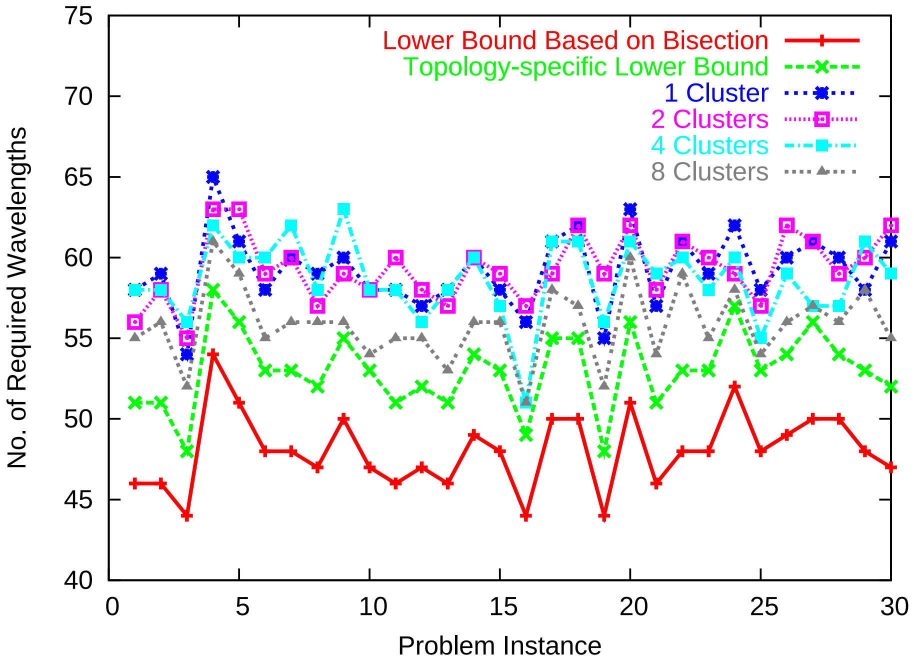


Figure 6.6: Number of wavelengths for various numbers of clusters, 32-node network of Figure 6.1 with random traffic pattern

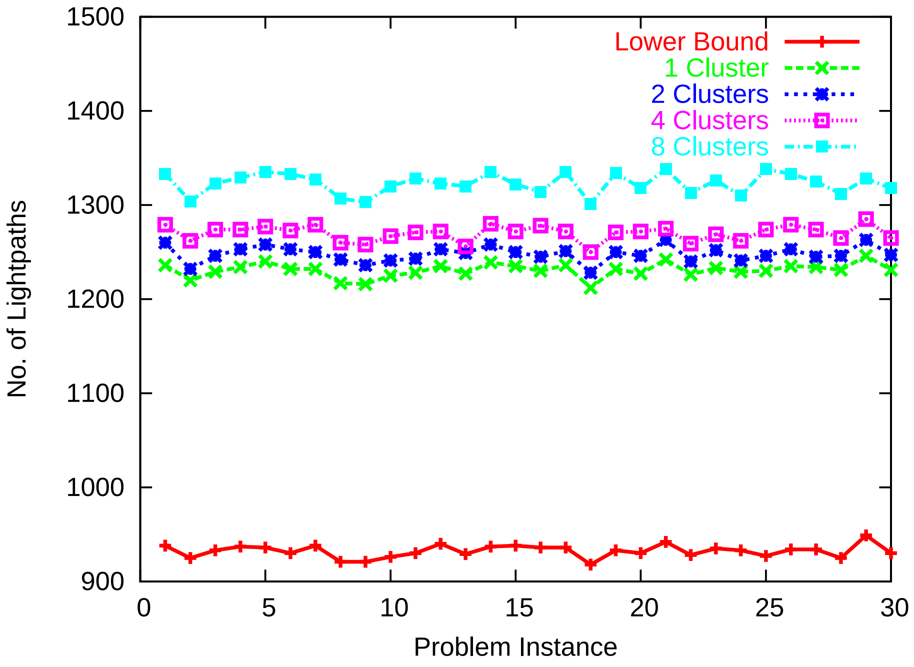


Figure 6.7: Number of lightpaths for various numbers of clusters, 32-node network of Figure 6.1 with the falling traffic pattern

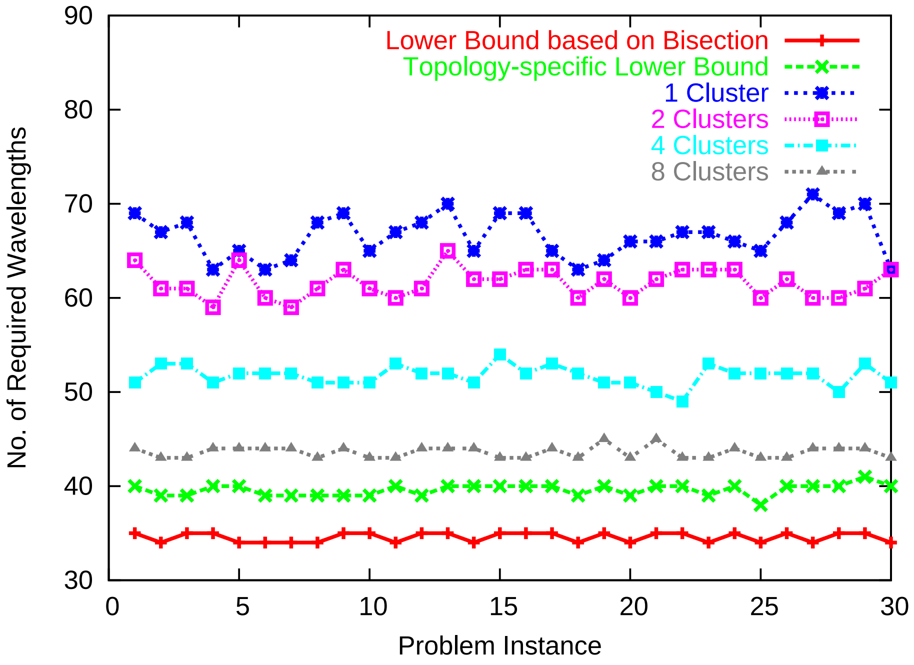


Figure 6.8: Number of wavelengths for various numbers of clusters, 32-node network of Figure 6.1 with the falling traffic pattern

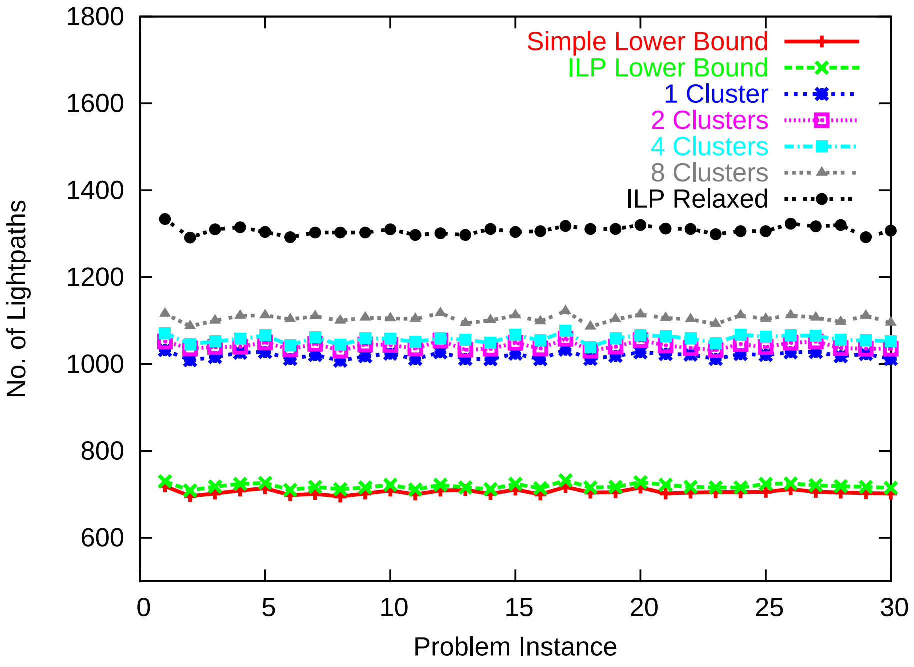


Figure 6.9: Number of lightpaths for various numbers of clusters, 32-node network of Figure 6.1 with the rising traffic pattern

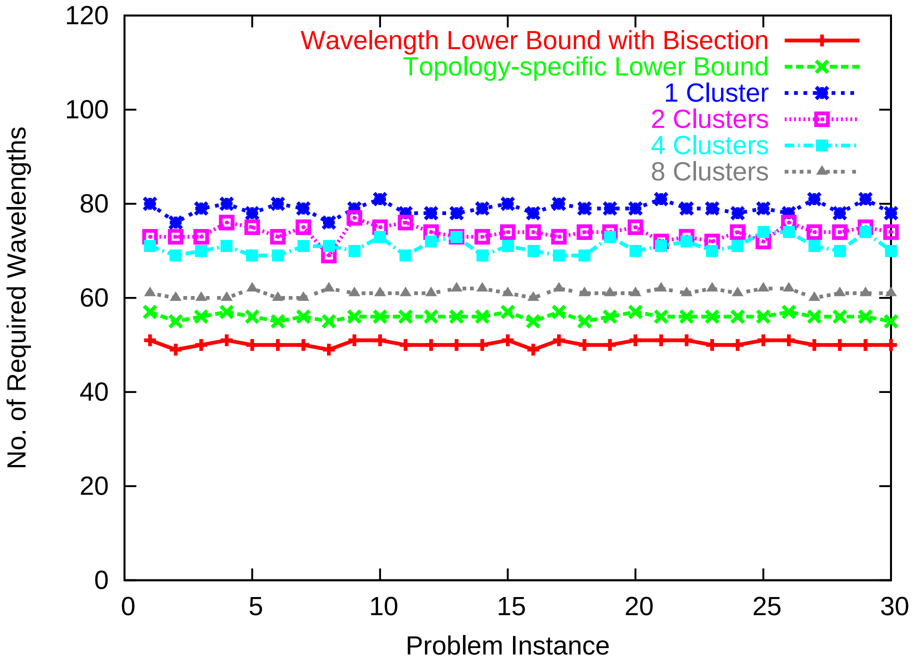


Figure 6.10: Number of wavelengths for various numbers of clusters, 32-node network of Figure 6.1 with the rising traffic pattern

Chapter 7

Clustering Algorithms for Grooming in Large General Networks

We discussed and studied the hierarchical grooming approach in the previous chapter, and showed its effectiveness in experiments on a 32-node network. One important yet challenging issue in the hierarchical grooming approach is the automatic selection of clusters and hub nodes on networks of larger size.

Although clustering techniques are used in a wide range of network design problems, there is little work related to traffic grooming. In this chapter, we first discuss network clustering methods in general, and then develop a new parameterized clustering algorithm appropriate for traffic grooming. The algorithm is flexible and allows the network designer to achieve a desired balance among a number of conflicting goals. We also improve the lower bounds we consider on the two metrics of importance, the number of lightpaths and wavelengths needed for the static grooming problem. To demonstrate the effectiveness of the clustering and hierarchical grooming algorithms, we apply them to two large networks, including a 128-node, 321-link topology corresponding to a worldwide backbone network;

the latter is approximately an order of magnitude larger than networks that have been considered in most previous grooming studies.

7.1 Clustering Methods in Network Design

Clustering is a function that arises frequently in problems related to network design and organization. A classic book [33] defines clustering as "grouping of similar objects", and discusses many mainstream clustering algorithms. The algorithms are classified as either *minimum cut* or *spanning tree*, depending on the underlying methodology. The input to the algorithms generally consist of a set of nodes (objects) and edge weights (node relationships), while the output is a partition of the nodes that optimizes a given objective function. In our case, the goal is to find a clustering that will minimize the number of lightpaths *after* applying the hierarchical grooming (logical design) approach, a fact that adds significant complexity to the problem. Specifically, the input to our problem consists of a traffic demand matrix and several constraints, in addition to the physical network topology; furthermore, unlike typical objective functions considered in the literature (e.g., the physical cut size or the amount of inter-cluster traffic), ours cannot be easily expressed as a function of the resulting clusters. Therefore, most of the existing clustering techniques are not directly applicable to the problem at hand.

Some clustering studies only consider the communication (traffic) pattern between nodes. For instance, an algorithm that can group a nearly completely decomposable (NCD) matrix into blocks, so that the weighted arcs between blocks have values not exceeding a given threshold, was introduced in [12]. The algorithm, called TPABLO, can be used to group the states of large Markov chains. A similar objective exists in the traffic grooming context, as it is desirable for traffic demands within a cluster to be "denser" than inter-cluster traffic. However, the TPABLO algorithm does not take into account the physical topology, hence it may group together nodes that are far apart. Such clusters are inappropriate for the hierarchical logical topology we consider, since the long lightpaths created for intra-cluster traffic may significantly increase the number of wavelengths required in the whole network.

Other work has focused on the physical topology only. Typically, the goal is to partition the nodes into contiguous clusters containing roughly equal numbers of nodes, and at the same time minimize the overall cut size. An example is the work in [48] on multi-objective graph partitioning, which was implemented in the METIS software package. These algorithms were designed for VLSI design, a very different problem, where equality in size and a minimum of cross-layer connections are essential for each module, and are not applicable to traffic grooming. First, there is no requirement that all clusters be equal in size; more importantly, a small physical cut size may result in bottlenecks for inter-cluster traffic, which in turn may increase the wavelength requirements during the RWA phase.

Another family of clustering problems concerned with the physical network topology includes the well-known *K-Center, K-Clustering, K-Median* and *Facility Location* problems [4, 36, 49, 51]. Unlike the applications targeted by METIS, they do not require clusters to be of equal size. Of all the variants, the *K-Center* problem is of most interest to us. The goal of the *K-Center* problem is to find a set S of K nodes (centers) in the network, so as to minimize the maximum distance from any network node to the nearest center. Thus, the set S implicitly defines K clusters with corresponding hub nodes in S. We choose to apply a *K-Center* algorithm to compare its performance on hierarchical grooming with our own clustering algorithm, so we introduce it in more detail.

7.1.1 The *K-Center* Problem and Algorithms

A solution to the *K-Center* problem may be useful for hierarchical traffic grooming since it is likely to lead to short lightpaths within a cluster, thus lowering the wavelength requirements. In addition, this type of clustering tends to avoid creating path-like physical topologies for each cluster; path-like topologies are not a good match for the StarTopology algorithm we use for grooming local traffic, which treats each cluster as a *virtual star*.

The *K-Center* problem is NP-Complete, and the best approximation ratio that can be obtained in polynomial time is 2 [30, 35]. We implemented the 2-approximation algorithm in [30] for *K-Center*, and we compare it to our own clustering method in Section 7.5. For completeness, the steps of the algorithm are listed below.

1. Create a single cluster, $B_1 = \{v_1, \dots, v_n\}$, with hub node $h_1 = v_1$. Calculate the all-pair shortest paths, record the distances in matrix $dist$, and let $x \leftarrow 1$.

2. Let x be the number of clusters, and d be the maximum distance between any node and its hub, i.e., $d = \max\{dist(v_i, h_j)\}, v_i \in B_j$. Let v be a node such that the

distance between v and its hub is d.

3. Create a new cluster B_{x+1} with $h_{x+1} = v$ as the only node. Then for each node v', if v' is closer to v than to its current hub, move v' from its current cluster to the new cluster B_{x+1}. Let $x \leftarrow x + 1$.

4. Repeat Steps 2 and 3 $K - 1$ times, adding one cluster at each iteration, for a total of K clusters.

We implement the above *K-Center* algorithm directly, to compare it with our own clustering algorithm later in our experiments.

More recently, some studies have explored clustering techniques in the context of traffic grooming: a hierarchical design for interconnecting SONET rings with multi-rate wavelength channels was proposed in [25], and in [16], the "blocking island" paradigm is used to abstract network resources and find groups of bandwidth hierarchies for a restricted version of the traffic grooming problem. Our work presented in this chapter is more comprehensive and it is applicable to many variants of the grooming problem.

7.2 Analysis of Factors in Clustering for Traffic Grooming

As we discussed before, the clusters and hubs are the input to the subsequent logical topology design and RWA phases of the framework, and is important for obtaining good solutions in the hierarchical grooming approach.

In Section 6.2, we have already analyzed the effect of cluster sizes on the resulting lightpath count assuming there is no physical constraint, and the traffic demands are uniform. Now we further consider the composition of each cluster with arbitrary traffic pattern. If the average traffic demand between nodes within a cluster is higher than the average inter-cluster demand, there will tend to be fewer inter-cluster lightpaths, which are typically longer than local lightpaths. Therefore, it is desirable to cluster together nodes with "denser' traffic between each other: doing so reduces the number of longer lightpaths, alleviates hub congestion, and provides more flexibility to the RWA algorithm (since long lightpaths are more likely to collide during the routing and wavelength assignment phase).

On the physical topology side, we also need to consider the cut links that connect different clusters. Each cluster has a number of fibers that link to nodes outside the cluster, and all traffic between a node outside the cluster and one within must traverse these cut links. Since the cut links must have sufficient capacity to carry the inter-cluster traffic, it is important to select clusters so that their cut size is not too small, in order to keep the wavelength requirements low.

Another important consideration arises in physical topologies for which there exists a critical small cut set that partitions the network into two parts. In such a topology, all traffic between the two sides of the bisection will have to go through the cut. In this case, creating clusters that consist of nodes on different sides of the cut may be undesirable, because it may generate unnecessary traffic that goes back and forth through the cut. Consider a cluster with nodes i, j, on one side of the bisection, and the hub h on the other. Due to the nature of the hierarchical grooming approach, traffic between i and j may need to be sent to the hub first, creating additional traffic across the cut links, with a corresponding increase in the number of required wavelengths. This additional traffic can be eliminated by forcing nodes on different sides of the bisection to be in different clusters. We describe shortly a pre-cutting technique that can be useful in such situations.

The physical shape of each cluster may also affect the wavelength requirements. In particular, it is important to avoid the creation of clusters whose topology resembles that of a path, since in such topologies the links near the hub can become congested. Since we use a *virtual star* approach for logical topology design within each cluster, topologies with relatively short diameter are more attractive in terms of RWA.

In the next section, we describe an algorithm that takes into account all the above factors in partitioning the network into clusters so as to yield good grooming solutions.

7.3 A Clustering Algorithm for Hierarchical Grooming

Now we introduce a clustering algorithm we design especially for the hierarchical grooming approach. The objective of the algorithm is twofold: to partition the network into some number m of clusters, denoted $B_1, \cdots, B_m$, and to select one node in each cluster to serve as the hub where grooming of intra- and inter-cluster traffic is performed.

Figure 7.1 provides a pseudocode description of our MeshClustering algorithm that we use to partition a network of general topology in order to apply our hierarchical traffic grooming framework. The algorithm includes several user-defined parameters that can be used to control the size and composition of clusters, either directly or indirectly. Parameters *MinCS* and *MaxCS* represent the minimum and maximum cluster size, respectively. Our algorithm treats these parameters as an *indication* of the desirable range of cluster sizes, rather than as hard thresholds that cannot be violated. Although the algorithm attempts to keep the size of each cluster between the values of these two parameters (inclusive), it has the freedom, based on the values of the other parameters, to determine what it thinks may be the best clustering. Consequently, the final result may contain clusters larger than *MaxCS* (see also the discussion below regarding Step 26 of the algorithm).

The parameter Δ ($0.5 \leq \Delta \leq 0.8$, default value $\Delta = 0.8$) is used to test whether there is sufficient capacity at the hub node, as well as the edges connecting the cluster to the rest of the network, to groom/carry the traffic demands. Specifically, we require that the inter-cluster traffic originating from or terminating at a given cluster do not exceed a fraction Δ of the hub capacity (this is the HUBTEST in Step 9 of the algorithm); similarly, this intra-cluster traffic must not exceed a fraction Δ of the capacity of the links connecting the cluster to the rest of the network (the CUTTEST in Step 10 of the algorithm). The algorithm will consider a node to add to a cluster only if doing so will not violate these two constraints.

The parameter δ is meant to control the ratio of the diameter of a cluster to the number of nodes it contains. In order to avoid cluster topologies that resemble long paths, we require that $0 < \delta \leq 0.75$. We used the value $\delta = 0.75$ in our experiments; this value corresponds to a 4-node path, hence restricting the longest path within a cluster to no more than three links. Finally, the parameter $\rho, 0.8 \leq \rho \leq 1.25$, specifies the acceptable range for the ratio of intra- to inter-cluster traffic for a given cluster. As we discussed earlier, it is desirable to cluster together nodes that exchange a substantial amount of traffic among themselves relative to traffic they exchange with the rest of the network; therefore, we let $\rho = 1.25$ in our implementation.

The MeshClustering algorithm in Figure 7.1 generates one cluster during each iteration of the main **while** loop between Steps 1 and 25. Initially, in Steps 2-4, the hub of a new cluster B is selected as the node with the maximum remaining capacity among those not yet assigned to a cluster; by "remaining capacity" we mean the capacity remaining on its incident links after subtracting the bandwidth taken up by any direct lightpaths set up as discussed in Section 6.4.1. Following the hub selection, we grow the cluster by adding one node during each iteration of the innermost **while** loop between Steps 5 and 24. At each iteration, the set Q of candidate nodes for inclusion in cluster B consists of all nodes, not yet assigned to another cluster, which are adjacent to nodes in B. For each node $q \in Q$, we first check whether including q in B would result in a cluster that passes both the HUBTEST (Step 9) and CUTTEST (Step 10); if not, node q is removed for consideration for inclusion into cluster B (Step 17). For all nodes q that pass both tests, we compute the diameter-to-nodes ratio δ_q and intra-to-inter-cluster traffic ratio ρ_q, assuming that q is added to cluster B (Steps 11-16). Let q_0 be a node that passes both tests and has the largest ρ_q value among the candidates; if there are multiple such nodes, we select the one with the smallest δ_q value. We include q_0 to cluster B (Steps 21-23), and the process is repeated as long as the size of B is less than *MaxCS*.

Once all nodes have been assigned to clusters, it is possible for one or more of the clusters to have fewer than *MinCS* nodes. In this case, at Step 26, the algorithm removes these clusters and includes their nodes into adjacent clusters. We just simply combine them with the first adjacent cluster found in the detection process, because normally with carefully chosen parameters, this case rarely happens, and even if it does, how to combine the smaller clusters with existing large ones will affect the final results very much. As a result, at the end of the algorithm some clusters may contain more than *MaxCS* nodes.

The running time complexity of the algorithm is determined by Steps 1-25. The main **while** loop executes a number of times equal to the number m of clusters; for each cluster, the innermost **while** loop executes a number of times equal to the size of the cluster. Therefore, Steps 5-24 of the algorithm are repeated N times, where N is the number of nodes in the network. Each time, the **foreach** loop executes at most N times. Steps 9, 10, 12, and 13 each take no more than $O(N^2)$ time in the worst case, while all other steps take constant time. Therefore, the asymptotic complexity of the algorithm is $O(N^4)$. However, this bound is quite loose; in practice, we have found that the algorithm takes only a few seconds for the 128-node, 321-link network we consider in Section 7.5.

7.3.1 Pre-Cutting for Imbalanced Topologies

As we mentioned in Section 7.2, when the topology has a bisection of small cut size, the cut links are likely to become congested, as they have to carry all traffic between the two parts of the network on either side of the bisection. In order to reduce congestion, and hence the number of required wavelengths, in such topologies, it may be necessary to disallow nodes on different sides of the bisection from being in the same cluster. However, identifying such a critical bisection in a large, imbalanced topology, is a difficult task. Later in Section 7.4.2, we describe a method we developed for this purpose, and which we also use to obtain better lower bounds on the number of wavelengths.

Once we identify a critical bisection, we apply the following approach. First, we use the MeshClustering algorithm to determine a clustering that does not take the bisection into consideration. Then, we partition the network into two parts along the bisection, and we apply the MeshClustering algorithm on each part separately; this ensures that no cluster contains nodes from both sides of the bisection. We then select the clustering that requires the fewest lightpaths after the logical topology and RWA phases, unless it requires a significantly larger number (e.g., 10% or more) of wavelengths; in this way, we achieve balance between the lightpath objective and the wavelength requirements.

7.4 Improving Lower Bounds

In Section 6.6.1 and 6.6.2, we have introduced methods for obtaining lightpath and wavelength lower bounds for traffic grooming in general topology networks. As the network size grows larger, we want to improve the methods to calculate tighter lower bounds for evaluating our algorithms. The following subsections discuss the improvements we make on the two values we consider.

7.4.1 Improving Lower Bound for the Number of Lightpaths

The lightpath lower bound introduced in 4.6.5 is based on the observation that each node must source and terminate a sufficient number of lightpaths to carry the traffic demands from and to this node, respectively. This bound can be determined directly from the traffic matrix T.

It is possible to obtain a better lower bound using a standard ILP relaxation technique. Starting from the ILP formulation (e.g., the one we have given in Section 2.3) of the traffic grooming problem, we remove some variables and constraints, so that we can obtain a relaxed solution that serves as a lower bound for the original objective. In the relaxed ILP formulation, let b_{sd} denote the number of direct lightpaths set up from s to

d. Since all traffic originating at source node s must be carried on some lightpath also originating at s, the following constraints must be observed:

$$\sum_d b_{sd} C \geq \sum_d t^{(sd)} \qquad \forall s \tag{7.1}$$

Similarly, for each destination d we have that:

$$\sum_s b_{sd} C \geq \sum_s t^{(sd)} \qquad \forall d \tag{7.2}$$

Since our goal is to minimize the overall number of lightpaths, the resulting relaxed ILP formulation is:

Minimize: $\sum_{s,d} b_{sd}$

Subject to: Constraints (7.1) and (7.2).

We emphasize that this ILP will not necessarily yield a meaningful solution to the original grooming problem, only a lower bound. By configuring CPLEX to use dual steepest-edge pricing, we are able to compute this bound within a few seconds even for the 128-node topology we consider in the next section. Although this bound performs better than the simple bound F_2^l expressed in Equation 4.15, we believe that it is still somewhat loose. However, we have found that introducing additional constraints on traffic flow and/or routing into the relaxed ILP in order to improve the lower bound tends to increase substantially the running time of CPLEX, to the point that it becomes impractical for the large networks we consider in this work. Therefore, we use the lower bound obtained from the above simple ILP in our experimental study.

7.4.2 Lower Bound on Wavelength Requirements for Imbalanced Topologies

For the calculation of wavelength requirement bounds, we still use the bisection method discussed in Section 6.6.2. The main challenge for the bisection method is to find a bisection of the network that yields a good lower bound. To this end, we observe that there is a tradeoff between the cut size and the relative sizes of the node sets at each side of the bisection. On the one hand, using software such as METIS [48] to divide the network in two equal parts may not yield a good bound if, due to the irregular nature of the topology, the cut size turns out to be large. On the other hand, a small cut size may not be effective either if one node set is significantly larger than the other, resulting in little traffic across the cut links.

In order to reconcile these conflicting objectives, we used the CHACO software [34] that implements the partitioning algorithm in [43]. The software uses the parameter KL-IMBALANCE to control the relative sizes of the node sets on either size of the bisection. To determine a cut that achieves a good balance between the two objectives, we apply CHACO several times, varying the KL-IMBALANCE parameter, and obtain several different bisections of the physical topology. We then select the bisection that corresponds to the best (highest) lower bound, under the assumption of uniform traffic. We use this bisection to calculate wavelength bounds for all problem instances on this topology, even though the actual traffic matrix is generated based on a different pattern.

7.5 Numerical Results for Clustering Algorithms

In this section, we present experimental results to demonstrate the performance of our clustering and hierarchical grooming algorithms. The traffic matrix $T = [t^{(sd)}]$ is generated in the same way as before, with Random, Falling and Rising traffic patterns. To see detailed definitions, please refer to corresponding parts of Section 6.6.

For a given network topology and traffic pattern, we generate thirty problem instances and we compare our MeshClustering algorithm (shown in Figure 7.1) to the *K-Center* algorithm [30] we described in Section 7.1. We consider two performance metrics in our study: the *normalized lightpath count* and the *normalized wavelength count.* The former is the ratio of the number of lightpaths required for hierarchical traffic grooming, when clustering is performed by one of the two algorithms above, to the lightpath lower bound obtained using the relaxed ILP in Section 7.4.1; the latter is the ratio of the number of wavelengths required to the wavelength lower bound computed as explained in Section 7.4.2. The normalized metrics filter out the effect of the traffic matrix, allowing us to compare results among problem instances created by very different traffic patterns; obviously, a smaller value of the two metrics implies a better solution. We also consider two network topologies, one in each of the following two subsections. We emphasize that the size of the second topology is about an order of magnitude larger than the typical topology considered in previous grooming studies, a fact that demonstrates the scalability of our hierarchical grooming approach.

7.5.1 Results on a 47-node Network

We first consider a 47-node, 96-link network adapted from a topology which appeared in a historical paper on network design [5]. The node degree of the network is relatively high and the physical topology is balanced, in the sense that there is no bisection with a small cut size that can be bottleneck in traffic grooming.

Figures 7.2 and 7.3 plot the normalized lightpath and wavelength count, respectively, for each of thirty problem instances whose traffic matrix was generated according to the falling pattern. For each problem instance, four values are shown, corresponding to four different clusterings. The first two are from the *K-Center* algorithm, with the number of clusters K equal to 4 and 6, respectively. The other two are from our MeshClustering algorithm. Recall that our algorithm does not take the number of clusters as input; rather, it tries to optimize it. Consequently, the algorithm may produce different clusters for two different problem instances, even if they are defined on the same topology and their matrices are drawn from the same distribution. To make the comparison against *K-Center* as fair as possible, we selected two sets of values for the user-defined parameters of MeshClustering (refer to Figure 7.1) so that the average number of clusters over all thirty instances is 3.52 and 5.45, respectively.

Let us first consider the normalized lightpath count. From Figure 7.2, we observe that the number of lightpaths required for hierarchical grooming is about 40% higher than the lower bound, regardless of the clustering algorithm used. Recall that the lower bounds were obtained by relaxing most of the constraints in the ILP formulation, hence we believe that they are rather loose. Therefore, these results (as well as the ones to be discussed shortly) serve as a validation of our approach, as they demonstrate that the lightpath requirements for hierarchical grooming are close to optimal. We also observe that except for a couple of instances, the curves corresponding to the MeshClustering algorithm lie below those corresponding to the *K-Center* algorithm. Although the difference is not high, we would like to point out that a 1% reduction in the number of lightpaths in this network with relatively dense demands would result in about 40 fewer electronic ports, a substantial savings in cost.

Let us now turn our attention to the normalized wavelength count. As Figure 7.3

demonstrates, the MeshClustering algorithm requires significantly fewer wavelengths than the *K-Center* algorithm. This result is due to the fact that unlike the *K-Center* algorithm, ours is designed to take the wavelength requirements into account. In absolute terms, the difference in the number of wavelengths for these problem instances is around 30. Put another way, given a constraint on the number of available wavelengths, our algorithm is more likely to generate clusters that admit a feasible grooming solution. We also note that the large values of the normalized wavelength count are because the lower bound is loose in this case. Recall that a good bound depends on finding a network cut of small size and large cross-cut traffic. However, this particular 47-node network was designed to avoid such a bottleneck cut. Furthermore, the falling traffic pattern makes it unlikely that a large amount of traffic will cross any network cut; as we shall see in a moment, the rising pattern yields better bounds.

Figures 7.4 and 7.5 are similar to Figures 7.2 and 7.3, respectively, but show results for the rising traffic pattern. Note that due to the nature of this pattern, relatively large amounts of traffic will cross any network cut, resulting in the much tighter wavelength bounds in Figure 7.5. Again, except for a few instances, our clustering algorithm outperforms the *K-Center* algorithm. Regardless of the clustering algorithm, we observe that hierarchical grooming is also close to optimal, confirming our earlier observations.

7.5.2 Results on a 128-node Network

We now consider a 128-node, 321-link network, which corresponds to the worldwide backbone operated by a large service provider; we obtained the topology information from data documented on CAIDA's web site [2]. This topology is imbalanced, in the sense that there exists a bisection with a small cut size of 5 links that divides it into two parts of 114 and 14 nodes, respectively. We identify this critical cut with the method discussed in Section 7.4.2, and use it to calculate the lower bound on the number of wavelengths. We also steer our algorithm towards creating clusters that contain nodes on one side of the cut only, as we explained in Section 7.3.1.

Figures 7.6 and 7.7 plot the normalized lightpath and wavelength count, respectively, for thirty instances generated according to the random traffic pattern; Figures 7.8 and 7.9 show similar plots but for instances generated according to the rising pattern. For the *K-Center* algorithm, we let the number K of clusters be either 9 or 10, and we selected the parameters of the MeshClustering algorithm so that it also produces either 9 or 10 clusters (the average over all instances is 9.33 and 9.03 for the random and rising patterns, respectively). As we can see, our clustering algorithm slightly outperforms *K-Center* in terms of the number of lightpaths, and both algorithms are relatively close to the (loose) lower bound. However, in terms of the number of wavelengths, our algorithm produces results that are within 5% of the lower bound, whereas *K-Center* requires more than twice the number of wavelengths of our algorithm.

From the numerical results, we can conclude that our clustering algorithm is effective especially on lowering the wavelength requirements for general topology traffic grooming, and for imbalanced network topologies, this is very important for getting feasible solutions. As to the number of lightpaths, our clustering algorithm also provides some improvement. Since thr traffic patterns we consider use the same mean value with statistics randomness, we do not consider the more extreme patterns in which traffic density varies apparently from region to region. Even so, the clustering algorithm still tries to give observable improvement within the given inputs.

A Clustering Algorithm for Mesh Networks
Input: A mesh network with a set V of $|V| = N$ nodes, capacity C for each wavelength, and reduced traffic matrix $T_r = [t_r^{(sd)}]$.
User-defined parameters: $MinCS, MaxCS$ for the desired minimum and maximum cluster size, respectively, a threshold $0.5 \leq \Delta \leq 0.8$, a cluster diameter-to-nodes ratio $0 < \delta \leq 0.75$, and an intra-to-inter-cluster traffic ratio $0.8 \leq \rho \leq 1.25$.
Output: A partitioning of the node set V into some number m of clusters, $B_1, \ldots, B_m$, and the selection of node h_i as the hub of cluster B_i, such that the size of each cluster is roughly between $MinCS$ and $MaxCS$ and the clustering will lower the lightpath and wavelength requirements of the subsequent hierarchical logical topology design and RWA algorithms.

```
Procedure MeshClustering
begin
while $V \neq \phi$ do
    $v \leftarrow$ node in $V$ with maximum remaining capacity
    $B \leftarrow \{v\}$    // new cluster $B$ with hub $v$
    $V \leftarrow V - \{v\}$
    while $V \neq \phi$ and $|B| < MaxCS$ do    // grow cluster $B$
        $Q \leftarrow$ set of nodes $\in V$ adjacent to nodes in $B$
        foreach node $q \in Q$ do
            $B' \leftarrow B \cup \{q\}$    // assume $q$ is included in $B$
            HUBTEST: does traffic between $B'$ and $\overline{B'}$ occupy more than $\Delta$ of the
                remaining hub capacity?
            CUTTEST: does traffic between $B'$ and $\overline{B'}$ occupy more than $\Delta$ of the
                remaining cut link capacity?
            if $q$ passes both tests then
                $x \leftarrow$ total traffic between $q$ and nodes in $B$
                $y \leftarrow$ total traffic between $q$ and nodes in $\overline{B'}$
                $\rho_q \leftarrow x/y$    //intra- to inter-cluster traffic ratio
                $d \leftarrow$ diameter of induced subgraph $B'$
                $\delta_q \leftarrow d/|B'|$    // diameter-to-nodes ratio
            else $Q \leftarrow Q - \{q\}$
        end for
        if $Q = \phi$ then break    // cannot grow cluster $B$
        else
            $q_0 \leftarrow$ node $\in Q$ with largest $\rho_q$ and smallest $\delta_q$
            $B \leftarrow B \cup \{q_0\}$    // grow cluster $B$ to include $q_0$
            $V \leftarrow V - \{q_0\}$
    end while    // continue until cluster $B$ cannot grow further
end while
Combine clusters of size $< MinCS$ with adjacent clusters
end
```

Figure 7.1: Clustering algorithm for mesh networks

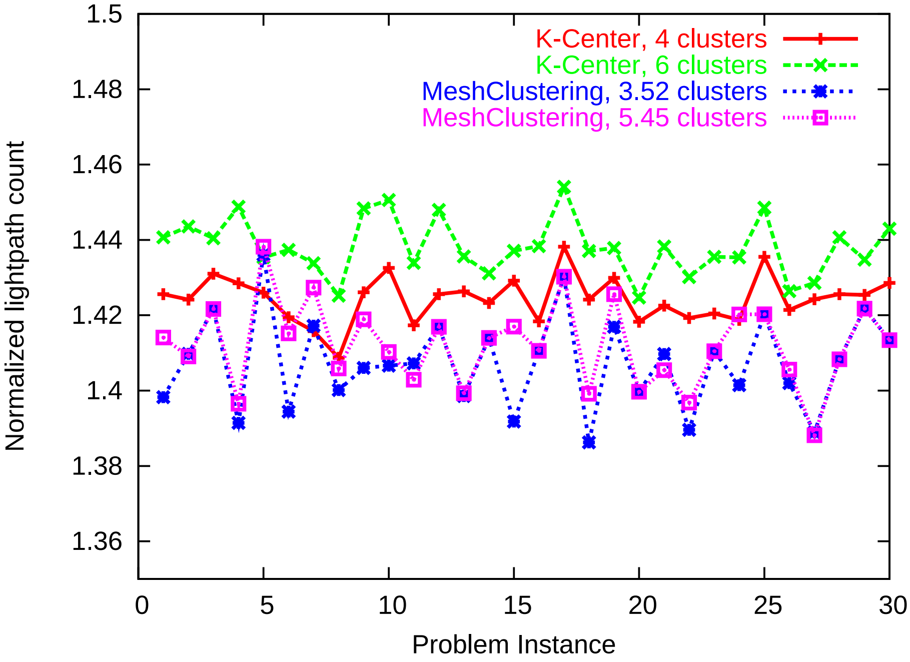


Figure 7.2: Lightpath comparison, falling pattern, 47-node network

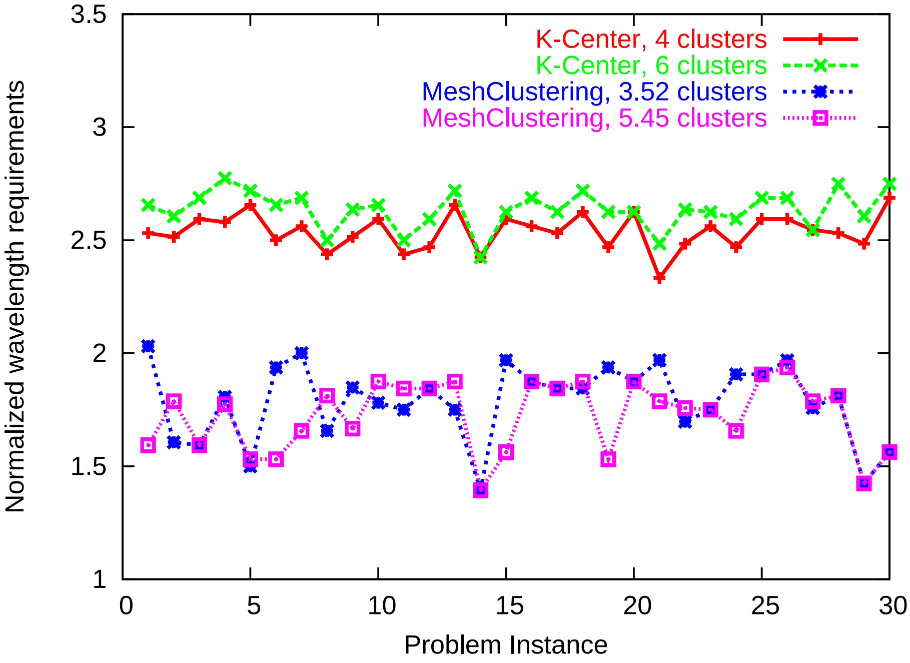


Figure 7.3: Wavelength comparison, falling pattern, 47-node network

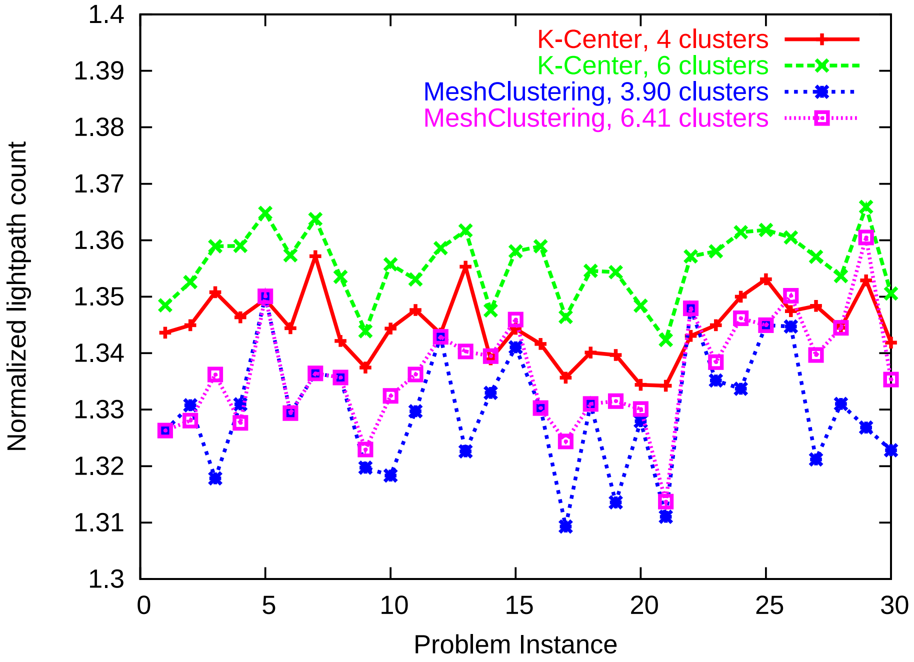


Figure 7.4: Lightpath comparison, rising pattern, 47-node network

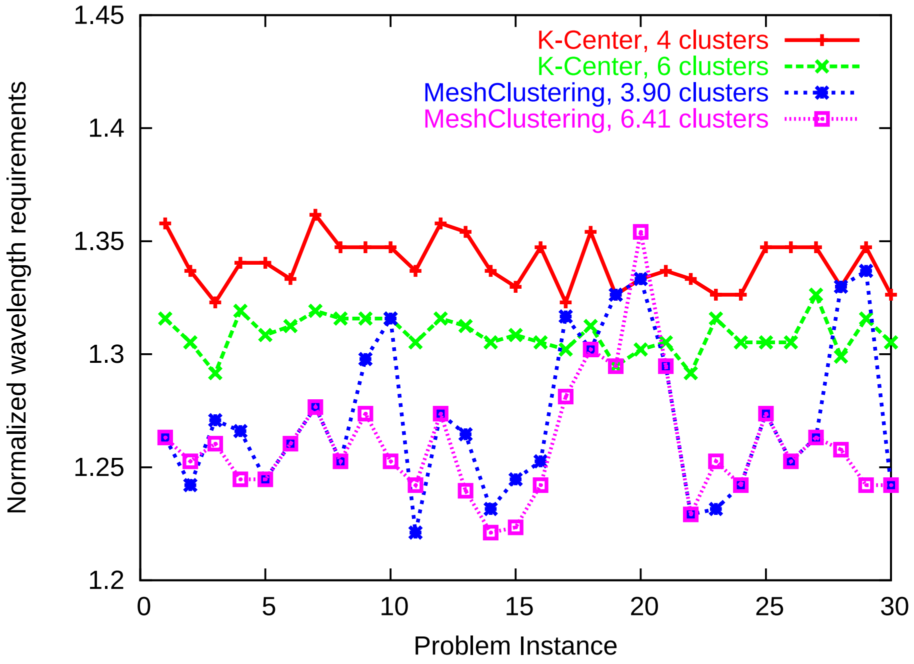


Figure 7.5: Wavelength comparison, rising pattern, 47-node network

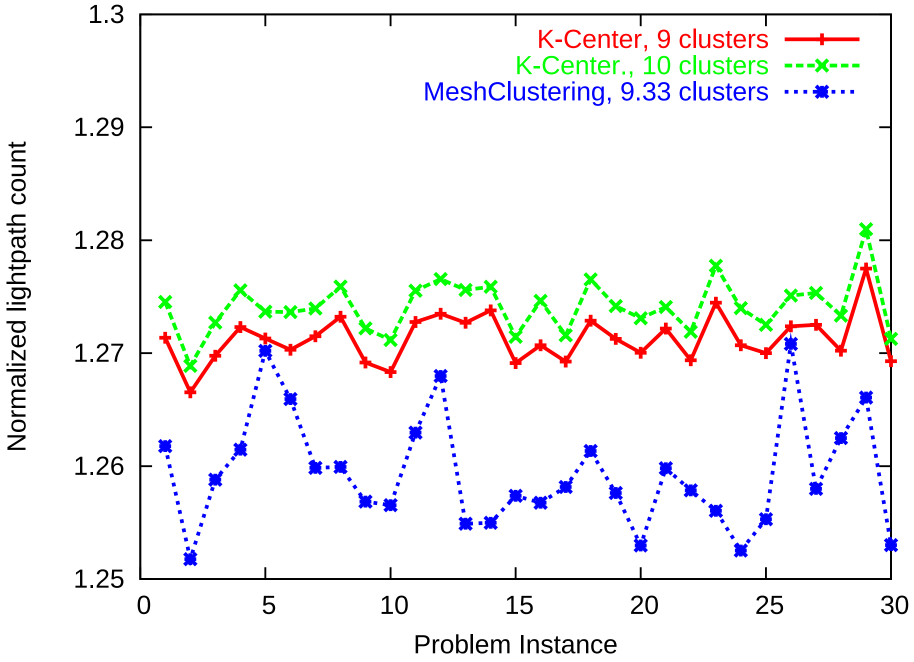


Figure 7.6: Lightpath comparison, random pattern, 128-node network

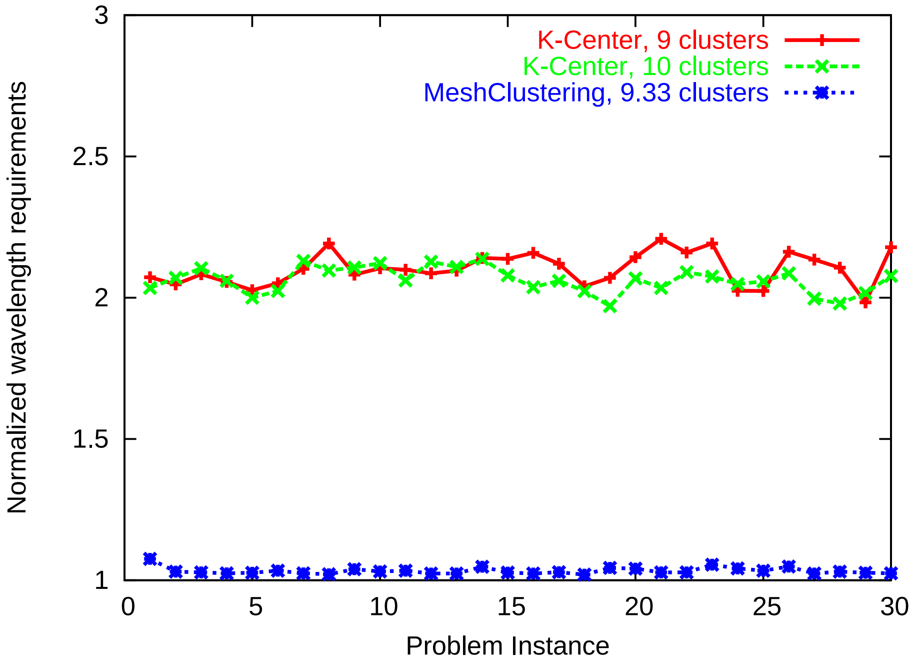


Figure 7.7: Wavelength comparison, random pattern, 128-node network

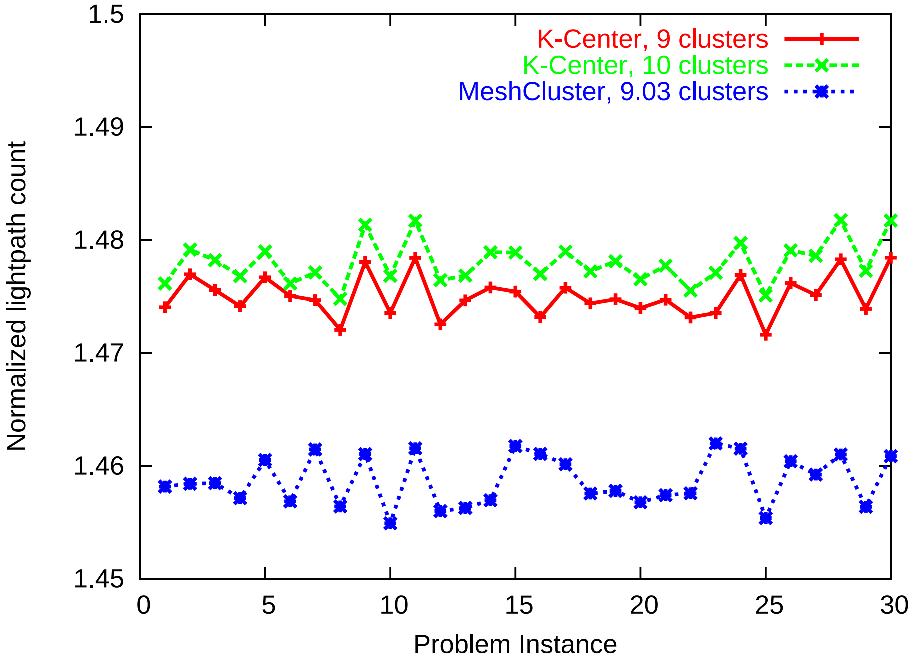


Figure 7.8: Lightpath comparison, rising pattern, 128-node network

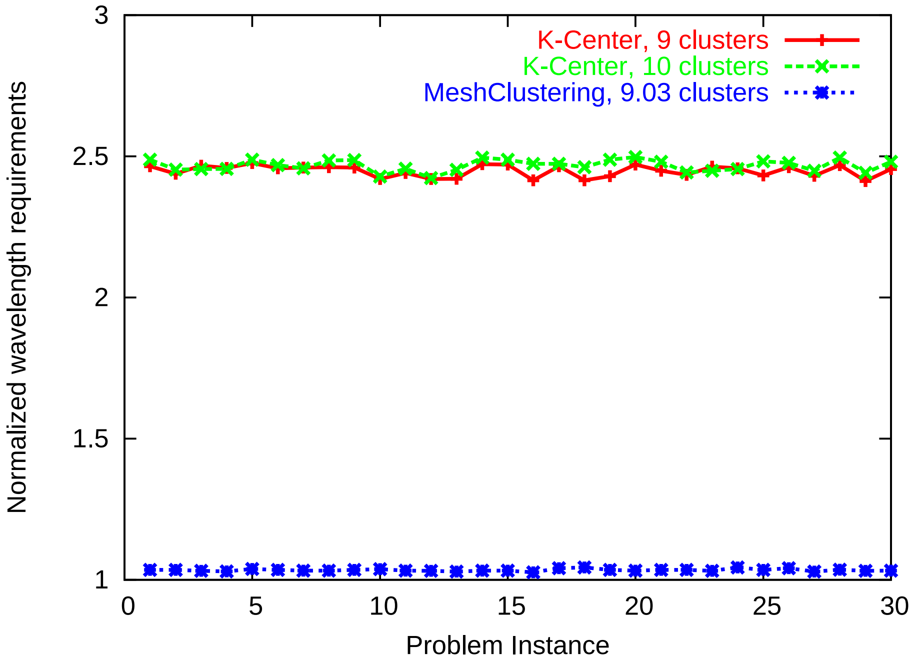


Figure 7.9: Wavelength comparison, rising pattern, 128-node network

Chapter 8

Conclusions

We have studied the static traffic grooming problem with the objective of minimizing the dominating electronic port (LTE) cost, thus minimizing the whole network cost, on networks of elemental topologies as well as those with general mesh topologies.

Studying on the elemental networks helped us understand the characteristics of the problem, and was helpful in our decomposition approaches for coping with more general topology networks. For path, ring and star topology networks, we first gave Integer Linear Formulations to model the problems, so we can program with linear programming software to obtain optimal results for small size instances. Then we made NP-Completeness proofs to show that even in the elemental topologies where lightpath routing and wavelength assignment is in P, the whole problem is still intractable. We then designed heuristic algorithms for solving the grooming problem in rings and stars, respectively. In addition to the objective of minimizing the overall LTE cost, we also studied the objective of minimizing the maximum nodal cost, which results in more balanced design for symmetric topologies. Experimental results have shown that our heuristic algorithms are very effective. We have published some of the results on our research for grooming in the elemental network topologies [10, 7, 8].

We then extended the work to traffic grooming in more general mesh topologies. We proposed a hierarchical grooming approach, to use the divide and conquer method for large-sized networks of arbitrary topology. The approach is to first partition a large network into clusters, then apply a virtual star design for local logical topology of lightpaths; at a higher level, the star hubs form another network and can also be treated with the method we designed for the star networks. We found ways for calculating lower bounds on the lightpath requirement objective, as well as on the wavelength requirement, which are important in feasibility concerns. Numerical results again showed that the hierarchical approach was effective. For larger networks where manual clustering is difficult, we also designed a clustering algorithm especially for the hierarchical grooming problem. We published our work on the framework of the hierarchical grooming approach in [9]. Other results on clustering algorithms are also submitted for publication.

We are currently continuing our study on the performance and improvements of clustering algorithms. Because of the complexity of the traffic grooming problem itself, finding a good clustering is also a difficult task. We are trying to find good methods inspired from different problems such as Traveling Circus, Dominating Sets and Facility Location Problems.

The hierarchical grooming approach also gives us a framework for coping with the *dynamic grooming* problem. Given a new traffic demand element, we can use the existing hub nodes and capacities in the lightpaths to groom the traffic with existing elements efficiently, without re-designing the entire logical topology. This is also an interesting topic for future research directions.

Similar approaches as we use for solving the grooming problem to minimize LTE cost may be used for the placement of other devices as well. For instance, the problem of optimal placement for wavelength converters is proven to be a hard problem as well.

Bibliography

[1] http://www.ilog.com/products/cplex/.

[2] http://www.caida.org.

[3] Eric Angel, Evripidis Bampis, and Fanny Pascual. Traffic grooming in a passive star WDM network. *SIROCCO 2004*, June 2004.

[4] J. Bar-Ilan, G. Kortsarz, and D. Peleg. How to allocate network centers. *J. Algorithms*, 15:385–415, 1993.

[5] P. Baran. On distributed communications networks. *IEEE Transactions on Communications*, 12(1):1–9, March 1964.

[6] R. Berry and E. Modiano. Reducing electronic multiplexing costs in SONET/WDM rings with dynamically changing traffic. *IEEE Journal on Selected Areas in Communications*, 18(10):1961–1971, 2000.

[7] Bensong Chen, Rudra Dutta, and George N. Rouskas. Traffic grooming in star networks. In *Proceedings of the First Workshop on Traffic Grooming*, October 29 2004.

[8] Bensong Chen, George N. Rouskas, and Rudra Dutta. Traffic grooming in WDM ring networks with the min-max objective. In *Proceedings of Networking 2004, LNCS 3042*, pages 174–185, May 9-14 2004.

[9] Bensong Chen, George N. Rouskas, and Rudra Dutta. A framework for hierarchical traffic grooming in WDM networks of general topology. In *Proceedings of IEEE Broadnets 2005*, October 3-7 2005.

[10] Bensong Chen, George N. Rouskas, and Rudra Dutta. Traffic grooming in WDM ring networks to minimize the maximum electronic port cost. *Optical Switching and Networking*, 2(1):1–18, May 2005.

[11] L-W. Chen and E. Modiano. Efficient routing and wavelength assignment for reconfigurable WDM networks with wavelength converters. In *Proceedings of IEEE INFOCOM*, 2003.

[12] H. Choi and D. B. Szyld. Application of threshold partitioning of sparse matrices to markov chains. In *Proceedings of the IEEE International Computer Performance and Dependability Symposium (IPDS)*, 1996.

[13] Hongsik Choi, Nikhil Garg, and Hyeong-Ah Choi. Minimal delay traffic grooming for WDM star networks. *Proceedings of OPTICOMM 2003*, October 2003.

[14] T. Cinkler. Traffic and λ grooming. *IEEE Network*, 17(2):16–21, March/April 2003.

[15] S. Corteel, M. Valencia-Pabon, D. Gardy, D. Barth, and D. Denise. The permutation-path coloring problem on trees. *Theoretical Computer Science*, 297:119–143, 2003.

[16] Z. Ding and M. Hamdi. Clustering techniques for traffic grooming in optical WDM mesh networks. In *IEEE GLOBECOM 2002*, volume 3,17-21, pages 2711–2715, Nov 2002.

[17] D.Li, Z.Sun, X.Jia, and K.Makki. Traffic grooming on general topology WDM networks. *IEE Proc. Commun.*, 150(3):197–201, June 2003.

[18] R. Dutta, S. Huang, and G. N. Rouskas. On optimal traffic grooming in elemental network topologies. In *Proceeding of Opticomm 2003*, October 2003.

[19] R. Dutta, S. Huang, and G. N. Rouskas. Traffic grooming in path, star, and tree networks: Complexity, bounds, and algorithms. In *Proceeding of ACM SIGMETRICS*, pages 298–299, June 2003.

[20] R. Dutta and G. N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks*, 1(1):73–89, January 2000.

[21] R. Dutta and G. N. Rouskas. On optimal traffic grooming in WDM rings. *IEEE Journal on Selected Areas in Communications*, 20(1):110–121, January 2002.

[22] R. Dutta and G. N. Rouskas. Traffic grooming in WDM networks: Past and future. *IEEE Network*, 16(6):46–56, November/December 2002.

[23] Rudra Dutta. *Virtual Topology Design for Traffic Grooming in WDM Networks.* PhD thesis, North Carolina State University, Raleigh, NC, August 2001.

[24] E. S. Elmallah and J. C. Culberson. Multicommodity flows in simple multistage networks. *Networks*, 25:19–30, 1995.

[25] M. Esfandiari, S. Gloeckle, A. Zolfagheri, G. Clapp, J. Gannett, H. Kobrinski, V. Poudyal, and R. Skoog. Improved metro network design by grooming and aggregation of STS-1 demands into oc-192/oc-48 lightpaths. In *NFOEC 2003*, September 2003.

[26] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman and Co., New York, 1979.

[27] O. Gerstel, P. Lin, and G. Sasaki. Wavelength assignment in a WDM ring to minimize cost of embedded SONET rings. In *IEEE INFOCOM*, pages 94–101, 1998.

[28] O. Gerstel, R. Ramaswami, and G. Sasaki. Cost-effective traffic grooming in WDM rings. *IEEE/ACM Transactions on Networking*, 8(5):618–630, October 2000.

[29] H. Ghafouri-Shiraz, G. Zhu, and Y. Fei. Effective wavelength assignment algorithms for optimizing design costs in SONET/WDM rings. *Journal of Lightwave Technology*, 19(10):1427–39, Oct 2001.

[30] T. Gonzalez. Clustering to minimize the maximum inter-cluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.

[31] U. Gupta, D. Lee, and J. Leung. Efficient algorithms for interval graphs and circular-arc graphs. *Networks*, 12:459–467, 1982.

[32] E. Harder, S.-K. Lee, and H.-A. Choi. On wavelength assignment in WDM optical networks. In *Proceedings of the 4th International Conference on Massively Parallel Processing Using Optical Interconnections*, Geneva, Switzerland, April 1997.

[33] John A. Hartigan. *Clustering ALgorithms.* Wiley, 1975.

[34] B. Hendrickson and R. Leland. The chaco user's guide. Technical Report SAND95-2344, Sandia National Laboratories, Albuquerque, NM 87185-1110, July 1995.

[35] D. Hochbaum and D.B. Shmoys. A best possible heuristic for the k-center problem. *Math. Oper. Res.*, 10:180–184, 1985.

[36] D. Hochbaum and D.B. Shmoys. A unified apporach to approximation algorithms for bottleneck problems. *J. ACM*, 33:533–550, 1986.

[37] H.Siregar, H.Takagi, and Y.Zhang. Efficient routing and wavelength assignment in wavelength-routed optical networks. *Proceedings of the 7th Asia-Pacific Network Operations and Management Symposium*, pages 116–127, October 2003.

[38] I.Chlamtac, A. Ganz, and D. Karmi. Lightpath communications: An approach to high bandwidth optical WAN's. *IEEE Transactions on Communications*, 40(7):1171–1182, July 1992.

[39] Prashant Iyer. The complexity of traffic grooming in optical path networks with egress traffic. Master's thesis, North Carolina State University, Raleigh, NC, May 2003.

[40] J.Q.Hu and B.Leida. Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks. *Proceedings of the IEEE INFOCOM 2004*, 23(1):495–501, March 2004.

[41] Zenilton K.G. Patrocínio Jr. and Geraldo Robson Mateus. A lagrangian-based heuristic for traffic grooming in WDM optical networks. In *IEEE GLOBECOM 2003*, volume 5,1-5, pages 2767–2771, Dec 2003.

[42] G. Karypis and V. Kumar. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 1998.

[43] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 29:291–307, 1970.

[44] V. R. Konda and T. Y. Chow. Algorithm for traffic grooming in optical networks to minimize the number of transceivers. In *IEEE Workshop on High Performance Switching and Routing*, pages 218–221, 2001.

[45] Chiewon Lee and E. K. Park. A genetic algorithm for traffic grooming in all-optical mesh networks. In *IEEE SMC 2002*, volume 7, 6-9, Oct 2002.

[46] G. Li. and R. Simha. Efficient routing to reduce the cost of add-drop multiplexers in WDM optical ring networks. In *SPIE Conference*, volume 3843, pages 258–267, 1999.

[47] E. Modiano and P. J. Lin. Traffic grooming in WDM networks. *IEEE Communications*, 39(7):124–129, Jul 2001.

[48] Kirk Schloegel, George Karypis, and Vipin Kumar. A new algorithm for multi-objective graph partitioning. Technical Report 99-003, Dept. of Computer Science, Univ. of Minnesota, September 1999.

[49] David B. Shmoys. Approximation algorithms for facility location problems. *Proc. APPROX 2000, LNCS*, 1913:27–32, 2000.

[50] Koundinya B. Srinivasarao. Traffic grooming in translucent optical ring networks. Master's thesis, North Carolina State University, Raleigh, NC, August 2003.

[51] Mikkel Thorup. Quick k-median, k-center, and facility location for sparse graphs. *Proc. ICALP 2001, LNCS*, 2076:249–260, 2001.

[52] A. Tucker. Coloring a family of circular-arc graphs. *SIAM Journal of Applied Math*, 29:493–502, 1975.

[53] P-J. Wan, G. Calinescu, L. Liu, and O. Frieder. Grooming of arbitrary traffic in SONET/WDM BLSRs. *IEEE Journal on Selected Areas in Communications*, 18(10):1995–2003, 2000.

[54] D. West. *Introduction to Graph Theory.* Prentice Hall, 1996.

[55] G. Wilfong and P. Winkler. Ring routing and wavelength translation. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 333–341, 1998.

[56] Y. Xin, G. N. Rouskas, and H. G. Perros. Physical and logical topology design of large-scale optical networks. *IEEE/OSA Journal of Lightwave Technology*, 21(4):904–915, April 2003.

[57] X. Zhang and C. Qiao. On scheduling all-to-all personalized connections and cost-effective designs in WDM rings. *IEEE/ACM Transactions on Networking*, 7(3):435–445, Jun 1999.

[58] X. Zhang and C. Qiao. An effective and comprehensive approach to traffic grooming and wavelength assignment in SONET/WDM rings. *IEEE/ACM Transactions on Networking*, 8(5):608–617, October 2000.

[59] K. Zhu and B. Mukherjee. Traffic grooming in an optical WDM mesh network. *IEEE Journal on Selected Areas in Communications*, 20(1):122 –133, Jan 2002.

[60] K. Zhu and B. Mukherjee. A review of traffic grooming in WDM optical networks: Architectures and challenges. *Optical Networks*, 4(2):55–64, March/April 2003.