COST-EFFECTIVE ROUTING IN WAVELENGTH DIVISION MULTIPLEXING (WDM) OPTICAL NETWORKS USING SUPER LIGHTPATHS

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Burakhan Yalçın

July, 2006

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Oya Ekin Karaşan (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ezhan Karaşan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Osman Oğuz

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

COST-EFFECTIVE ROUTING IN WAVELENGTH DIVISION MULTIPLEXING (WDM) OPTICAL NETWORKS USING SUPER LIGHTPATHS

Burakhan Yalçın
M.S. in Industrial Engineering
Supervisor: Assoc. Prof. Dr. Oya Ekin Karaşan
July, 2006

In this study, we analyze the routing and wavelength assignment problem for one of the most recent applications of wavelength division multiplexing (WDM) networks, namely super lightpaths. We assume that the traffic is static and each node has the wavelength conversion capability. We try to determine the number of fibers to open for use on each physical link and the routing of the given traffic through super lightpaths so as to minimize the network cost, composed of fiber and wavelength usage components. The problem is proved to be NP-Hard and an integer linear program is proposed as an exact methodology to solve the problem for small scale networks. For larger network sizes, different heuristic approaches are developed. To evaluate the quality of the heuristic solutions, where optimal values are not available, the LP relaxation of the proposed model is strengthened through the use of valid inequalities. The heuristics are tested on a large set of varying network topologies and demand patterns. In terms of the deviation from lower bounds, the heuristic solutions attained are promising.

Keywords: WDM Optical Networks, Super Lightpaths, Routing and Wavelength Assignment Probem.

ÖZET

DALGABOYU BÖLÜŞÜMLÜ ÇOĞULLAMA KULLANILAN OPTİK İLETİŞİM AĞLARINDA SÜPER IŞIKYOLU KULLANARAK MALİYET-ETKİN ROTALAMA

Burakhan Yalçın
Endüstri Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Oya Ekin Karaşan
Temmuz, 2006

Bu çalışmada, super ışıkyolu adı verilen Dalgaboyu Bölüşümlü Çoğullama kullanılan ağların en güncel uygulamaları için rotalama ve dalgaboyu atama problemini inceledik. Trafiğin durağan olduğunu ve her düğümün dalgaboyu değiştirme özelliğine sahip olduğunu varsaydık. Fiber kablo ve dalgaboyu kullanımı maliyetlerinden oluşan ağ maliyetini enazlamak için fiziksel bağlardaki fiber kablo sayısını bulmaya ve verilen trafiği rotalamaya çalıştık. Problemin NP-Hard olduğu ispatlandı ve problemi küçük çaplı ağlarda optimal olarak çözebilmek için bir tamsayı doğrusal programı sunuldu. Büyük çaplı ağlar için ise, çeşitli sezgisel yöntemler geliştirildi. Optimal çözümlerin bulunamadığı durumlarda, sezgisel yöntem çözümlerinin kalitesini değerlendirmek için sunulan modelin gevşetilmiş hali geçerli eşitsizlikler kullanılarak güçlendirildi. Sezgisel yöntemler değişik ağ topolojileri ve talep modelleri için test edildi. Alt sınırdan sapmalar açısından sezgisel yöntem sonuçlarının ümit verici olduğu görüldü.

Anahtar sözcükler: Süper Işıkyolu, Rotalama ve Dalgaboyu Atama Problemi.

Acknowledgement

I would like to express my most sincere gratitude to my advisor and mentor, Assoc. Prof. Oya Ekin Karaşan for all the trust and encouragement during my graduate study. She has been supervising me with everlasting interest and great patience for this research.

I am also grateful to Assist. Prof. Ezhan Karaşan for his invaluable guidance, remarks and recommendations.

I am also indebted to Osman Oğuz for accepting to read and review this thesis and for his invaluable suggestions.

I want to thank to Yunus Emre İlkorkor, Utku Koç and Barış Nurlu for their invaluable friendship. The only ordering I can make among them is the surname order.

Yunus Emre has helped me to make my time worthwhile when I am not studying. I appreciate his belief in my abilities regardless of the subject.

Utku has been always supportive during my studies. His helps on my studies as well as my motivation means a lot to me.

Baris has always been a good confidant. I will always appreciate his support and friendship.

I would never forget to thank to Güneş Erdoğan. Whenever I consult him with even not properly shaped question marks in my mind, I returned with solutions and clearly shaped smileys.

Last but not the least, I would like to thank my family. Without their trust and belief in me I would never be able to complete this thesis.

Contents

1 Introduction **1**

1.1 Lightpaths 3

1.2 Light Trails 6

1.3 Super Lightpaths 10

1.4 Other Issues About ONs 11

1.4.1 Traffic Pattern 11

1.4.2 Wavelength Converters 12

1.4.3 Hop Length 13

1.5 Scope of This Thesis 14

2 Literature Review **15**

2.1 Lightpaths 16

2.2 Light Trails 20

2.3 Super Lightpaths 23

3 Problem Definition **25**

3.1 The Integer Linear Program 27

3.1.1 Assumptions 27

3.1.2 Notation 28

3.1.3 Decision Variables 28

3.1.4 Parameters 29

3.1.5 The Model 30

3.2 NP-Hardness Proof 34

4 Improving The Lower Bound 37

4.1 Valid Inequality 38

4.2 Most Violated ST-Cut Algorithms 38

4.2.1 Most Violated ST-cuts over the Number of Wavelengths . 41

4.2.2 Most Violated ST-cuts over the Number of Fibers 45

4.3 Experimental Results 49

5 The Proposed Heuristic Methodology 56

5.1 LP Relaxation Based Algorithms 57

5.1.1 Whole Super Lightpath at Once 58

5.1.2 Single Arc at Once 59

5.2 Multicommodity Network Flow and Super Lightpath Assignment Problem 61

5.2.1 Multicommodity Network Flow (MCNF) Problem 62

5.2.2 Processing Data Acquired From MCNF Problem 63

5.2.3 Super Lightpath Assignment Problem 66

5.3 Application of Tabu Search 70

5.4 Experimental Results 75

6 Conclusion 81

List of Figures

1.1 Structure of a fiber cable. 3

1.2 Data transmission capabilities of lightpaths. 4

1.3 Data transmission capabilities of light trails. 7

1.4 Data transmission capabilities of super lightpaths. 10

1.5 Wavelength continuity. 12

List of Tables

4.1 Experimental results for ST-Cuts without Valid Inequalities (VI-1) 51

4.2 Summary of Table 4.1 (Average values) 52

4.3 Effect of traffic pattern on the performance of ST-Cut Algorithms 52

4.4 Experimental results for ST-Cuts with Valid Inequalities (VI-1) . 53

4.5 Summary of Table 4.4 (Average values) 54

4.6 Effect of traffic pattern on the performance of ST-Cut Algorithms with (VI-1) cuts 54

4.7 Effect of (VI-1) cuts on the performance of ST-Cut algorithms . . 55

5.1 Results for Algorithm 2, Algorithm 3 and 2-Stage Algorithm . . . 78

5.2 Performance of Tabu Search for three different initial solutions . . 79

5.3 Average gap and CPU times for Table 5.1 80

5.4 Average percent improvements and CPU times for Tabu Search . 80

5.5 Performance of overall solution approach 80

5.6 Optimal Results 80

Chapter 1

Introduction

The focus of this thesis is the Routing and Wavelength Assignment (RWA) problem in optical networks. The problem has a wide range of variations depending on the underlying network structures, restrictions imposed by the technological limitations and different objectives. In this chapter, the available network structures and possible network restrictions as well as the historical development of optical networks will be discussed. In Chapter 2, previous studies related to the RWA problem with different network structures and different objectives will be reviewed. Subsequently, problem definition and the proposed integer linear program to solve the problem exactly will be presented in Chapter 3. Chapters 4 and 5 are for presenting the approaches and corresponding experimental studies performed in order to improve lower and upper bounds of the problem, respectively. Finally, Chapter 6 is a conclusion chapter, in which the results of the thesis and possible areas of future research are discussed.

Telecommunication Networks have been subject to dramatic transformation during the last decade, and especially the last few years. The driving force for this transformation is the demand. The change in demand affects telecommunication networks in two ways. Firstly, increase in the amount of demand requires more and more capacity every day. Indeed, the Internet traffic has been doubling every 4 to 6 months [27]. In addition to the Internet traffic demand, the voice traffic increases as well, due to the decrease in the costs in the competitive telephone service market. The second effect is on the architecture of the network, which is caused by the change in the type of the traffic. The dominating traffic type becomes the data rather than the voice. The factors mentioned above triggered the deployment of optical networks. As the name implies, optical networks (ONs) are the networks in which optical fibers are utilized to transmit data, instead of copper wires. Optical fibers have some certain advantages over copper wires. First of all, they are not affected by electro magnetic interferences. They also have higher bandwidth, which means higher capacity for carrying data. Finally, they provide a higher speed transmission, since the transmission is done in light form. Aforementioned properties of optical fibers made them the best candidate for carrying the increasing Internet traffic.

Former implementations of optical networks (First-Generation ONs) can be viewed like transitions from the copper wire networks to optical networks, that is, some advantages of optical networks have been utilized but not completely. Namely, only the high capacity transmission of fiber links are utilized, whereas, all the other issues such as routing and switching are still done by electronic devices at the nodes. Hence, an optical signal can not pass through a node without being processed even though the node is not the destination. Any received optical signal should first be transformed into an electric signal in every node in order to process (routing, switching, etc.) the signal. After being processed, it should again be transformed into optical signal in order to route it through the optical fibers. However, as the data transmission rates get higher, switching electronically becomes harder and decreases the efficiency of the network utilization.

In order to overcome the limitations of former implementations, new networks, called wavelength-routing networks (Second-Generation ONs), are developed. These networks are the result of complete transformation to optical networks, that is, in addition to the transmission through optical fibers, routing and switching are also performed in optical domain [27]. Hence, there is no need for any optic-electric or electric-optic switching in order to pass through a node, which ultimately leads to an increased network utilization.

Initial implementations of second-generation ONs are facilitated by the utilization of Wavelength Division Multiplexing (WDM) technology. WDM technology is used to transmit data simultaneously at multiple wavelengths in a single fiber. This means a single fiber is utilized as if it is composed of several fibers whose capacities add up to the capacity of that fiber. With such a help of fiber division, the whole fiber does not have to be dedicated to a single demand. Rather, a single demand can be transmitted on a wavelength leaving the other wavelengths available for use. Obviously, WDM technology increases the capacity and ultimately decreases the probability of blocking demand (increase in quality of service).

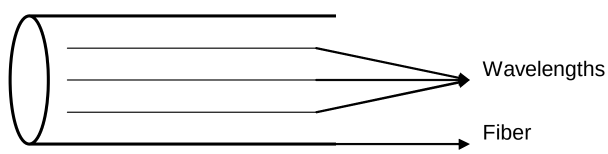

Figure 1.1: Structure of a fiber cable.

The forms of applications of second-generation ONs are lightpaths, light trails and super lightpaths. In all three forms, WDM technology as well as all the other optical network advantages such as higher bandwidth and higher transmission speed are utilized. On the other hand, these forms differ in terms of the techniques employed for using a wavelength, which means they have different utilization levels of a single wavelength leading to different capacity utilizations on the whole network. Below are detailed explanations of these forms of applications:

1.1 Lightpaths

A lightpath is a path originating from the source node and terminating at the destination node, for which the same wavelength is reserved at every link it passes through. This restriction on which wavelength to use is called *wavelength continuity constraint* and it requires that if a lightpath is assigned to a specific wavelength at the first fiber it passes through, then it has to be assigned to the same wavelength at each fiber throughout its route. At this point, another important feature of wavelength-routing networks arises. This feature implies that a lightpath is not processed at any intermediate node, rather it is only routed towards another node in optical domain by a network device called Optical Crossconnect (OXC) that is located on each node. Therefore, costly and slow optical-electronic-optical conversion becomes redundant. If the current node is not the destination of an incoming optical signal, then it is routed through the signal's predetermined path in optical domain by the help of OXCs. Consequently, a ligthpath can be viewed as a dedicated channel that directly connects the source node and the destination node. The intermediate nodes do not perform any process on the lightpath, which means no data can be added to the lightpath or no data can be dropped at any intermediate node. The following figure depicts the data transmission capabilities of a lightpath:

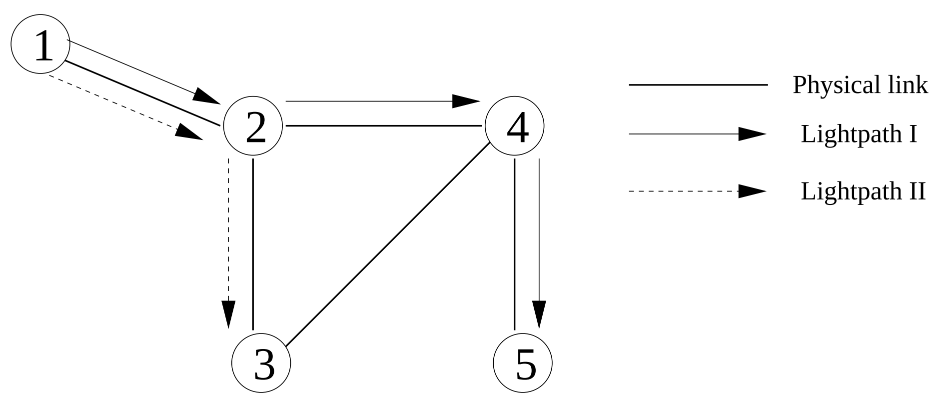

Figure 1.2: Data transmission capabilities of lightpaths.

In this example, lightpath I can carry data only from node 1 to node 5 and lightpath II can carry data from node 1 to node 3. That is, lightpath I can not carry data to node 2 even though it passes through that node.

Routing and Wavelength Assignment Problem associated with lightpaths in its most general form can be defined as:

Input

- The underlying graph representing the optical network G=(V,E), where V is the node set $V = \{1, 2, ..., |V|\}$ and E is the edge set $E \subseteq \{\{i, j\} : i, j \in V, i < j\}$
- Indexed traffic set $K = \{1, ..., |K|\}$, where each k in K corresponds to a traffic pair (s_k, d_k). s_k and d_k represents the source node and destination node of traffic pair k, respectively.
- Set of available wavelengths on each fiber $W = \{1, ..., |W|\}$
- F_e number of fibers available on each link $e \in E$

Output

For all $k \in K$ find a path, say P_k, from s_k to d_k in G and assign a wavelength to this path say P_k^w where $w \in W$

such that

$|\{k : P_k \text{ uses link e and } P_k^w = w\}| \leq F_e$, $\forall e \in E$, $\forall w \in W$

The formulation above implies that solving the RWA problem associated with lightpaths is equivalent to mapping each traffic pair to a path, which uses the same wavelength on the fibers it passes through. The source and the destination nodes of the path is the source and the destination node of the mapped traffic pair, respectively. That is, a direct connection between source and destination nodes of the traffic pair is constructed, which is called a lightpath. Hence, the RWA problem associated with lightpaths tries to construct lightpaths in order to transmit all necessary traffic. During this process, the route of the lightpath should be determined too. Nevertheless, this is not a straightforward task, since the length of the route is important as well as the availability of wavelengths on any of the links in the route. The shortest route may not be selected for a lightpath due to the lack of available wavelengths in one of the links in this route. The RWA problem for lightpaths is proved to be NP-Hard [9] with the assumptions that wavelength continuity constraints must be satisfied, traffic is static (the traffic matrix is known apriori and does not change over time) and there are equal number of wavelengths available at each link. The proof is done by showing that a simplified version of the problem (assigning wavelengths to prerouted lightpaths) is also NP-Hard.

Although networks using lightpaths improve the capacity by using wavelength division multiplexing technology, which is dividing a fiber into wavelengths, it has some limitations too. These limitations arise from the fact that a lightpath is a channel between the source and the destination nodes and the channel is closed to any intermediate node in terms of affecting the data carried by the lightpath. Therefore, data can be added to the lightpath only at the source node and the added data can be accessed only by the destination node. If the size of data to transmit is around the capacity of the wavelength, then there is no problem. However, if the size of the data is small compared to the capacity of the wavelength, then there arises a capacity under-utilization problem, because the lightpath is dedicated to that data, which means no other data request can use that lightpath. This limitation is overcome by light trail technology, which is explained in the next section.

1.2 Light Trails

A new technology, called Optical Time Division Multiplexing (OTDM), offers the opportunity of utilizing wavelength capacities more efficiently. As mentioned, wavelength division multiplexing (WDM) technology provides the access to different parts of a fiber separately, that is wavelengths can be used independently from each other. Similarly, OTDM technology provides access to different parts of a wavelength separately. That is, we can think of a wavelength as it is composed of slots, which can be used independently. Hence, a wavelength can be used for carrying different data packages at the same time as long as the sum of their bandwidth requirements does not exceed the capacity of the wavelength. The process of integrating different data packages onto a single lightpath is called *grooming*.

There are two types of grooming: dedicated-wavelength grooming (DWG) and shared-wavelength grooming (SWG) [15]. In a DWG network, only the demands, which have the same source and destination, can share a lightpath. However, in the SWG network, demands with different sources and destinations can share a lightpath. In this case, the lightpaths are called the light trails. A light trail is a unidirectional optical bus between nodes that allows intermediate nodes to access the bus [16]. Specifically, a light trail is a lightpath, where the intermediate nodes can add data to the specific lightpath in order to send data to subsequent nodes in the route, and read data that are destined to them. That is, each intermediate node can act like both source and destination nodes. Hence, a single light trail can provide up to C(t,2) (t choose 2) number of connections as long as the wavelength capacity is not exceeded, where t is the number of nodes that the light trail passes through [10]. The following example represents the data transmission capabilities of a light trail:

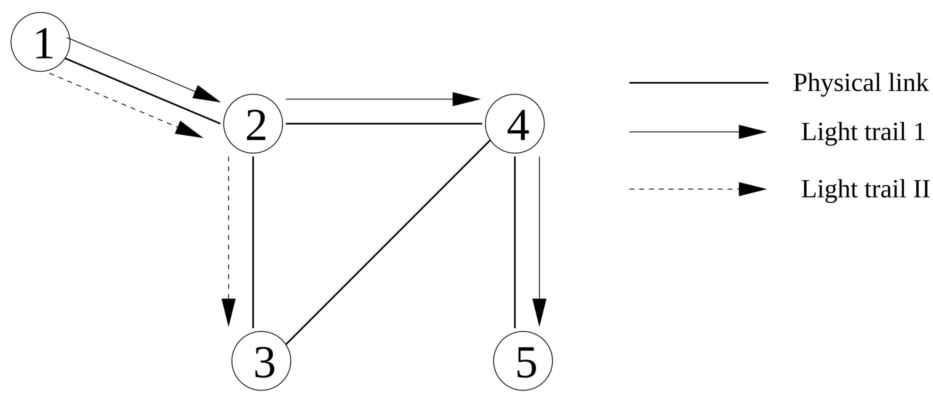

Figure 1.3: Data transmission capabilities of light trails.

In this example, depicted by Figure 1.3, light trail I can be accessed by nodes 1, 2 and 4 in order to add data to the light trail. The data added by node 1 can be read by nodes 2, 4 and 5. However, the data added by node 2 can be read by nodes 4 and 5, but not by node 1. Hence, data added at a node can be read by only the nodes downstream of the light trail. Consequently, light trail I can carry data between nodes (1,2), (1,4), (1,5), (2,4), (2,5) and (4,5) as long as the capacity of a wavelength is not exceeded by the sum of the data added. So, light trail I can accommodate up to 6 connections, which is equal to C(4,2). Similarly, light trail II can carry between nodes (1,2), (1,3) and (2,3).

The RWA problem associated with light trails can be stated in its most general form as follows:

Input

• The underlying graph representing the optical network G=(V,E), where V is the node set $V = \{1, 2, ..., |V|\}$ and E is the edge set $E \subseteq \{\{i, j\} : i, j \in V, i < j\}$

• Indexed traffic set $K = \{1, ..., |K|\}$, where each k in K corresponds to a traffic pair (s_k, d_k). s_k and d_k represents the source node and destination node of traffic pair k, respectively.

• Set of available wavelengths on each fiber $W = \{1, ..., |W|\}$

• F_e number of fibers available on each link $e \in E$

• C is the capacity of a wavelength

• D_k is the amount of traffic between s_k and d_k, $\forall k \in K$

Output

Divide set K into M mutually exclusive and complementing subsets (K_m where $m \in \{1, .., M\}$). For all $m \in \{1, .., M\}$ find a path (to correspond to a light trail), say P_m, that visits all s_k and d_k nodes for all $k \in K_m$ and assign a wavelength to this path, say P_m^w where $w \in W$.

such that

$K_m \cap K_l = \emptyset$, $\forall m, l \in \{1, .., M\} : m \neq l$

$\cup_{m=1}^{M} K_m = K$

P_m visits s_k before d_k for each traffic pair $k \in K_m$, $\forall m \in \{1, .., M\}$

$|\{m : P_m \text{ uses link e and } P_m^w = w\}| \leq F_e$, $\forall e \in E$, $\forall w \in W$

$\sum_{k \in K_m} D_k \leq C$, $\forall m \in \{1, .., M\}$

The formulation above implies that solving the RWA problem associated with light trails is equivalent to mapping a subset of traffic set to a path, which uses same wavelength on the fibers it passes through. The path has to visit all the source and destination nodes of the traffic entries in the corresponding subset in order to satisfy their traffic requirements. This problem is more complicated than the one associated with lightpaths, because in this problem there is another issue to be decided, that is, which data packages to groom on each light trail, that is, constructing K_m. Moreover, this decision affects the routing decisions, since the light trail has to visit all the nodes, which are going to add data to it and read the data added. The RWA problem associated with light trails is proved to be

NP-Hard [26] with the assumptions that wavelength continuity constraints must be satisfied, traffic is static, traffic between two nodes can not be split to carry with different light trails and there exists at most one fiber on any link of the network.

The grooming is performed at each node by a device called optical add/drop multiplexer (OADM). Such devices increase the network cost, so a new concept called sparse grooming capability has emerged. Sparse grooming capability means achieving similar network resource utilizations by using less number of OADMs. Mukherjee et al. [2] show that through careful network design, a sparse-grooming WDM network can achieve similar network performance as a full-grooming network, while significantly reducing the network cost.

Contrary to the tremendous advantages of the light trails, there is a limitation. This limitation arises from the fact that accessing a light trail at an intermediate node in order to add data needs synchronization. Because, if some data is going to be inserted to a specific portion of the light trail, then a temporary empty light trail is constructed at the node, where the portions of the empty light trail, which corresponds to the target portions of the original light trail, is filled with the data to be inserted. Then, when the light trail is passing through the optical add/drop multiplexers (OADM) the temporary light trail is also passed through OADM simultaneously, which means all the portions of two light trails match. Finally, OADM combines them. This process requires synchronization of the node to the light trail. Since a light trail can visit several nodes through its route and several light trails can pass through the same node, all of the nodes must be synchronized. Maintaining synchronization of the network is both expensive and difficult. A solution to this problem is proposed by Gaudino et al. [25], namely super lightpath technology, which is explained in the next section.

1.3 Super Lightpaths

The advantage of super lightpaths related to the synchronization is due to the fact that reading data from a light trail does not require synchronization, since the light trail is not modified during the reading process. Hence, what Gaudino et al. [25] propose is that a super lightpath would be generated at the source node and none of the intermediate nodes would add data to it, however they can read data from it. This means that, several data packages can be groomed onto a single super lightpath if their sources are the same. Since there is no grooming activity at any intermediate nodes, there is no need for synchronization. On the other hand, since grooming of several data packages onto a single super lightpath is possible, network capacity is utilized efficiently. Consequently, super lightpaths are similar to lightpaths in the sense that they have the same single source restriction, which means, there is no synchronization constraint. On the other hand, they are similar to light trails in the sense that they have the same multiple destinations opportunity, which means, capacity utilization is efficient like the light trails. This approach might not produce optimal solutions in terms of capacity utilization, nonetheless it avoids all the technological issues related to network synchronization with acceptable capacity utilization.

A super lightpath can accommodate up to (t-1) connections as long as the wavelength capacity is not exceeded, where t is the number nodes that the super lightpath visits. The following figure depicts the data transmission capabilities of super lightpaths:

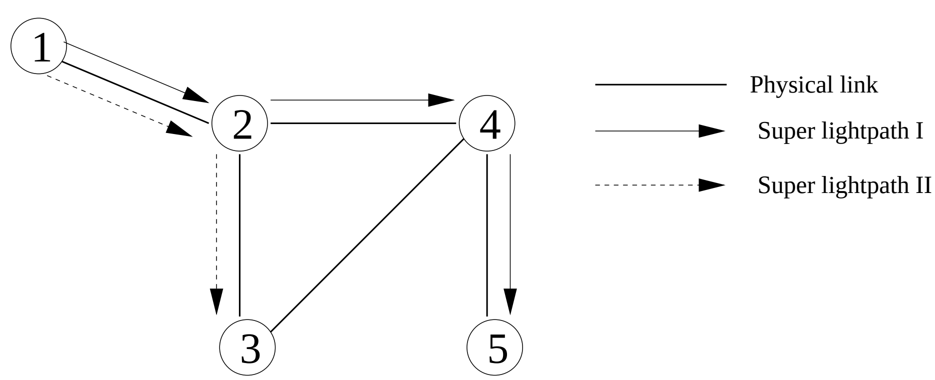

Figure 1.4: Data transmission capabilities of super lightpaths.

In this example, super lightpath I can distribute only the data packages originating from node 1. Hence, it can only carry data between the nodes (1,2), (1,4) and (1,5). Similarly, super lightpath II can carry data between the nodes (1,2) and (1,3).

The definition of the RWA problem associated with super lightpaths is the same as the definition of the problem associated with light trails, except that the definition of K_m must be modified so that the source nodes of each traffic pair in K_m are the same. Hence the following line has to be added to the *such that* part: $s_k = s_l$, $\forall k, l \in K_m$, $\forall m \in \{1, .., M\}$.

This thesis is mainly about the implementation of super lightpaths. More specifically, it is about the routing and wavelength assignment problem associated with super lightpaths.

1.4 Other Issues About ONs

In this section, some of the issues that are necessary to define a RWA problem are discussed. That is, if there is no information on whether the traffic scheme is known apriori or not, then the problem definition will not be precise. This is also valid for whether the wavelength continuity constraint is relaxed or not and whether there is a limit on the one-hop distance. For any RWA problem, all of these issues have to be specified. Hence, these issues are discussed in detail in the following subsections.

1.4.1 Traffic Pattern

The Routing and Wavelength Assignment problems can vary due to the nature of the traffic requirements between nodes. If the traffic requirements change over time, then the traffic is said to be dynamic. On the other hand, if they do not change at all or change slightly over time then the traffic is said to be static. Typically, static demand assumption is used when the problem involves designing the network.

In our study, the traffic pattern is assumed to be static.

1.4.2 Wavelength Converters

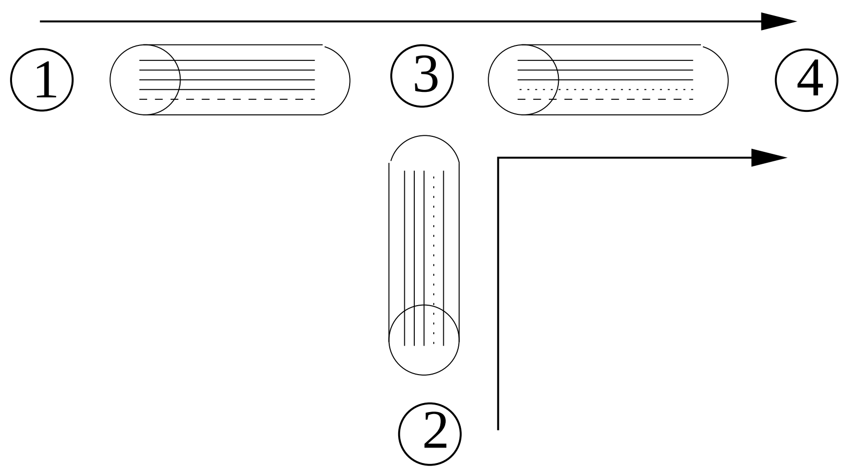

Figure 1.5: Wavelength continuity.

In Figure 1.5, the dark arrows represent the lightpaths, which are constructed between nodes 1-4 and 2-4. The dashed lines are the wavelengths that the first lightpath, which is constructed between nodes 1 and 4, uses at each link. Similarly, the dotted lines are the wavelengths that second lightpath uses.

Wavelength continuity constraint implies that lightpaths are carried on the same wavelength throughout their route, that is, if the lightpath I in Figure 1.5 is assigned to the first wavelength of the link 1-3, then it must be assigned to the first wavelength of the link 3-4 and similarly it must be assigned to the first wavelength on all the fibers it passes through. Since no two lightpaths can use the same wavelength on the same fiber, no two lightpaths can use the same wavelength if they share at least one fiber. That is, lightpath II cannot use first wavelength of the link 2-3 because if it does, it must use the first wavelength of the link 3-4, too. However, the first wavelength of the link 3-4 is assigned to the lightpath I. So, the second wavelength is assigned to the lightpath II. Nevertheless, wavelength reusability is achieved by the capability of using the same wavelength for two link disjoint lightpaths.

Dedicating a wavelength to a lightpath throughout its path would result in inefficient utilization of the network resources, because a wavelength that is free at each fiber in the route may not be available to assign to a lightpath, although the fibers are not fully utilized. This limitation is overcome by the utilization of wavelength converters. If a wavelength converter is available at a node, then the wavelength continuity constraint is relaxed, that is, the lightpath can be assigned to a different wavelength at this node. Hence, assuming that every node has conversion capability, theoretically the fiber capacity can be fully utilized.

Since wavelength converters increase the network cost, some studies are performed in order to minimize the number of nodes that have converters while still maintaining the same capacity utilizations as full conversion networks. These kinds of networks are called sparse conversion networks.

Within the context of this thesis, we relax the wavelength continuity constraint, which means all nodes are assumed to have wavelength converters.

1.4.3 Hop Length

Since the power of an optical signal depreciates as it travels through the network, there might be problems about losing the signal before it reaches the destination node, especially if the network is wide. In these cases, transmitting the data to the destination node at once may not be possible. In order to solve this problem, the message should be regenerated at an intermediate node. Some studies in the literature consider this fact as an additional constraint by limiting the length of the lightpath to some threshold.

Our study does not include hop length limitations, that is, we assume that each node is in one-hop distance to every other node in the network.

1.5 Scope of This Thesis

The problem studied in this thesis is routing and wavelength assignment problem associated with super lightpaths, where the traffic is static, every node has wavelength converters and every node is within one-hop distance to every other node. Since super lightpath is a new concept, there are not many previous studies on the RWA problem associated with super lightpaths. There were only two papers [5], [25] before this study, which are going to be discussed in the next chapter in detail. None of them proposes exact solution approaches. They propose heuristic solutions and they do not try to find how much their solutions deviate from the optimal solution. Hence, there is no information about the quality of their solutions. However, in this thesis an integer linear program is developed to solve the problem optimally. For the cases, where finding an optimal solution is impossible due to computational complexity of the ILP, lower bounds are generated using various algorithms to determine the quality of the solutions gathered by approximate algorithms. Furthermore, there are studies about the complexity of the routing and wavelength assignment problem associated with lightpaths and light trails, but there are no studies on the complexity of the RWA problem associated with super lightpaths. In this thesis, this issue is also considered and the problem with the assumptions we make is proved to be NP-Hard in Chapter 3.

Chapter 2

Literature Review

Optical networks have gained vital importance due to the tremendous increase in the Internet traffic. Because, optic fibers provide higher bandwidths, higher transmission speed and less susceptibility to electro magnetic interference, they have become an answer to the increasing bandwidth requests. Therefore, design of optical networks, as a research topic, drew the attention of many researchers. This attention continued even after the transition to optical networks from copper wire networks, because new applications about optical networks have emerged, such as, wavelength division multiplexers, optical add/drop multiplexers, wavelength converters, etc. In order to utilize the capacity provided by these technological advances more efficiently, Routing and Wavelength Assignment (RWA) problem has emerged and has been studied by many researchers.

This thesis discusses the RWA problem associated with one of the most recent technologies associated with optical networks, namely the use of super lightpaths. However, since it is a relatively new topic, there are not many studies concerning it in the literature. Therefore, in this chapter, studies about the previous applications of optical networks will also be discussed in order to present the development of optical networks till the introduction of super lightpaths. The first application to be discussed is lightpaths.

2.1 Lightpaths

Application of lightpaths are facilitated by wavelength division multiplexing technology (WDM), which means dividing a fiber into sub-channels called wavelengths that are to be used independently. Obviously, with WDM technology, capacity utilization has become more efficient. Furthermore, with the help of optical cross-connect (OXC) technology a lightpath does not have to be converted to electrical signal on any node, which avoids costly and slow optic-electric-optic conversions. The RWA problem associated with lightpaths is widely studied in the literature.

Although using lightpaths provide fast transmission and high capacity, it has some limitations, especially if the sizes of transmission requests are smaller than the wavelength capacities. Since, a lightpath can only accommodate a single data package, there arises a limitation on the utilization of network capacity.

Choi et al. [12] present a functional classification of RWA schemes for the static traffic case. They think of the RWA problem as two separate problems. First problem is routing and the second one is wavelength assignment. Then, they classify the algorithms used to solve these subproblems and they provide an overview of these algorithms. For each subproblem, the algorithms are divided into two subclasses called *search* and *selection* type algorithms and they further divide selection type algorithms into two subclasses called *sequential* and *combinatorial*. The sequential algorithms are the greedy algorithms. The combinatorial type algorithms are further divided into two subclasses called *heuristic* and *optimal* algorithms. These classifications are identical for both subproblems. After defining the classifications, they compare different algorithms in each class. Their study can not propose a clean winner among the algorithms, but they present some insights about classes, which can lead to reasonable choices among candidate algorithms [12].

Jaumard et al. [20] consider the RWA problem on general topology without wavelength conversion capability. They also assume that there is no limit on the hop length. They divide the problem into two cases according to the structure of the traffic matrix: symmetric and asymmetric. When the matrix is symmetric, the links are bidirectional. On the other hand, if the matrix is asymmetric then the links are directional. For the symmetric matrix case they compare the performances of two different integer linear programming formulations using link (flow) formulations and path formulations, and they show that the objective value of the continuous relaxation of link formulation is always greater than or equal to the objective value of the continuous relaxation of path formulation. For the asymmetric matrix case, they compare two different formulations of Krishnaswamy and Sivarajan [21] and they show that both formulations yield the same relaxation results [20].

Chen and Banarje [7] study routing and wavelength assignment problem on general topology where there is no wavelength converter on any node. They consider both dynamic and static traffic cases. They come up with a graph reformation technique in order to overcome the difficulty of having no wavelength converter. They transform the physical topology into a so-called *layered-graph*, which is the core of the solution approach for both dynamic and static traffic cases. The main property of the layered-graph is that if the paths formed in the layered-graph are disjoint then they can be supported by the physical network topology. For dynamic traffic case, they construct an ILP with an objective of minimizing blocking probability, which is the probability of rejecting a new transmission request due to lack of available wavelengths. They use this ILP as a part of so-called layered-graph based dynamic RWA algorithm. For the static traffic case, they consider two different traffic cases, namely uniform and non-uniform. Uniform traffic implies traffic demands between nodes are the same, whereas non-uniform traffic implies randomly generated traffic demands. They develop a multicommodity 0-1 flow based formulation with an objective of maximizing network throughput, that is the total amount of traffic transmitted through network for both cases. For the uniform case, the objective is equivalent to maximizing number of lightpaths established. Since the ILP is intractable for big networks, they propose heuristics which combine greedy and layered-graph approach. They compare the performance of their solution approach with a greedy heuristic and show that their method outperforms the greedy heuristic [7].

Cavendish et al. [6] focus on specific type of network structure, namely mesh networks, where failure of a node does not block the data transmission over the network, since there are at least two paths between each node pair. Their main objective is to minimize the blocking probability, however since achieving this objective directly is difficult they propose four different solution approaches with distinct objectives, which indeed have led to lower blocking probabilities. The network they study is supposed to have limited wavelength conversion capability. First objective is to minimize the number of wavelength conversions. They develop an ILP to solve this problem when traffic matrix is known apriori. However, due to its computational complexity, it is not tractable for large networks. Hence they propose heuristics, which are applicable for both static and dynamic traffic. They use the heuristic they propose also for the remaining objectives with some modifications. The remaining objectives are minimizing the number of wavelengths used, minimizing the hop count, which means minimizing the number of times we need to regenerate a signal due to the hop length restriction, and minimizing the use of scarce resources such as wavelengths available on a link or wavelength conversion capability at a node. They compare the results of their heuristic for minimizing the number of wavelength conversions and their ILP, and they show that the algorithm ends up with less efficient in terms of number of wavelength conversions, but equivalent in terms of hop count. Furthermore, they compare the results of their four algorithms in terms of blocking probability, average number of hops per request and average number of wavelength conversions per request and they propose that the objective of minimizing the number of wavelengths used, which is commonly used in literature, gives the worst results in terms of blocking probability [6].

Lee et al. [22] focus on the RWA problem on ring networks, where each node is connected to two other nodes in order to form a ring. They assume that there is no wavelength conversion capability at any node. Their reason for selecting ring networks is that ring networks are not as efficient as mesh networks but they have simple routing policy, simple control and management, simple hardware system and simple protection from failures. They assume static traffic, that is, the traffic requirements are known apriori. They develop an ILP to solve the problem and they propose an algorithm to solve this ILP efficiently. They first try to solve the LP relaxation using column generation technique, and then in order to get the integral solution they use branch and price approach. After they test their solution approach on various ring networks they show that LP relaxation of their model gives a tight lower bound on the optimal objective value of the RWA problem [22].

Phung et al. [13] consider the RWA problem over a general topology with full wavelength conversion capability and static traffic. They propose a two-stage heuristic in order to minimize the number of wavelengths used. At first step they generate the first K shortest paths (KSP) for each source-destination pairs. At the second stage they generate an ILP in order to select the suitable shortest paths for each source-destination pairs in order to minimize the number of wavelength used. After they applied their approach on NFSNET with 14 nodes and 21 bidirectional links they come up with the result that the time complexity is not affected dramatically by the constant K, however performance on reaching optimal results are significantly improved by the increase of constant K. Hence, their approach achieves significantly better performance in terms of time complexity while still being able to reach optimal results [13].

Quang and Lee [18] focus on limited wavelength conversion capability on networks with general topology. They assume that the traffic is dynamic that is lightpath requirements may vary over time. They propose an algorithm called Congestion Avoidance and Lambda-Run-based (CALR) in order to minimize the blocking probability. Their algorithm splits the RWA problem into two subproblems, namely routing and WA (wavelength assignment) problems. At the first step they use an algorithm that they call Link Congestion Avoidance (LCA). LCA tries to route a new connection request so as to both minimize the total fiber distance and balance the load on each fiber. At the second step they use another algorithm that they call Heuristic Lambda-Run-based (HLR). HLR aims to minimize the number of required converters. Therefore, the main algorithm (CALR) is sequential application of these two algorithms. After they simulate their heuristic as well as other available heuristics in the literature on both small and large networks they come up with the result that their heuristic (CALR) gives the best results in terms of blocking probability [18].

Bertsekas and Ozdaglar [4] consider networks with full conversion, no conversion and sparse conversion capability. They construct integer linear programs to solve the RWA problem for these three cases with both static and dynamic traffic with the objective of minimizing network cost. The network cost is the total cost of using a link for a lightpath. They show that their model generates integer solutions for most of the cases even when the integrality constraints are relaxed. For the cases, where their model can not generate integer solutions they provide a rounding algorithm to round the fractional parts of the solution to integer. They present sample results for some special networks and prove the optimality of their results [4].

Chlamtac et al. [8] consider wide area fiber optic networks with wavelength conversion capability. Their objective is to minimize cost, which is composed of two components: routing cost and conversion cost. They propose an algorithm in order to perform routing and wavelength assignment optimally within short time periods [8].

2.2 Light Trails

The capacity limitations arising from the usage of lightpaths is overcome by a new technology called Optical Time Division Multiplexing (OTDM). OTDM lets a wavelength to accommodate multiple data packages as long as the capacity of the wavelength is not exceeded. A lightpath with the capability of carrying multiple data packages and adding and/or dropping data packages at intermediate nodes is called a light trail. So, by using light trails, network capacity can be utilized efficiently when compared to using lightpaths. However, adding data to a light trail at an intermediate node requires the node to be synchronous with the light trail. Since a light trail passes through several nodes and different light trails can pass through the same node, all the nodes in the network should be synchronized. Maintaining synchronization in the network is difficult and costly.

Balasubramanian et al. [3] study the problem of designing networks with no wavelength conversion case. Since the problem involves designing network, they assume static traffic. They also assume that traffic requirements between two nodes are smaller than the wavelength capacity and can not be split. They restrict the length of a light trail (limited hop length) due to the power loss of the signal through its route. Hence, some connection requests can not be carried out directly with a single light trail, rather they are first carried to a hub node and then to the destination node with different light trails. These hub nodes are like the other nodes, except that a special grooming hardware is located at them. They have two different objectives, namely, maximizing throughput for a given number of hub nodes and minimizing the number of wavelengths and hub nodes used while carrying all the traffic. For both of the objectives, they develop integer linear programming (ILP) models. Besides these models they utilize some heuristic approaches (H-node Selection, Hubbing, and Trail Routing and Wavelength Assignment). Finally, they use simulation in order to see the performance of their approach and conclude that with only a small number of hub nodes, high network throughput and good wavelength utilization can be achieved [3].

Gumaste et al. [11] study a variation of light trail networks, which is, clustered light trail (CLT) networks. A CLT is a tree-shaped variant of light trail. They consider any given network and traffic matrix that may vary over time, but has essentially average flows over large time intervals. They develop a linear program in order to minimize the number of wavelengths used. After the simulation study they propose that for dynamic demand pattern light trails are really efficient, on the other hand for static demand pattern lightpaths are better. Therefore they emphasize that it is possible to move from light trail communication to light-path communication as needed, since the light trail communication also supports lightpath communication [11].

The problem Fang et al. [14] study is minimizing the number of light trails used to carry the given traffic for a given network where hop length is limited and none of the nodes has the wavelength conversion capability. They assume that each link has only one fiber however there is no limit on the number of wavelengths that a fiber can have. The solution approach that they adopt has two phases. In the first phase, the traffic matrix is preprocessed in order to divide the traffic entries, whose source and destination nodes are beyond one hop length, into multiple hops to satisfy hop length constraint. That is, an intermediate node is selected such that the distance between source node and the selected node is within one hop length as well as the distance between the selected node and the destination node. Then, the traffic between the source node and the selected node is increased by the amount of the initial traffic. The traffic between the selected node and the destination node is also increased by the same amount and the initial traffic is cancelled. At the second phase an ILP formulation is developed in order to minimize the number of light trails that are required in the network [14].

Li et al. [23] try to minimize the number of wavelengths on general topology without any wavelength converter. Nevertheless, they do not consider routing. So, they come up with traffic grooming problem (TGP), which they define as: "given a set of t connections, their routes and the grooming factor g, find an optimal wavelength assignment and grooming such that the number of wavelengths required in the network is minimized". Grooming factor is defined as the maximum number of connections that can be groomed on a light trail. They develop an ILP to solve the problem, however since the problem (TGP) is proved to be NP-Hard in the paper, they propose a heuristic solution based on binary search and LP relaxation of ILP. Furthermore they perform a simulation study in order to analyze the relationship between grooming factor and the number of wavelengths on a fiber. As a result, they find that application of traffic grooming can significantly decrease the number of wavelengths used in the network [23].

Hu and Leida [19] focus on mesh topologies with no wavelength conversion capability. They study grooming, routing and wavelength assignment (GRWA) problem in order to minimize the number of wavelengths used in the network. They develop an ILP as well as a decomposition method which divides GRWA into grooming and routing (GR) and wavelength assignment (WA). The decomposition method is not only much more efficient in terms of solution times but also yields optimal results under some sufficient conditions that they provide. This makes the decomposition method a good candidate to be used for large optical mesh networks (with a few hundred nodes and fiber spans) [19].

Mukherjee and Zhu [26] propose an ILP to maximize throughput for irregular mesh topologies, which are less symmetrical compared to general mesh topologies. They assume that there is no wavelength conversion capability and traffic pattern is static. Besides the ILP, they provide heuristics [26].

The grooming is performed at each node by optical add/drop multiplexers (OADMs). These devices increase the network cost, so a new concept, namely sparse grooming capability has emerged. Sparse grooming capability means achieving similar network resource utilizations by using less number of OADMs. Mukherjee et al. [2] show that through careful network design, a sparse-grooming WDM network can achieve similar network performance as a full-grooming network, while significantly reducing the network cost. In their study, they provide an ILP to solve the problem exactly and a heuristic method to obtain sparse-grooming capability for static demand [2].

2.3 Super Lightpaths

Synchronization problem about the light trails can be solved by restricting the data transmission flexibility of light trails. That is, if data can be added to a light trail only at the source node, but data can be read at several nodes in the route, synchronization problem can be overcome, because reading data from a light trail does not require synchronization. A light trail of a single source node, where data can be added only at one node, but there are multiple destination nodes is called a super lightpath. Consequently, super lightpaths do not require synchronization like light trails, on the other hand they provide better capacity utilization than lightpaths. The RWA problem associated with super lightpaths is not studied by many researchers.

Gaudino et al. [25] study this problem on general topology without any wavelength converters. They call the RWA problem with super lightpaths *super routing and wavelength assignment* (S-RWA) problem. Two different greedy algorithms are applied to solve the S-RWA problem in order to minimize the number of wavelengths used. Algorithms are super shortest path first fit (S-SPFF) and super maximum fill (S-MF). Their studies show that using super lightpaths yield large reductions in the number of wavelengths required compared to using lightpaths [25].

Calafato et al. [5] consider the RWA problem on general topology where there is no wavelength conversion capability and traffic pattern is dynamic. They call the RWA problem with super lightpaths *routing, time and wavelength assignment* (RTWA). They extend two heuristics existing in the literature, First-Fit Alternate (FF-ALT) and First-Fit Least-Congested (FF-LC), in order to solve RTWA. Moreover, they come up with a new heuristic. They define a cost, which estimates the impact of accommodating a traffic request. Then, they select the one with minimum cost among all possible allocation solutions. Finally, by simulation, they show that using super lightpaths either reduces the network costs or significantly improves the network performance compared to using lightpaths [5].

The two papers above are the most related papers to our study, however, they also differ in some ways. First of all, they both consider the case with no wavelength conversion capability, whereas our study assumes that there exist wavelength converters at each node. Hence, a comparison between the results of our study and their studies is not possible.

Chapter 3

Problem Definition

The problem we study is a variation of the routing and wavelength assignment problem associated with the super lightpaths. Hence, we consider an all optical network with no electrical-optical or optical-electrical switch during transmission of the super lightpaths. The topology of the network is not restricted, that is, our problem is defined for any given network. However, we assume that there is no limitation on the number of fiber cables that can be opened for use on any link of the network, which means that we can determine the fiber cable requirements for each link without any upper bound on the number of fiber cables. This can be justified as we are leasing the necessary number of fiber cables on each link, where we know that there are excessive number of fiber cables available to lease that are already installed by the leasing company. Another assumption that we make about the network is that all nodes have the grooming capability, that is, every node can construct super lightpaths. Otherwise, a super lightpath that carries data to a single destination node is a lightpath, which would yield less efficient capacity utilizations. Moreover, for further improvement in capacity utilizations, we assume that each node has wavelength conversion capability. By the help of wavelength conversion, we can relax the wavelength continuity constraint, which will lead to a more efficient utilization of available wavelengths. These two capabilities are facilitated by the use of two different network equipments, namely optical time division multiplexing (OTDM) devices and wavelength converters. Availability of these two equipments at each node can be justified, since once we lease a fiber, these equipments are provided by the leasing company. Last assumption we make about network structure is that the super lightpaths can be transmitted without any need for regeneration of the signal, that is, every node in the network is reachable from any other node within one hop distance. This assumption can be valid for the networks that are not very wide.

Since a super lightpath uses wavelengths to transmit data, it's capacity is equal to the capacity of a single wavelength. Furthermore, since we can accommodate several traffic requests on a single wavelength, we can assume that a wavelength is composed of slots that can be accessed independently, and the capacity of a wavelength is equal to the number of slots available on the wavelength. Therefore, the bandwidth requirements of the traffic requests are also assumed to be in terms of the number of slots that they require.

We have two assumptions about the traffic pattern. First, the traffic pattern is static, which means that traffic requirements are known apriori and are not subject to change in time. This can be thought as we are considering the traffic requirements of different branches of a company in the long run. Although there may be some little variations in the traffic requirements in a daily basis, if we calculate the requirements in the long term, the traffic pattern can be thought as static. Second assumption is that the traffic between two nodes can be split integrally and routed with different super lightpaths originating from the source node. Obviously, the capability of splitting the traffic provides a good packing of super lightpaths. Let the capacity of a super lightpath be 3 units and assume that we have to transmit data from a node to three different nodes with each of size 2 units. So, without traffic split, we would need to construct three super lightpaths, whereas if traffic split is allowed 2 super lightpaths would be sufficient.

The network cost is assumed to be composed of two elements. First one is the fiber cable cost, which is incurred when a fiber cable is opened for usage no matter what percent of its capacity is being used. The costs of two fibers are assumed to be the same if they are installed on the same link, which makes sense, since the cost of a fiber is typically determined according to its length. This cost can be justified as the leasing cost of a fiber cable. The second one is the transmission cost, that is the cost of transmitting a super lightpath through a link. This cost can be thought as the cost of occupying a wavelength on a fiber, which is assumed to be the same for all the wavelengths at all fibers.

After discussing the assumptions, our problem can be thought of as the following: Assume that we are assigned to construct a network that will facilitate the communication between different branches of a company, where each branch is located at a different city. We calculate the traffic requirements of these branches in the long run. Now that we know the traffic requirements, we have to find resources to realize this traffic flow. Assume that there is a company which owns a network that is covering all the cities that we are concerned with. This company has already installed excessive number of fiber cables on each link of their network and they lease these fiber cables on demand at a certain price. Once it leases a fiber cable it provides wavelength conversion and grooming equipments as well. Now, we have to determine how many fiber cables to lease at any link of the underlying network in order to route all traffic requirements with minimum cost.

We propose an ILP to solve the problem optimally. The ILP is defined in the next section.

3.1 The Integer Linear Program

3.1.1 Assumptions

The assumptions that are discussed in the beginning of this chapter are summarized below:

- There is no restriction on the number of fibers on any link.
- The traffic is static.
- Every node has the wavelength conversion capability.

• All nodes have the grooming capability.
• Traffic between two nodes can be split integrally and routed with different super lightpaths.
• There is no hop length limitation.
• Fiber cost is the same for the fiber cables that are installed at the same link.
• The cost of occupying a wavelength is the same for all wavelengths.

3.1.2 Notation

Let G = (V,E) be the network topology where V is the node set and E is the edge set. In our problem the direction of edges are important in terms of wavelength usage. Hence, we define the arc set $A = \{(k,l) \cup (l,k) : \{k,l\} \in E\}$. Let D be the traffic matrix, where D_{kl} represents the amount of traffic that has to be routed from node k to node l and $D_{kk} = 0$ for all $k \in V$. The wavelength capacity and the bandwidth requirements are mapped to integers in our model. So, wavelength capacity, flow values and all the entries of traffic matrix (D_{kl}) are integers. The traffic matrix is not restricted to be symmetric, that is, D_{kl} may not be equal to D_{lk}. Furthermore, let t be the maximum number of super lightpaths that a node is allowed to construct. This restriction is not imposed by technological limitations, rather it is calculated according to the traffic matrix in order to decrease the computational complexity. The summation of all outgoing traffic from a node is divided by the wavelength capacity and the result is rounded up to the closest integer to get the minimum number of super lightpaths to be constructed at that node. After this calculation is carried out for all nodes, t is set to be the maximum of the calculated numbers among all nodes.

3.1.3 Decision Variables

In order to determine the design with the optimal network cost, we have to decide on the number of fibers and the number of wavelengths used for each arc, that is, we have to decide on the route of each super lightpath. Hence, Y variables are defined to represent the routes of the super lightpaths and F_{kl} is defined to be the number of fibers used on each edge $\{k, l\} \in E$. Note that, a fiber installed on an edge can be used in both directions. The route of a super lightpath depends on the nodes to which it carries data. So, X variables represent the amount of data that a super lightpath carries to any node. And, finally S variables are defined in order to satisfy flow conservation. Below are the definitions of all the necessary decision variables.

X_{ijk} : amount of data carried by j^{th} super lightpath, originating from node i, to node k, where $i, k \in V$ and $j \in \{1, .., t\}$

S_{ijkl} : amount of data that j^{th} super lightpath of node i carries on arc $(k, l) \in A$, where $i \in V$, $j \in \{1, .., t\}$

F_{kl} : number of fibers used at edge $\{k, l\} \in E$

$$Y_{ijkl} = \begin{cases} 1 & \text{if arc } (k,l) \in A \text{ is used by } j^{th} \text{ super lightpath of node i} \\ 0 & \text{otherwise} \end{cases}$$

3.1.4 Parameters

The parameters represent the information that is available and necessary to solve the problem. For example, how much traffic has to be transmitted between any two nodes has to be given to the model. Moreover, in order to find the minimum cost network configuration, the costs of opening a fiber and occupying a wavelength in a fiber have to be known. Furthermore, since the number of wavelengths in a fiber is limited as well as the capacity of a wavelength in term of the amount of data that it can carry, these limiting values have to be known. Consequently, the following parameters have to be determined and given to the model:

D_{kl} : amount of traffic that has to be transmitted from node k to node l

L_{kl} : cost of opening a fiber at edge $\{k, l\} \in E$.

C : wavelength capacity, i.e, the maximum amount of data that can be groomed to a super lightpath

W : number of wavelengths available in a fiber

α : cost of occupying a wavelength in a fiber

3.1.5 The Model

Our problem is to transmit all necessary traffic by using super lightpaths with minimum network cost. Hence, the objective is to minimize the network cost, which is composed of fiber and wavelength usage costs. The fiber cost is the summation of the costs of each fiber that is opened. Therefore, it can be defined as $\sum_{\{k,l\}\in E} F_{kl} \times L_{kl}$. The other cost component is the total number of wavelengths occupied at each fiber multiplied by the cost of occupying a single wavelength. Hence it can be defined as: $\alpha \times \sum_{i\in V}\sum_{j=1}^{t}\sum_{(k,l)\in A} Y_{ijkl}$. Indeed, these two terms are not independent. Because, number of wavelengths used at an edge in both directions determines the number of fibers to open at that edge. For example, let a fiber has 4 wavelengths. If totally 5 wavelengths have to be used at edge $\{k,l\}$ in the union of two directions, then 2 fibers have to be opened at that edge. Hence, we have to construct this relationship in the constraint set. Also, we have to ensure that all traffic requirements are fulfilled and the capacity of a wavelength is not exceeded. Finally, we have to ensure the flow conservation. Hence, the ILP representing the objective function and constraints mentioned above is presented below:

(ILP-1)

$$\text{Min} \sum_{\{k,l\}\in E} F_{kl} \times L_{kl} + \alpha \times \sum_{i\in V}\sum_{j=1}^{t}\sum_{(k,l)\in A} Y_{ijkl}$$

$$(1) \quad \sum_{k\in V} X_{ijk} \le C, \quad i \in V, j \in \{1,..,t\}$$

$$(2) \quad \sum_{j=1}^{t} X_{ijk} = D_{ik}, \quad i,k \in V$$

$$(3) \quad \sum_{i\in V}\sum_{j=1}^{t}(Y_{ijkl} + Y_{ijlk}) \le W \times F_{kl}, \quad \{k,l\} \in E$$

$$(4) \quad \sum_{l:(i,l)\in A} Y_{ijil} - \sum_{l:(l,i)\in A} Y_{ijli} \le 1, \quad i \in V, j \in \{1,..,t\}$$

$$(5) \quad \sum_{l:(l,k)\in A} Y_{ijlk} - \sum_{l:(k,l)\in A} Y_{ijkl} \ge 0, \quad j \in \{1,..,t\}, i,k \in V : k \ne i,$$

(6) $\sum_{l:(i,l)\in A} S_{ijil} - \sum_{l:(l,i)\in A} S_{ijli} = \sum_{k\in V} X_{ijk}, \quad i \in V, j \in \{1,..,t\}$

(7) $\sum_{l:(l,k)\in A} S_{ijlk} - \sum_{l:(k,l)\in A} S_{ijkl} = X_{ijk}, \quad j \in \{1,..,t\}, i,k \in V : k \neq i$

(8) $S_{ijkl} \leq C \times Y_{ijkl}, \quad i \in V, (k,l) \in A, j \in \{1,..,t\}$

(9) $S_{ijkl} \geq Y_{ijkl}, \quad i \in V, (k,l) \in A, j \in \{1,..,t\}$

(10) Y_{ijkl} binary

(11) $S_{ijkl}, X_{ijk}, F_{kl}$ integer

Constraint (1) is the wavelength capacity constraint, that is it ensures that the capacity of a wavelength can not be exceeded on any super lightpath originating at any node. (2) implies that the sum of the transmitted parts, which are carried with different super lightpaths, of a traffic request between two nodes has to be equal to the traffic requirement presented in the traffic matrix. Hence, it makes sure that all the traffic requirements are fulfilled. Constraint (3) is used to construct the relationship between the number of wavelengths used at edges in any direction and the number of fibers that has to be opened at that edges. First, the number of wavelengths used on arcs (k,l) and (l,k) is calculated, then summation of them is divided by the number of wavelengths available in a single fiber. Finally, the result is rounded up to the closest integer to find the necessary number of fibers. Hence, $F_{kl} = \left\lceil \frac{\sum_i \sum_j (Y_{ijkl} + Y_{ijlk})}{W} \right\rceil$ is calculated for each $\{k,l\} \in E$. (4) ensures that a super lightpath can have only one source node.

Constraints (5) - (9) are conservation constraints, where (5) is the super lightpath conservation of the nodes other than the source node in terms of arc usage variables (Y_{ijkl}). It guarantees that, if the node is not the source, then it can not have negative super lightpath balance, which means that super lightpath can only pass through that node or it can be terminated at that node. (6) is the flow conservation constraint for the source node in terms of the flow variables of super lightpath (S_{ijkl}). It enforces that a super lightpath is loaded with the total amount of data that it should drop on its way. Constraint (7) is the flow conservation for the nodes other than the source node. It ensures that a node, other than the source node, has the flow balance equal to the amount of data that the super lightpath should drop at that node. Constraints (8) and (9) are used to construct the relationship between arc usage and flow variables associated with a specific arc. If there is a positive flow of a super lightpath at an arc, then this means that the arc is used by that super lightpath and vice versa ($S_{ijkl} > 0 \longleftrightarrow Y_{ijkl} = 1$). Finally (10) and (11) are domain constraints. Defining X_{ijk} as an integer variable, rather than binary, lets the model to divide single traffic entry into smaller parts and route them with different super lightpaths. Also, traffic requirements that are greater than the wavelength capacity can also be transmitted by dividing them into smaller parts.

There are some further underlying relations between the decision variables, which are discussed in the following propositions and conjecture.

Proposition 1: If S_{ijkl} takes integer values $\forall i, j, k, l$ then X_{ijk} takes integer values $\forall i, j, k$ even though integrality constraints are relaxed for X_{ijk}.
Proof:
Case 1: (k = i)
Due to (2) of (ILP-1): $\sum_j X_{iji} = D_{ii} = 0$, $\forall i$
Then, $X_{iji} = 0$, $\forall i, j$
Hence, X_{ijk} takes integer values $\forall i, j, k : k = i$.
Case 2: (k $\neq$ i)
Due to (7) of (ILP-1): $\sum_l S_{ijlk} - \sum_l S_{ijkl} = X_{ijk}$, $\forall ijk$
Since all S_{ijkl} values are integer, summation or subtraction of integers are also integers.
Hence, X_{ijk} takes integer values $\forall i, j, k : k \neq i$

$\square$

Proposition 2: If $Y_{ijkl} \in \{0, 1\}$ $\forall i, j, k, l$ and X_{ijk} takes integer values $\forall i, j, k$ then S_{ijkl} takes integer values $\forall i, j, k, l$ even though the integrality constraints are relaxed for S_{ijkl}.
Proof: Assuming that the constraints, at which S_{ijkl} does not appear are satisfied, we end up with four remaining constraints to satisfy:
(6) $\quad \sum_l S_{ijil} - \sum_l S_{ijli} = \sum_k X_{ijk}, \quad i \in V, j \in \{1, .., t\}$

(7) $\sum_l S_{ijlk} - \sum_l S_{ijkl} = X_{ijk}, \quad i,k \in V : k \neq i, j \in \{1,..,t\}$

(8) $S_{ijkl} \leq C \times Y_{ijkl}, \quad \forall ijkl$

(9) $S_{ijkl} \geq Y_{ijkl}, \quad \forall ijkl$

We have to show that these four constraints form a totally unimodular coefficient matrix and the right hand side values are integer in order prove the proposition. First of all, we know that all X_{ijk} and Y_{ijkl} values are integer, which means that all right hand sides are integer.

Think of constraints (6) and (7). Both can be written as a single constraint by multiplying (6) by -1 as:

$\sum_l S_{ijlk} - \sum_l S_{ijkl} = RHS_{ijk}, \forall ijk$ where RHS represents the right hand side and it gets integer values for all $i,k \in V$ and $j \in \{1,..,t\}$. Let us call the coefficient matrix of this combined constraint A.

For a specific i and j, the submatrix, say A_{ij}, is well known network flow balance matrix and it is known to be totally unimodular [28].

Therefore, $A = \begin{bmatrix} A_{11} & & & \\ & A_{12} & & \\ & & . & \\ & & & . \end{bmatrix}$, which means that A is totally unimodular too, because none of the A_{ij}'s share a column or row.

Constraints (8) and (9) form two identity coefficient matrices.

Then, the coefficient matrix for remaining four constraints is $\begin{bmatrix} A \\ I \\ I \end{bmatrix}$.

If A is totally unimodular then $\begin{bmatrix} A \\ I \end{bmatrix}$ and $\begin{bmatrix} A \\ I \\ I \end{bmatrix}$ are totally unimodular too [29].

□

Conjecture: Our experimental studies on different network topologies and different traffic matrices show that if $Y_{ijkl} \in \{0,1\}$ $\forall i,j,k,l$, then both X_{ijk} and S_{ijkl} takes integer values $\forall i,j,k,l$ even though the corresponding integrality constraints are relaxed. This observation will be used in one of our algorithms, where we determine the binary values of Y variables first and then solve the problem with fixed Y values within a short time period.

3.2 NP-Hardness Proof

Theorem 1: The static RWA problem associated with super lightpaths with wavelength conversion capability (RWA-S) is NP-Hard.
Proof: RWA-S is in NP, since the feasibility of any given solution can be tested by plugging the values into the constraints in polynomial time.
Feasibility version of RWA-S (RWA-SF) is as following:
(RWA-SF)
INSTANCE: Graph G=(V,E), fiber costs L_{kl} for each $\{k, l\} \in E$, wavelength usage cost α, traffic requirements D_{kl} for all $k, l \in V : k \neq l$, wavelength capacity C and maximum number of super lightpaths that a node can construct t.
QUESTION: Are there Y_{ijkl}, S_{ijkl}, X_{ijk} and F_{kl} such that (1)-(11) of (ILP-1) are satisfied with an objective value smaller than or equal to K?

Let us first recall the Hamiltonian Path (HP) problem, which is well known to be NP-complete [17].
(HP)
INSTANCE: Graph G=(V,E)
QUESTION: Does G contain a Hamiltonian path?

Take an arbitrary instance of HP problem, which is defined by a network topology G=(V,E). Let $N = |V|$.
Define the corresponding instance of RWA-SF as follows: Same network topology, $L_{kl} = 1$ for all $\{k, l\} \in E$, $D_{kl} = 1$ for all $k, l \in V : k \neq l$, $W = N$, $C = N - 1$, $t = 2$ and $K = N - 1$.
Let us consider a solution to this specific RWA-SF instance. The edges, at which a fiber is opened, and all the nodes must form a connected subgraph, because there exist traffic requirements between each node pair. Therefore, if the subgraph is not connected, then some of the traffic requirements would not be satisfied. In

order to route all traffic requirements of N nodes by using (N-1) fibers, the edges, at which a fiber is opened, must form a spanning tree. It can not be a cycle, because then there would be N fibers installed due to the connectivity requirement. If it is a tree, but not a path at the same time, then there will be an intermediate node, to which three edges are connected because of the branching of the tree. This means that, the node has to construct 3 super lightpaths to route necessary traffic, which is not allowed by the definition of the instance ($t = 2$). Hence, the edges that have fibers must form a path, which covers all the nodes. Then, the path is called a *Hamiltonian path*. Therefore, given a solution to the RWA-SF instance, we can get a feasible solution to the HP problem in polynomial time.

Now, let us consider a solution to HP problem, $i_1 - i_2 - ... - i_l - ... - i_N$, where '-' represents the edge between nodes.

Let us assume that we install a single fiber on each edge of the given Hamiltonian path. Then, there would be $N - 1$ fibers installed, which complies with the objective of the RWA-SF instance being considered.

Now, we have to route the traffic requirements. Since all $D_{kl} = 1$ given that $k \neq l$, any leaf node on this Hamiltonian path (i_1 and i_N) can route its outgoing traffic of $N - 1$ units by using a single super lightpath, because wavelength capacity (C) is $N - 1$. Therefore, i_1 will construct a super lightpath through i_N and similarly, i_N will construct a super lightpath through i_1. The intermediate nodes have to construct 2 super lightpaths, because they have to route traffic through both i_1 and i_N. So, they construct a super lightpath through i_1 and another one through i_N.

Then, all traffic requirements are satisfied. Now, we have to check whether the number of wavelengths used at each edge is smaller than or equal to the fiber capacity ($W = N$) or not. Because, if it is bigger, then 2 fibers would be necessary on that edge, which leads to the usage of more than $N - 1$ fibers. The super lightpaths originating from the leaf nodes will pass through each fiber, therefore, occupy a wavelength at each fiber. For an intermediate node, one of the constructed super lightpaths will pass through each fiber between the node and i_1, whereas the other constructed super lightpath will pass through each fiber between the node and i_N. Hence, the super lightpaths originating from the intermediate nodes will also occupy a wavelength at each fiber. Consequently, each node will occupy one wavelength at each fiber in order to route its outgoing flow. This means that the number of wavelengths used at each fiber is N, which is equal to W. So, fiber capacity is not exceeded.

Therefore, given a solution to HP problem, we can find a solution to our RWA-SF instance in polynomial time.

Two findings above imply that feasibility version of our problem is NP-complete. Hence, optimality version is NP-hard

$\square$

Since our problem is NP-hard, seeking optimality for big networks is not reasonable. Hence, we propose some heuristics so as to get good feasible solutions, which are discussed in Chapter 5. In order to be able to judge the quality of these solutions, we propose some methods to improve the lower bound for the optimal solution, which are discussed in Chapter 4. The percent gap between the lower bound and the gathered solution will provide us an upper bound on the amount of the deviation from the optimal solution.

Chapter 4

Improving The Lower Bound

The model provided in the previous chapter seeks optimal results, however, solution time increases exponentially as the network size increases. Hence, for big networks, getting solution from the model is almost impossible within reasonable time. For these cases, instead of seeking optimal solution, we seek solutions with low optimality deviations. The algorithms used for this sake are discussed in the next chapter. Approximate solutions can be reached within short time periods, however, we can not be sure about their quality. Therefore, in order to evaluate the quality of the solution, we try to find some lower bounds to the optimal solution. The size of the gap between the lower bound and the solution would give us an idea about the quality of the solution. The focus of this chapter is to achieve strong lower bounds.

One obvious way of getting a lower bound is relaxing the integrality constraints and solving the linear programming relaxation of the problem. However, in general, it generates really weak lower bounds. Hence, it can mislead us about the quality of the solution gathered. So, it is necessary to improve this bound. The improvement can be performed by adding valid inequalities. Two different valid inequalities are discussed in this chapter. The first one is based on the traffic flow between two distinct nodes, whereas the second one is based on the total traffic flow across two mutually exclusive and complementing sets of nodes, namely ST-Cuts [24]. Constraints corresponding to the first type of valid inequalities are added to the (ILP-1) at once and the relaxation of the new model is again solved to get a better lower bound. For further improvement, second type of valid inequalities are used.

4.1 Valid Inequality

This valid inequality considers the traffic between two distinct nodes and determines at least how many wavelengths must be allocated to transmit this traffic. The traffic requirement between nodes i and k is D_{ik} and C is the capacity of the wavelength. Hence, if D_{ik} is divided by C and then the result is rounded up to the closest integer, then the minimum number of wavelengths, which are at the union of arcs whose tails are the node k, necessary to realize the transmission is calculated. Therefore, the sum of the arc usage variable values associated with the arcs whose tails are the node k, for the super lightpaths originating from node i should be greater than or equal to the minimum number of wavelengths as calculated above.

This inequality calculates the minimum number of wavelengths necessary for each node pair and enforces the relaxed model to use at least predetermined number of wavelength by adding a new constraint to the model for each node pair. All the inequalities are added to the model at once and the relaxation is solved to get the new bound. The aforementioned cut is as follows:

$$\sum_{j=1}^{t} \sum_{l:(l,k)\in A} Y_{ijlk} \geq \left\lceil \frac{D_{ik}}{C} \right\rceil \quad i,k \in V : D_{ik} \neq 0 \qquad \text{(VI-1)}$$

4.2 Most Violated ST-Cut Algorithms

ST-Cuts [24] are another way of improving lower bounds. They are based on partitioning the node set and observing the traffic across them. Therefore, the set V must be divided into subsets S and T, such that $S \cup T = V$ and $S \cap T = \emptyset$. Then, the traffic flowing from set S to set T is calculated by adding up the D_{ik} values, where $i \in S$ and $k \in T$. After calculating the total flow from S to T, it is divided by the capacity of a wavelength and finally, the result is rounded up to the closest integer. Then, the minimum number of wavelengths, which connect sets S and T, necessary to realize the transmission is calculated. This calculation is also carried out for the traffic flow from set T to set S. More formally, we let

$$outflow[S] = \sum_{i \in S} \sum_{k \in T} D_{ik}$$
$$inflow[S] = \sum_{i \in S} \sum_{k \in T} D_{ki}$$

The calculated necessary number of wavelengths constitutes a lower bound on the actual summation of wavelength usage variables associated with the wavelengths, which connect sets S and T. Hence, the following inequalities are valid for the model.

$$\sum_{i \in S} \sum_{j=1}^{t} \sum_{(l,k) \in A: l \in S, k \in T} Y_{ijlk} \geq \left\lceil \frac{outflow[S]}{C} \right\rceil$$
$$\sum_{i \in T} \sum_{j=1}^{t} \sum_{(l,k) \in A: l \in T, k \in S} Y_{ijlk} \geq \left\lceil \frac{inflow[S]}{C} \right\rceil$$

The important part of generating ST-Cuts is to determine the sets S and T, because some formations of sets may be very useful to lift the lower bound, whereas some may be unnecessary to try. Adding all possible ST-Cuts at once is infeasible especially for big networks, because the number of possible ST-Cuts increases exponentially as the number of nodes increases in the network. Hence, some sort of an algorithm has to be used in order to determine which formations of sets S and T have to be tried. We used the most violated ST-Cut model for this sake. This is a step-by-step algorithm, where at each repetition of the algorithm one S-T partition is selected and the associated cut is added to the model as a new constraint. The algorithm starts with solving the relaxation of (ILP-1) with all (VI-1) inequalities, then an integer linear program is used in order to determine the sets S and T so that the violation of the cut is maximum. Later, the constraint corresponding to the determined sets S and T is added to the model. This procedure goes on until there exists no ST-Cut violation in the relaxation solution. More formal explanation of the algorithm is presented in Algorithm 1, where v_{max} is the amount of violation gathered from the solution of most violated ST-Cut ILP.

Algorithm .1 Most Violated ST-Cut Algorithm

Solve relaxation of (ILP-1) with (VI-1) inequalities, say LP
$v_{max} \leftarrow 1$
while $v_{max} > 0$ **do**
    Solve most violated ST-Cut ILP with relaxation results to get v_{max}
    if $v_{max} > 0$ **then**
        Add corresponding constraint to LP
        Solve the new LP
    end if
end while

Most Violated ST-Cuts algorithm is a pretty generic algorithm, which can work on any network. However, we can strengthen it by exploiting some properties of the network that we study. Because, when super lightpaths are used to transmit data, the content of the super lightpath can only be altered at the source node, that is, once the super lightpath is constructed at the source node there is no possibility to add data at any other node. Moreover, no two super lightpaths can be combined to form a single super lightpath at any node. So, these two properties ensure that no two super lightpaths can use the same wavelength of the same fiber, which means that we can calculate the necessary number of wavelengths, which connect sets S and T, separately for each node in set S and then sum them up to get the total number of wavelengths necessary that connect sets S and T. This will strengthen the effect of ST-Cuts on the model because of the following relation:

For a fixed S-T partition, define the flow from S to T that originates from node i for each $i \in S$

$$flow[i] = \sum_{l \in T} D_{il}$$

Then, the following relation is valid based on the preceding argument, where the term in the left is the right hand side of generic ST-Cut approach, whereas the term in the right is the right hand side of our ST-Cut approach.

$$\left\lceil \frac{outflow[S]}{C} \right\rceil \leq \sum_{i \in S} \left\lceil \frac{flow[i]}{C} \right\rceil$$

We used another variant of most violated ST-Cut algorithm sequentially for further improvement of its performance. First, the algorithm is executed on the number of wavelengths so that the lower bound is improved. This also improves the number of fibers necessary because of the constraint (3) of (ILP-1). However, since we are still solving the relaxation, we probably get continuous values for number of fibers, which means that there is a room for improvement on the number of fibers used. Hence, the second variant of the most violated ST-Cut algorithm is executed over the number of fibers, which is the same as the first variant, except that the violation values are calculated over the number of fibers. In the next two subsections these two variants are discussed in detail.

4.2.1 Most Violated ST-cuts over the Number of Wavelengths

This algorithm is executed by the help of an ILP, which determines the sets S and T so that the violation is maximized. In order to find the violation, we need two components, actual wavelength usage variable values and the minimum number wavelengths, which connect sets S and T, to be used for a specific S-T configuration. First one is gathered from the solution of the relaxation of (ILP-1) with (VI-1) cuts, and second one is calculated in a similar way explained at the beginning of Section 4.2. The difference arises from the properties of super lightpaths, which enables us to calculate the necessary number of wavelengths, which connect sets S and T, separately for each node in set S and then sum them up to find the total requirement. The outflow value for each node i in set S is calculated as

$$outflow_i = \sum_{l \in T} D_{il}$$

Hence the necessary number of wavelengths, which connect sets S and T, to realize this flow is

$$\sharp wavelength_i = \lceil \tfrac{outflow_i}{C} \rceil$$

where C is the wavelength capacity. After calculating $\sharp wavelength$ for each node in set S, then we sum them up to gather the total requirement:

$$\sharp wavelength[S] = \sum_{i \in S} \sharp wavelength_i$$

After the ILP finds the set S and T so that the violation is maximum by the help of calculations above, the following inequality is added to the model:

$$\sum_{i \in S} \sum_{j=1}^{t} \sum_{(l,k) \in A: l \in S, k \in T} Y_{ijlk} \geq \sharp wavelength[S]$$

After adding this inequality to the model, the relaxation of the model is executed again with the new cut and the procedure above repeats until the most violated ST-Cut ILP finds a maximum violation of 0.

The objective function of the ILP, which is used to find the most violated ST-Cut, is the difference between the calculated necessary number of wavelengths, which connect sets S and T, and the summation of the values of the arc usage variables gathered from the execution of the relaxation of (ILP-1) with (VI-1) cuts. The model determines the sets S and T so that this difference is maximized, hence we need variables that indicates whether the node is in set S or T. Also we need variables that represent the number of wavelengths necessary for each node in set S, where their sum would give the total minimum wavelength requirement. Hence, we define the following variables:

4.2.1.1 Variables

$$X_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

$$P_{ik} = \begin{cases} 1 & \text{if } i \in S \text{ and } k \in S \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{ikl} = \begin{cases} 1 & \text{if } i \in S \text{ and } k \in S \text{ and } l \in S \\ 0 & \text{otherwise} \end{cases}$$

A_i : minimum number of wavelengths, which connect sets S and T, necessary for node i in set S to distribute its outgoing traffic to the nodes in set T

v : total number of wavelengths, which connect sets S and T, necessary for all nodes in set S to distribute their outgoing traffic to the nodes in set T

Besides these variables, we need the amount of traffic between any two nodes, capacity of a wavelength and arc usage variable values gathered from the solution of the relaxation of the current liner program at hand. Hence, we define following parameters:

4.2.1.2 Parameters

D_{ik} : amount of traffic from node i to node k

C : wavelength capacity

$$Y^*_{ijkl} = \begin{cases} 1 & \text{if arc (k,l) is used by the lightpath j originating from} \\ & \text{node i (from LP relaxation)} \\ 0 & \text{otherwise} \end{cases}$$

ϵ : very small positive number $(0 < \epsilon << 1)$

After defining all the variables and parameters we can present the model.

4.2.1.3 The Model

The objective of the model is to find a configuration of sets S and T so that the violation is maximized. Hence, objective function is the difference between total number of wavelengths, which connect sets S and T, of the nodes in set S and the summation of the actual arc usage values gathered from the relaxation solution.

Max $v - \sum_{i \in V} \sum_{j=1}^{t} \sum_{(k,l) \in A} (P_{ik} - Z_{ikl}) \times Y^{*}_{ijkl}$

where v is the total number of wavelengths required. The second part of the objective function represents the summation of arc usage values, where the source of the lightpath is in set S and node k of arc (k, l) is in set S, whereas node l is in set T. This condition is satisfied by the term $(P_{ik} - Z_{ikl})$. Hence, the relations between X_i, P_{ik} and Z_{ikl} have to be constructed mathematically in the formulation. Furthermore, calculation of A_i values and v must be represented in constraint format. Hence, the following constraints should be defined:

(ILP-2)

(1) $P_{ik} \leq X_i, \quad i, k \in V$

(2) $P_{ik} \leq X_k, \quad i, k \in V$

(3) $P_{ik} \geq X_i + X_k - 1, \quad i, k \in V$

(4) $Z_{ikl} \leq P_{ik}, \quad i, k, l \in V$

(5) $Z_{ikl} \leq X_l, \quad i, k, l \in V$

(6) $Z_{ikl} \geq P_{ik} + X_l - 1, \quad i, k, l \in V$

(7) $A_i \geq \frac{1}{C} \times \sum_{l \in V} (X_i - P_{il}) \times D_{il}, \quad i \in V$

(8) $A_i \leq \frac{1}{C} \times \sum_{l \in V} (X_i - P_{il}) \times D_{il} + 1 - \epsilon, \quad i \in V$

(9) $v = \sum_{i \in V} A_i$

(10) $\sum_{i \in V} X_i \geq 1$

(11) Z_{ikl}, P_{ik}, X_i binary and A_i, v integer

Constraints (1),(2),(3),(4),(5) and (6) are used to construct the relations between X_i, P_{ik} and Z_{ikl}, which are the linearizations of

$P_{ik} = X_i \times X_k$

$Z_{ikl} = X_i \times X_k \times X_l$ or $Z_{ikl} = P_{ik} \times X_l$

By the help of these relations we can use $(P_{ik} - Z_{ikl})$ term in the objective function and $(X_i - P_{il})$ term in constraints (7) and (8). The term in the objective function gets the value 1 if nodes i and k are in set S ($X_i = 1$ and $X_k = 1$, hence $P_{ik} = 1$) and node l is in set T ($X_l = 0$, hence $Z_{ikl} = 0$) and gets the value zero otherwise. On the other hand, the term in the constraints gets the value 1 if node i is in S ($X_i = 1$) and node l is in T ($X_l = 0$, hence $P_{il} = 0$) and gets the value zero otherwise. Necessary number of wavelengths for node $i \in S$ to be used at the arcs whose heads are in S and tails are in T is calculated by the constraints (7) and (8) as follows:

$$A_i = \left\lceil \frac{\sum_{l \in V} (X_i - P_{il}) \times D_{il}}{C} \right\rceil$$

Constraint (9) is used to calculate the total amount of necessary wavelengths to be used by all nodes in set S and constraint (10) ensures that set S is not empty. Finally, constraint (11) is the domain constraint.

4.2.2 Most Violated ST-cuts over the Number of Fibers

After applying the first variant, the number of fibers are also updated, because of the directly increasing relation between the number of fibers and the number of wavelengths used on the same edge. The direct relation is due to the relaxation, because in the original model the relation is stepwise increasing. Once the execution of first variant is completed, we know that there is no room for improvement in the arc usage variable values. However, since the result of fiber usage variables from the relaxation could be continuous, we can still improve the lower bound by finding violated ST-Cuts over number of fibers and adding the associated constraints to the model.

This algorithm uses a slightly modified version of the most violated ILP over number of wavelengths. The main reason of the modification is that we can use fibers in both directions. Hence, we have to observe the traffic between sets S and T in both directions (from S to T and from T to S), because two super lightpaths, one originating from a node in S and one originating from a node in T, can use the same fiber. So, the calculation of the necessary number of fibers,

which connect sets S and T, is as follows:
Wavelength requirements for each node are calculated again separately, but this time for all the nodes in both S and T.

$$outflow_i = \begin{cases} \sum_{l \in T} D_{il} & \text{if } i \in S \\ \sum_{l \in S} D_{il} & \text{if } i \in T \end{cases}$$

Wavelength requirements for each node is calculated in the same way:

$$\sharp wavelength_i = \lceil \frac{outflow_i}{C} \rceil$$

And the total wavelength requirement is calculated as follows:

$$\sharp wavelength = \sum_{i \in V} \sharp wavelength_i$$

Then, we have to convert this information to the number of fibers:

$$\sharp fibers = \lceil \frac{\sharp wavelength}{W} \rceil$$

Hence, following inequality will be added to the model as a constraint:

$$\sum_{\{k,l\} \in E : (k \in S, l \in T \;\; or \;\; k \in T, l \in S)} F_{kl} \geq \sharp fibers$$

The procedure is the same as described in Algorithm 1. Again, single cut is added at each repetition of the procedure, till there is no violation in terms of number of fibers to be used.

The objective function of the ILP for this variant is the difference between the number of fibers required between sets S and T and the actual fiber usage variables (associated with the fibers whose one side is in set S and the other is in T) gathered from the relaxation of (ILP-1) with (VI-1) cuts. Hence, we need variables to indicate in which set the nodes are in. Moreover, we again need A_i and v, but this time v represents the number of fibers instead of wavelengths. Therefore, necessary variables are as follows:

4.2.2.1 Variables

$X_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$

$P_{ik} = \begin{cases} 1 & \text{if } i \in S \text{ and } k \in S \\ 0 & \text{otherwise} \end{cases}$

A_i : minimum number of wavelengths, which connect S and T, necessary for node i

v : total number of fibers, which connect S and T, necessary.

We need the same parameters as the first variant case, except that a new parameter to indicate the number of wavelengths in a fiber is necessary.

4.2.2.2 Parameters

D_{ik} : amount of traffic from node i to node k

C : wavelength capacity

W : number of wavelengths in a fiber

F^*_{kl} : number of fibers gathered from the relaxation of (ILP-1) with (VI-1) inequalities.

ϵ : very small positive number ($0 < \epsilon << 1$)

4.2.2.3 The Model

The objective of the model is to find a configuration of sets S and T so that the violation in terms of the number of fibers is maximized. Hence, the objective function is the difference between the minimum number of fibers, which connect sets S and T, required to route traffic between S-T and the actual fiber usage values gathered from the relaxation solution, namely,

$$\text{Max } v - \sum_{\{k,l\} \in E} (X_k + X_l - 2P_{kl}) \times F^*_{kl}$$

where v is the total number of fibers required. The second part of the objective function represents the summation of fiber usage values, where node k and node l are in different sets. This condition is satisfied by the term $(X_k + X_l - 2P_{kl})$.

Hence, the relation between X_k and P_{kl} has to be constructed mathematically in the formulation. Furthermore, calculation of A_i and v values must be represented in constraint format. Hence, the following constraints should be defined:

(ILP-3)

(1) $P_{ik} \leq X_i, \quad i,k \in V$

(2) $P_{ik} \leq X_k, \quad i,k \in V$

(3) $P_{ik} \geq X_i + X_k - 1, \quad i,k \in V$

(4) $A_i \geq \frac{1}{C} \times \sum_{k \in V} (X_i + X_k - 2P_{ik}) \times D_{ik}, \quad i \in V$

(5) $A_i \leq \frac{1}{C} \times \sum_{k \in V} (X_i + X_k - 2P_{ik}) \times D_{ik} + 1 - \epsilon, \quad i \in V$

(6) $v \geq \frac{1}{W} \times \sum_{i \in V} A_i$

(7) $v \leq \frac{1}{W} \times \sum_{i \in V} A_i + 1 - \epsilon$

(8) $\sum_{i \in V} X_i \geq 1$

(9) $\sum_{i \in V} X_i \leq \left\lfloor \frac{|V|}{2} \right\rfloor$

(10) P_{ik}, X_i binary and A_i, v integer

Constraints (1),(2) and (3) are used to construct the relation between X_i and P_{ik}, which is the linearization of $P_{ik} = X_i \times X_k$. By the help of this relation we can use $(X_i + X_k - 2P_{ik})$ term in both objective function and the constraints (4) and (5). The term gets the value 1 if nodes i and k are in different sets ($X_i = 1$ and $X_k = 0$, hence $P_{ik} = 0$ or $X_i = 0$ and $X_k = 1$, hence $P_{ik} = 0$) and gets the value zero otherwise. Constraints (4) and (5) are used to calculate the necessary number of wavelengths for each node both in S and T, because fibers are bidirectional, which means that the same fiber can be used for both routing the outflow and inflow of set S. Total amount of data to transmit from node i to the nodes in the other set is divided by the capacity of the wavelength and the result is rounded up to the closest integer.

$$A_i = \left\lceil \frac{\sum_{k \in V} (X_i + X_k - 2P_{ik}) \times D_{ik}}{C} \right\rceil$$

Constraints (6) and (7) are used in order to calculate the total number of fibers necessary for the resulting S and T sets. The summation of the number of necessary wavelengths for each node is divided by the number of wavelengths per fiber and the result is rounded up to the closest integer:

$$v = \left\lceil \frac{\sum_{i \in V} A_i}{W} \right\rceil$$

Constraint (8) ensures that set S is not empty and constraint (9) is used to prevent the repetition in the sense that S={1,2,3} and T={4,5} is in fact the same as S={4,5} and T={1,2,3} since the fibers are bidirectional. This constraint is not necessary for (ILP-2) because a single wavelength can not be used in both directions at the same time. Hence, two S-T formations mentioned above are not the same for (ILP-2). Finally constraint (10) lists the domain requirements.

4.3 Experimental Results

In this section, the experimental studies performed on the techniques used to improve lower bounds are presented. The techniques used are (VI-1) cuts, most violated ST-Cuts over number of wavelengths and most violated ST-Cuts over number of fibers, which are explained in the previous sections in detail. Now, we are going to compare their performances in terms of the percent improvement in the initial lower bound (LP relaxation solution) and CPU time required. All the models and algorithms are developed in C with using CPLEX 8.1 callable library. The experimentation is performed on a computer with a CPU clock of 1133 Mhz and 1024 MB of real memory.

We used six different network topologies for experimentation. Number of nodes and number of edges for the networks are (5-7), (6-9), (7-11), (8-16), (9-12) and (14-21) respectively. We constructed the topologies of the networks with 5, 6, 7, 8 and 9 nodes, whereas the network with 14 nodes is the well known NFSNet Backbone Network with 21 links [6]. In addition to the different network topologies, we generated four traffic patterns. First pattern is the light demand (LD), in which each traffic requirement is small compared to the wavelength capacity. More specifically, the requirements between each node is generated randomly between 1 and 5 units, where the wavelength capacity is 10 units (C=10). Second pattern is the medium demand (MD), where the traffic requirements are generated randomly between 4 and 8 units. The remaining patterns are the heavy demand (HD) and mixed demand (XD), where the traffic requirements are generated randomly between 8-12 units and 1-12 units, respectively.

We set the number of wavelengths available in a single fiber to be 4 (W=4) and capacity of a wavelength to carry traffic to be 10 units (C=10). Furthermore, the cost of occupying a single wavelength is set to be 1 ($\alpha = 1$), whereas the cost of fibers are determined as the network topology is generated. For NFSNet network with 14 nodes, the actual length of a fiber is assumed to be the cost of it.

Table 4.1 represents the experimental results for ST-Cut algorithms without using (VI-1) cuts, where first column represents the network topology and traffic pattern used. That is, 9-HD means the network with 9 nodes is used with the heavy traffic pattern. The third column (*relaxation*) represents the LP relaxation of (ILP-1) for the corresponding topology and traffic pattern. Fourth column (stY) represents the results of most violated ST-Cut algorithm over number of wavelengths (MVST-W) when applied to the (ILP-1), whereas fifth column (stF) represents the results of most violated ST-Cut algorithm over number of fibers (MVST-F) applied to (ILP-1). Finally, last column ($stYF$) represents the results when two ST-Cut algorithms are applied sequentially. The rows of each cell show the result, the percent improvement from LP relaxation and the CPU time used, respectively. Percent improvement is calculated as $100 \times \frac{(result)-(relaxation)}{(relaxation)}$.

In Table 4.2, which presents a summary of Table 4.1, we see that the average improvement gathered by MVST-W is 48%, whereas it is 39% for MVST-F. This result is as expected, because MVST-W increases the number of fibers too, while trying to lift arc usage variables, due to the relation between number of wavelengths and the number of fibers. Hence, it affects both wavelength usage and fiber costs. On the other hand, MVST-F only affects the fiber cost. Therefore, using MVST-W gives better results, however it requires more CPU time. On the

Input		**Relaxation**	**stY**	**stF**	**stYF**
5-LD	Result	43,6	147,5	129	156
	Percent		238%	196%	258%
	Time(sec)		4	1	5
5-HD	Result	154,75	205	192,8	206
	Percent		32%	25%	33%
	Time(sec)		6	1	6
6-LD	Result	124,85	282,5	218,8	286
	Percent		126%	75%	129%
	Time(sec)		16	2	16
6-MD	Result	283,2	376,1	388,4	406
	Percent		33%	37%	43%
	Time(sec)		32	2	33
6-HD	Result	451,75	560,05	570,8	590
	Percent		24%	26%	31%
	Time(sec)		16	2	18
6-XD	Result	288,7	367,8	389,6	403
	Percent		27%	35%	40%
	Time(sec)		25	2	27
7-LD	Result	200,05	409,5	331,2	412,5
	Percent		105%	66%	106%
	Time(sec)		52	7	54
7-MD	Result	392,9	535,5	536,4	567
	Percent		36%	37%	44%
	Time(sec)		86	5	90
7-HD	Result	668,9	799,1	814,2	840
	Percent		19%	22%	26%
	Time(sec)		73	8	93
7-XD	Result	382,6	529	513	546
	Percent		38%	34%	43%
	Time(sec)		81	9	85
8-LD	Result	228,45	541	430,63	541,6
	Percent		137%	89%	137%
	Time(sec)		180	9	181
8-MD	Result	518,45	644,15	602,2	646,32
	Percent		24%	16%	25%
	Time(sec)		448	37	465
8-HD	Result	806,15	880,23	859,9	886,37
	Percent		9%	7%	10%
	Time(sec)		380	63	422
8-XD	Result	478,65	633,15	578,6	639,4
	Percent		32%	21%	34%
	Time(sec)		330	31	353
9-LD	Result	4821,8	7192	7274,4	7525,5
	Percent		49%	51%	56%
	Time(sec)		551	22	573
9-MD	Result	9173,6	11207,6	12110,6	12175
	Percent		22%	32%	33%
	Time(sec)		508	40	592
9-HD	Result	15583,7	17261	17480,6	17520
	Percent		11%	12%	12%
	Time(sec)		520	42	578
9-XD	Result	10754,1	12651,6	12695,1	12742
	Percent		18%	18%	18%
	Time(sec)		522	30	541
14-LD	Result	13070	19742,74	18767	19994,04
	Percent		51%	44%	53%
	Time(sec)		24777	652	26087
14-MD	Result	28124	31710,57	31126	31734,2
	Percent		13%	11%	13%
	Time(sec)		31739	1902	32925
14-HD	Result	44961,75	47728,04	47121,65	47926,84
	Percent		6%	5%	7%
	Time(sec)		34341	5301	35961
14-XD	Result	30039,7	33692,45	33670	34104
	Percent		12%	12%	14%
	Time(sec)		30583	4592	31623

Table 4.1: Experimental results for ST-Cuts without Valid Inequalities (VI-1)

	stY	**stF**	**stYF**
Percent	48%	39%	53%
Time(sec)	5694	580	5942

Table 4.2: Summary of Table 4.1 (Average values)

average, MVST-W requires 5694 seconds, whereas MVST-F requires only 580 seconds. If both algorithms are used sequentially, then the average improvement gathered increases up to 53%, however the average CPU time required also increases up to 5942 seconds on the average. Hence, by executing MVST-F after MVST-W gives us the chance of increasing the improvement by 5% with the extra CPU time of 248 seconds on the average.

Another observation is about the effect of traffic pattern on the percent improvements of the algorithms. The average improvements gathered by sequential application of MVST-W and MVST-F are presented in Table 4.3 for different traffic patterns.

Demand Pattern	**Percent Improvement**
Light Demand (LD)	123%
Medium Demand (MD)	32%
Heavy Demand (HD)	20%
Mixed Demand (MD)	30%

Table 4.3: Effect of traffic pattern on the performance of ST-Cut Algorithms

So, we can see that the traffic load affects the performance of the algorithm drastically. For the light demand pattern, the improvement is almost 125%, whereas for the heavy demand pattern it is only 20%. This result is due to the fact that, if the traffic load is high, then the wavelengths are utilized better in the relaxation solution, which does not leave much room for the algorithms to improve the solution. The relaxation results of the problem give better solutions as the traffic requirements increase up to the wavelength capacity. Because, in that case, the wavelengths will be used almost fully for each traffic entry.

Table 4.4 represents the performance of the algorithms where (VI-1) cuts are added to the (ILP-1) beforehand. The structure of this table is the same as the

Input		Relaxation	Valid Cut	stY	stF	stYF
5-LD	Result	43,6	146,45	147,5	156	156
	Percent		236%	238%	258%	258%
	Time (sec)		0,02	1	0,03	2
5-HD	Result	154,75	174,35	205	199	206
	Percent		13%	32%	29%	33%
	Time (sec)		0,04	5	1	5
6-LD	Result	124,85	274,81	283,05	286	286,5
	Percent		120%	127%	129%	129%
	Time (sec)		0,04	12	1	12
6-MD	Result	283,2	331,95	380,33	398,53	406,8
	Percent		17%	34%	41%	44%
	Time (sec)		0,08	17	2	18
6-HD	Result	451,75	523,5	567,25	584	591
	Percent		16%	26%	29%	31%
	Time (sec)		0,15	14	3	16
6-XD	Result	288,7	345	376,17	399,14	404,82
	Percent		20%	30%	38%	40%
	Time (sec)		0,07	13	2	15
7-LD	Result	200,05	397,68	409,5	410,69	412,5
	Percent		99%	105%	105%	106%
	Time (sec)		0,20	25	2	26
7-MD	Result	392,9	481,27	536,07	553	567
	Percent		22%	36%	41%	44%
	Time (sec)		0,19	51	5	54
7-HD	Result	668,9	757,31	801,65	828,8	840,4
	Percent		13%	20%	24%	26%
	Time (sec)		0,38	49	7	63
7-XD	Result	382,6	515	545,34	541,03	556,15
	Percent		35%	43%	41%	45%
	Time (sec)		0,30	49	6	53
8-LD	Result	228,45	536,34	544,29	541,47	545,4
	Percent		135%	138%	137%	139%
	Time (sec)		1	124	3	125
8-MD	Result	518,45	611,39	666,03	629,03	667,4
	Percent		18%	28%	21%	29%
	Time (sec)		1	353	22	359
8-HD	Result	806,15	885,77	922,7	895,79	925,78
	Percent		10%	14%	11%	15%
	Time (sec)		1	368	13	377
8-XD	Result	478,65	640,11	662,25	647,86	664,31
	Percent		34%	38%	35%	39%
	Time (sec)		1	161	15	163
9-LD	Result	4821,8	6530,6	7468,93	7472,33	7645,36
	Percent		35%	55%	55%	59%
	Time (sec)		1	418	22	429
9-MD	Result	9173,6	9830,19	11338,05	12136,4	12178,02
	Percent		7%	24%	32%	33%
	Time (sec)		1	375	49	462
9-HD	Result	15583,7	16112,2	17324,2	17499,8	17519,9
	Percent		3%	11%	12%	12%
	Time (sec)		1	510	80	599
9-XD	Result	10754,1	11594,34	12764,17	12727,53	12805,78
	Percent		8%	19%	18%	19%
	Time (sec)		1	464	64	486
14-LD	Result	13070	19337,74	20724,43	20117,19	20751,59
	Percent		48%	59%	54%	59%
	Time (sec)		12	13635	329	13370
14-MD	Result	28124	30331,59	32636,27	31408,1	32642,42
	Percent		8%	16%	12%	16%
	Time (sec)		11	24784	1009	24586
14-HD	Result	44961,75	47324,86	48880,22	47964,91	49020,78
	Percent		5%	9%	7%	9%
	Time (sec)		19	22896	643	23545
14-XD	Result	30039,7	33383,9	35087,85	34559,46	35346,94
	Percent		11%	17%	15%	18%
	Time (sec)		22	17923	1291	18250

Table 4.4: Experimental results for ST-Cuts with Valid Inequalities (VI-1)

Table 4.1, except that there is a new column, which represents the results of adding (VI-1) valid inequalities to (ILP-1).

	VI-1	stY	stF	stYF
Percent	41%	51%	52%	55%
Time(sec)	3	3739	162	3774

Table 4.5: Summary of Table 4.4 (Average values)

In Table 4.5, which presents a summary of Table 4.4, we see that the average improvement on the lower bounds gathered by adding (VI-1) cuts is 41% within 3 seconds of CPU time. After applying MVST-W to the updated model, the improvement increases up to 51%, however the CPU time required is also increases up to 3739 seconds on the average. If we apply both ST-Cut algorithms sequentially we get 55% of improvement within 3774 seconds of CPU time. An interesting observation is that applying MVST-F gives slightly better results than the MVST-W on the average (52% percent improvement within 162 seconds). This means that, adding the (VI-1) cuts improves the performances of both ST-Cut algorithms, however, performance of MVST-W is not improved as much as the performance of MVST-F, because some of the cuts to be added by MVST-F are already covered by (VI-1) cuts.

The observation about the effect of the traffic pattern on the percent improvement of the algorithms is also available in this case (see Table 4.6, where the percent improvement values are gathered by sequential application of both ST-Cut algorithms on (ILP-1) with (VI-1) cuts).

Demand Pattern	**Percent Improvement**
Light Demand (LD)	125%
Medium Demand (MD)	33%
Heavy Demand (HD)	21%
Mixed Demand (MD)	32%

Table 4.6: Effect of traffic pattern on the performance of ST-Cut Algorithms with (VI-1) cuts

In order to observe the effect of (VI-1) cuts on the performance of ST-cut algorithms, we have to compare the results presented in Tables 4.1 and 4.4. The average improvements and the required CPU times of ST-Cut algorithms with and without the (VI-1) cuts are presented in Table 4.7. We can see that, adding (VI-1) cuts improved the performance of the algorithms in terms of both percent improvement and required CPU time. The improvement in terms of percent improvement may be limited, but the improvement in terms of CPU times is important.

		stY	**stF**	**stYF**
with VI-1	Percent	51%	52%	55%
	Time(sec)	3739	162	3774
without VI-1	Percent	48%	39%	53%
	Time(sec)	5694	580	5942

Table 4.7: Effect of (VI-1) cuts on the performance of ST-Cut algorithms

Chapter 5

The Proposed Heuristic Methodology

In this chapter, some approximate algorithms are proposed to get good feasible solutions, because for big networks, the ILP proposed in Chapter 3 can not generate optimal results in reasonable time. The gap between the solutions gathered from the algorithms discussed in this chapter and the lower bounds gathered from the previous chapter would give us an idea on the level of the optimality gap.

Three types of algorithms are proposed in this chapter. Two of them are constructive algorithms, which means that they generate solutions from scratch, and one of them is an improvement algorithm, which means that it starts with an initial feasible solution and tries to move to better solutions by traveling in the feasible region. First one of the constructive algorithms is an LP Relaxation Based Algorithm. As the name implies, it starts with solving the LP relaxation of (ILP-1) and it assigns some selected nonzero arc usage variables (Y_{ijkl}) to 1. It repeats these two steps sequentially till there exists no fractional arc usage variable value in the relaxation solution. Then, finally (ILP-1) is solved with the fixed arc usage variable values and a feasible solution is gathered. Second one is a two-stage algorithm, where in the first stage multicommodity network flow problem is solved and paths are generated by processing the resulting flow values.

Then, in the second stage these paths are assigned to super lightpaths by solving an assignment problem in order to get a solution. Last one of the algorithms is an application of tabu search. As all improvement algorithms, it requires an initial solution. Then, it moves in the feasible region by rerouting a single super lightpath at a time. The details of these algorithms are available in the following sections.

5.1 LP Relaxation Based Algorithms

This is a constructive algorithm, which uses LP relaxation solutions to get a feasible solution. Indeed, this algorithm considers the problem as two interacting stages. First stage is determining the routes of super lightpaths and the second one is assigning traffic requirements to these super lightpaths. The role of this algorithm is to complete the first stage, which means determining the arc usage variable values (Y_{ijkl}). Then, the second stage is carried out by solving (ILP-1) with the arc usage values fixed at the first stage. Due to the conjecture in Chapter 3, second stage yields feasible solutions within a short time period. Because, stage 1 lets Y_{ijkl} variables to get the values 0 or 1 only, which means that X_{ijk} and S_{ijkl} gets integer values too, even though the corresponding integrality constraints are relaxed.

The algorithm starts with the results of relaxation of (ILP-1). The relaxation of (ILP-1) yields a rooted tree for each super lightpath route. This algorithm tries to convert these rooted trees to paths by setting some selected nonzero arc usage variable values to 1 at each repetition of the algorithm. This cycle goes on till there exists no fractional arc usage variable value in the relaxation solution. The selection of the variables, which are going to be set to 1, is performed in two different ways. Details of these are discussed in the next section.

5.1.1 Whole Super Lightpath at Once

The solution of the relaxation of (ILP-1) yields a rooted tree for each super lightpath. We start from the root node of the tree, which is also the source node of the corresponding super lightpath, and travel on the tree such that at each branch we select the arc with the maximum arc usage value and move through that arc till we reach a leaf of the tree. Then, the set of the arcs we pass through forms a path, i.e, super lightpath. This process is repeated for every rooted tree. Then, all the arc usage variables associated with the arcs that are selected to form the super lightpaths are set to 1 in (ILP-1). This completes a pass of the algorithm.

In other words, the arc usage variables (Y_{ijkl}), which are to be set to 1, are selected in such a way that all super lightpaths are constructed at a pass of the algorithm. All of the super lightpaths are constructed separately, that is, one super lightpath is constructed after the previous one is completed. So, after solving the relaxation of (ILP-1), the arc usage variable selection procedure starts with Y_{ijil} type variables, that is, the variables associated with the arcs whose head is the source node of the super lightpath. These type of variables are scanned for the super lightpath under consideration and the one with the maximum arc usage value is selected. That is the arc (i, l) is selected where i is the source node of the super lightpath and $l = argmax(Y^*_{ijik} : (i, k) \in A)$ for the j^{th} super lightpath originating from node i. Then, the procedure moves to node l, which is the tail of the selected arc. After moving, the arc with maximum arc usage value among all the arcs whose heads are node l is selected in the same way described above and the procedure moves to the tail node of the selected arc. This procedure goes on till the node where all the available arcs to select have the arc usage value of 0, that is, at the leaf of the corresponding original rooted tree. Therefore, a super lightpath is completed, then the procedure starts to process another rooted tree to form another path. When all the rooted trees are processed, a pass of the algorithm is over and all the selected arc usage variables are set to 1 in (ILP-1) by setting their lower bound to 1. Then, the relaxation of (ILP-1) is again solved with these fixed arc usage values. Subsequently, the variable selection procedure is executed again and construction of the super lightpaths is continued from where the previous pass of the algorithm left. So, variable selection procedure starts with the variables associated to the node, at which the previous pass of the procedure is stopped moving, for each super lightpath. This cycle goes on till all the values of arc usage variables are either 0 or 1 in the relaxation solution. Algorithm 2 represents these steps more formally, where Y_{sa} is arc usage variable value of super lightpath s on arc a gathered from relaxation solution, $A_m = \{(m,l) : (m,l) \in A\}$, and $tail((k,l)) = l$.

Algorithm .2 Whole Super Lightpath At Once

Initialize the array node[s] with the source node of each super lightpath s
Solve relaxation of ILP-1
while There is at least one fractional Y_{sa} value in relaxation solution **do**
  for all Super lightpath s **do**
    $maxY \leftarrow 1$
    while $maxY > 0$ **do**
      $maxY \leftarrow max(Y_{sb} : b \in A_{node[s]})$
      if $maxY > 0$ **then**
        $a \leftarrow argmax(Y_{sb} : b \in A_{node[s]})$
        $Y_{sa} \leftarrow 1$
        $node[s] \leftarrow tail(a)$
      end if
    end while
  end for
  Solve relaxation of ILP-1 with fixed Y_{sa} values
end while

5.1.2 Single Arc at Once

The procedure for selecting the arc usage variables for this variation is like a step-by-step version of the previous one. That is, only one arc is constructed for only one of the super lightpaths that originate from each node at a single pass of the algorithm. Hence, at most $|V|$ arc usage variables (one for each node to transmit its outgoing traffic) are set to 1 at a single pass. The algorithm starts from a source node, say i. For each super lightpath that originates from i, it checks the arc usage variable values for the arcs whose heads are i. Then it finds the ones with the maximum arc usage value for each super lightpath. Then, among these arcs, the one with the maximum arc usage value is selected. Say Y^*_{ijil} is selected, therefore, the algorithm moves to node l for j^{th} super lightpath originating from i. Next, the algorithm starts to process the super lightpaths that originates from another source node in the same manner. When all other source nodes are processed once, all the arc usage variables for selected arcs are set to 1 and the relaxation of (ILP-1) is solved again with the fixed arc usage variables. Then, it starts to process node i again. This time, it checks the arc usage variables for the arcs whose heads are i for all the super lightpaths originating from node i except the j^{th} one. It checks the arcs whose heads are l for j^{th} super lightpath of i, because the algorithm moved to node l for this super lightpath at the previous pass. The arc with maximum arc usage value among all super lightpaths is again selected and the algorithm moves to the tail node of this arc for the super lightpath that causes this arc usage value. After, all nodes are processed in the same manner, the arc usage variables for selected arcs are set to 1. Then, relaxation of (ILP-1) is solved again with the fixed arc usage variables. The algorithm repeats till there is no fractional arc usage value in the relaxation solution. More formal explanation of the algorithm is presented as Algorithm 3, where the definitions of Y_{sa}, A_m and $tail(a)$ are the same as for the Algorithm 2 and $source[s]$ indicates the source node of super lightpath s.

Algorithm .3 Single Arc At Once

Initialize the array node[s] with the source node of each super lightpath s
Solve relaxation of ILP-1
while There is at least one fractional Y_{sa} value in relaxation solution **do**
    for all $n \in V$ **do**
        $maxY \leftarrow max(Y_{sb} : b \in A_{node[s]}, source[s] = n)$
        if $maxY > 0$ **then**
            $a \leftarrow argmax(Y_{sb} : b \in A_{node[s]}, source[s] = n)$
            $Y_{sa} \leftarrow 1$
            $node[s] \leftarrow tail(a)$
        end if
    end for
    Solve relaxation of ILP-1 with fixed Y_{sa} values
end while

A further improvement in the performance of Algorithm 2 and Algorithm 3 is facilitated by a slight modification in the variable selection procedure. The aim of this modification is to let the flow accumulate at selected arcs. While selecting the arc usage variable with the maximum value, we do not select any of them if the one with the maximum value is smaller than 0,5 and we stop there, that is, we assume that the node as a leaf node of the corresponding rooted tree. Then, we start processing the next rooted tree. After a pass of the algorithm is completed and the relaxation is solved, the arc usage variable values mentioned above may turn out to be zero, because we set some other arc usage variables to 1, although they are smaller than 1. So, this extra capacity may be utilized to accumulate the small flows, which may yield zero arc usage values for the arcs mentioned above. The algorithm is repeatedly executed with the threshold of 0,5 till there is no fractional arc usage value that is greater than 0,5 in the relaxation solution, and then the execution of the algorithm goes on with the threshold of zero till there is no fractional arc usage value in the relaxation solution.

5.2 Multicommodity Network Flow and Super Lightpath Assignment Problem

The logic behind this algorithm is that a super lightpath can be assumed as a combination of several paths, where

- source nodes of each path is the same, because data can be added to a super lightpath only at the source node
- summation of data carried by the paths is smaller than or equal to the wavelength capacity
- the route of the combination is still a path, that is, all the path routes are subsets of the path with the longest route.

If several paths satisfy the restrictions above, then we can say that the traffic carried by all of them can be carried by a single super lightpath, because a super lightpath can carry traffic to multiple nodes from a single source as long as the summation of the amount of the traffic is smaller than or equal to the wavelength capacity. Hence, once the paths are constructed, we can transform them into super lightpaths. In order to construct paths, the multicommodity network flow problem (MCNF) is solved with the same topology and traffic matrix of the original problem. However, the results of MCNF problem are flows rather than paths. So, we convert these flows into paths with an algorithm. Once the paths are formed, second stage can be executed, which is assigning these paths to super lightpaths by solving an assignment problem. Details of these stages are presented in the following subsections.

5.2.1 Multicommodity Network Flow (MCNF) Problem

This is solved by an integer linear program, which is used to generate flows that would satisfy the traffic requirements while minimizing the network cost. Therefore, the only necessary decision variable is related to the amount of flow belonging to any source-destination pair on any arc, where K represents the set of all source-destination pairs.

X_{kl}^{sd} : amount of flow passing through arc $(k,l) \in A$ belonging to $(s,d) \in K$

Before defining objective function and constraints we have to determine the parameters, which are the capacity of each edge, unit cost of flow on each edge and the necessary amount of traffic to be transmitted between any source-destination pair.

C_{kl} : capacity of edge $\{k,l\} \in E$.

L_{kl} : cost of edge $\{k,l\} \in E$ for a unit flow.

D_{sd} : amount of traffic requirement between $(s,d) \in K$ from s to d

Once the decision variables and the parameters are defined, we can define objective function too. Definition of the objective function is important in the sense that it must reflect the objective function of (ILP-1) as much as possible. The objective function of (ILP-1) is composed of two components, namely fiber cost and wavelength cost. It is not possible to reflect these costs completely, because MCNF problem assumes an arc as a single entity, which means that we can not define fibers or wavelengths. Hence, we determine the cost for each edge to be the multiplication of the cost of an edge for a unit flow and the amount of flow passing through that edge in any direction. This does not represent the actual cost structure but it makes the costly arcs unattractive for flows.

In addition to the objective function, the constraints has to be defined, which are flow conservation and capacity constraints. Therefore, the integer linear program is as follows:

(ILP-4)

Min $\sum_{(s,d)\in K} \sum_{\{k,l\}\in E} (X_{kl}^{sd} + X_{lk}^{sd}) \times L_{kl}$

s.t.

(1) $\sum_{l:(k,l)\in A} X_{kl}^{sd} - \sum_{l:(l,k)\in A} X_{lk}^{sd} = 0, \quad (s,d) \in K, k \in V : k \neq s, k \neq d$

(2) $\sum_{l:(k,l)\in A} X_{kl}^{sd} - \sum_{l:(l,k)\in A} X_{lk}^{sd} = D_{sd}, \quad (s,d) \in K, k \in V : k = s$

(3) $\sum_{l:(k,l)\in A} X_{kl}^{sd} - \sum_{l:(l,k)\in A} X_{lk}^{sd} = -D_{sd}, \quad (s,d) \in K, k \in V : k = d$

(4) $\sum_{(s,d)\in K} (X_{kl}^{sd} + X_{lk}^{sd}) \leq C_{kl}, \quad \{k,l\} \in E$

(5) X_{kl}^{sd} integer

Objective function of the above model is the multiplication of the total flow on an edge with the cost of that edge for a unit flow. Hence, the flow will tend to use low-cost edges. Constraints (1), (2) and (3) are conservation constraints for intermediate, source and destination nodes, respectively. Constraint (4) is the capacity constraint, which limits the amount of flow that can pass through any arc and finally, (5) is the domain constraint. Our studies show that the capacity limitation yields worse solutions in the second stage, therefore, we set the capacity to a big number, which means that we basically solve shortest path problem between each source-destination pair.

5.2.2 Processing Data Acquired From MCNF Problem

Once (ILP-4) is solved, the first stage is over, however before going on with stage two, some transformations have to be performed on the results of (ILP-4), because the results are flows, whereas stage two requires paths. Therefore, the flow values have to be transformed into paths. This can be done by the algorithm presented in [1]. However, we have to make a slight modification in order to take the capacity of a path into account. Because, the path should not carry flow more than the capacity of a wavelength, otherwise, the path could not be assigned to any super lightpath. Hence, we can say that each path has a capacity of C.

(ILP-4) would not yield cycles, because a cycle increases the cost. Hence, according to the flow decomposition theorem [1] the (ILP-4) flow values can be represented as paths. The algorithm for transforming flow values into the paths is presented in Algorithm 4, where the definitions of A_m and $tail(a)$ are same as in the Algorithm 2 and additionally X_a^{sd} is the amount of flow belonging to source-destination pair (s, d) passing through arc $a \in A$, $A^m = \{(l, m) : (l, m) \in A\}$, $head((k, l)) = k$ and C is the capacity of a wavelength.

Algorithm 4 selects each source-destination pair and performs the following procedure for each selected pair: it picks the arc that has the minimum flow value belonging to the selected source-destination pair. After that, it moves from the tail node of the picked arc towards the destination node through the arcs with positive flow values in the same direction as arcs. If there exist more than one positive arcs departing from a node, then the one with the minimum flow value is selected to move through. After reaching to the destination node, it moves from the head node of the picked arc towards the source node through the arcs with positive flow values in the opposite direction of arcs. Then, the set of arcs that the algorithm passes through and the arc that is picked form a path from the source to the destination. Now, it calculates the amount of flow to be deducted from the arcs that form the path by taking the smaller one of C and the flow value of the picked arc. Then, it subtracts the calculated flow value from the flow values of each arc in the path. It again checks the same source destination pair with the updated flow values and construct paths till the arc with minimum flow value has a flow value of 0.

The paths formed by Algorithm 4 are to be assigned to super lightpaths in the second stage. Hence, in order to perform this assignment, some information

Algorithm .4 Transforming Flow Values into Paths

for all Source-destination pair $(s, d) \in K$ **do**
  $minflow \leftarrow C$
  while $minflow > 0$ **do**
    $Path \leftarrow \emptyset$
    $flow \leftarrow min(X_b^{sd} : b \in A)$
    if $flow > 0$ **then**
      $a \leftarrow argmin(X_b^{sd} : b \in A)$
      $Path \leftarrow Path \cup \{a\}$
      $minflow \leftarrow min(X_a^{sd}, C)$
      $X_a^{sd} \leftarrow X_a^{sd} - minflow$
      $b \leftarrow a$
      while $tail(b) \neq d$ **do**
        $b \leftarrow argmin(X_g^{sd} : g \in A_{tail(b)})$
        $Path \leftarrow Path \cup \{b\}$
        $X_b^{sd} \leftarrow X_b^{sd} - minflow$
      end while
      $b \leftarrow a$
      while $head(b) \neq s$ **do**
        $b \leftarrow argmin(X_g^{sd} : g \in A^{head(b)})$
        $Path \leftarrow Path \cup \{b\}$
        $X_b^{sd} \leftarrow X_b^{sd} - minflow$
      end while
    end if
  end while
end for

about the paths must be defined as parameters to the assignment problem, such as the amount of data each path carries, source node and route of each path. Moreover, it has to be known whether two paths can be assigned to the same super lightpath or not. Therefore, the following parameters has to be defined:

$$T_{ikl} = \begin{cases} 1 & \text{if path i passes through arc } (k, l) \\ 0 & \text{otherwise} \end{cases}$$

d_i: amount of data that path i carries

S_i: source node of path i

$$K_{im} = \begin{cases} 1 & \text{if paths i and m can be assigned to the same super lightpath} \\ 0 & \text{otherwise} \end{cases}$$

Calculating first three parameters are straightforward, however K_{im} needs more computation. K_{im} depends on three different properties of paths i and m. First one is that their sources must be same. This follows from the definition of the super lightpaths, that is, one can add data to a super lightpath only at the source node. The second one is that one of the paths must be a subset of the other one, otherwise resulting super lightpath will not have a path structure. The third one is related to the amount of data that they carry. If the sum of the amounts of data carried by two paths is greater than the wavelength capacity (C), then they can not be assigned to the same super lightpath. Hence, the summation must be smaller than or equal to C. So,

$$K_{im} = 0 \text{ if } (S_i \neq S_m) \text{ or } (I \nsubseteq M \text{ and } M \nsubseteq I) \text{ or } (D_i + D_m \geq C)$$

where I and M represents the sets of arcs that paths i and m pass through, respectively.

5.2.3 Super Lightpath Assignment Problem

The assignments of paths to the super lightpaths is performed by solving an integer linear program. In order to construct the ILP, a decision variable regarding the assignment of a path to a super lightpath is necessary. Moreover, a variable regarding the route of the super lightpaths and another variable for the number of fibers used on any edge are required. Hence, the following variables should be defined:

$$P_{jkl} = \begin{cases} 1 & \text{if super lightpath j uses arc } (k,l) \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ij} = \begin{cases} 1 & \text{if path i is assigned to super lightpath j} \\ 0 & \text{otherwise} \end{cases}$$

F_{kl} : number of fibers on edge $\{k,l\} \in E$

After defining the decision variables, the information gathered from intermediate stage should be introduced to the model. The information to be introduced includes the routes of each path, amount of data that each path carries and whether any two paths can be assigned to the same super lightpath or not. Furthermore, we have to introduce the capacity of a wavelength and the cost of a fiber on any edge. Hence, following parameters are defined:

$$T_{ikl} = \begin{cases} 1 & \text{if path i passes through arc } (k,l) \in A \\ 0 & \text{otherwise} \end{cases}$$

$$K_{im} = \begin{cases} 1 & \text{if paths i and m can be assigned to the same super lightpath} \\ 0 & \text{otherwise} \end{cases}$$

d_i : amount of data that path i carries

L_{kl} : cost of a fiber on edge $\{k,l\} \in E$

W : number of wavelengths that a fiber has

The objective function of this integer linear program should match with the objective function of (ILP-1), that is, it has to be the sum of total fiber cost and total wavelength usage cost. Additionally, the necessary relations must be constructed as constraints. For example, the relation that uses K_{im}, that is, if $K_{im} = 0$, then paths i and m can not be assigned to the same wavelength. Furthermore, the relationships between the routes of paths, routes of super lightpaths and number of fibers must be constructed. Hence, the integer linear program is below, where p is the number of paths. Since we do not know how many super lightpaths we need, we assume that there are p super lightpaths available, which is the worst case.

(ILP-5)

Min $\displaystyle\sum_{\{k,l\}\in E} F_{kl} \times L_{kl} + \alpha \times \sum_{j} \sum_{(k,l)\in A} P_{jkl}$

s.t.

(1) $Y_{ij} + Y_{mj} \leq 1, \quad i,j,m \in \{1,..,p\} : K_{im} = 0$

(2) $P_{jkl} \geq Y_{ij} \times T_{ikl}, \quad i,j \in \{1,..,p\}, (k,l) \in A : T_{ikl} = 1$

(3) $\displaystyle\sum_{j=1}^{p} P_{jkl} + \sum_{j=1}^{p} P_{jlk} \leq W \times F_{kl}, \quad \{k,l\} \in E$

(4) $\displaystyle\sum_{i=1}^{p} Y_{ij} \times d_i \leq C, \quad j \in \{1,..,p\}$

(5) $\displaystyle\sum_{j=1}^{p} Y_{ij} = 1, \quad i \in \{1,..,p\}$

(6) F_{kl} integer

(7) P_{jkl}, Y_{ij} binary

Objective function of the above model is the summation of the fiber cost and the wavelength usage cost. Fiber cost is calculated as the summation of the costs of each edge, where cost of an edge is the multiplication of the number of fibers used at that edge and the cost of a single fiber at that edge, whereas, the wavelength usage cost is calculated as α times the number of wavelengths used to route all the super lightpaths. Hence, this function exactly represents the original objective function. First constraint ensures that any two lightpaths that does not satisfy the grooming limitations, that is $K_{im} = 0$, can not be groomed onto the same super lightpath. Constraint (2) determines the route of the super lightpath, which may be a combination of more than one path. That is, if two paths are assigned to a super lightpath, then the route of the super lightpath must be the route of the longest path. (3) is the fiber constraint, which determines the number of fibers to be used on any edge. Constraint (4) limits the number of lightpaths to be assigned to a single super lightpath so that the summation of amount of data on each lightpath would not exceed the wavelength capacity. (5) is the assignment constraint, which ensures that each path is assigned to a super lightpath. Finally, (6) and (7) are the domain constraints.

As the number of paths and number of nodes increase, (ILP-5) becomes intractable. Hence, a heuristic has to be developed in order to solve the assignment problem for big networks. The heuristic we propose is described formally in Algorithm 5. It picks the first path, it opens a super lightpath to assign the path to it. Hence, it updates the route, remaining capacity and source of the super lightpath accordingly. Then, it picks the second path and it checks if the path can be assigned to the first super lightpath by comparing sources, routes and demand-capacity status. After the comparison, if the path can be assigned to it, then it assigns and updates route and remaining capacity of the super lightpath. If the path can not be assigned to the first super lightpath according to the comparison result, then a new super lightpath is opened and the path is assigned to this super lightpath. Hence, it updates the properties of the new super lightpath

accordingly. This procedures goes on until all the paths are assigned. More formal explanation of the procedure is given in Algorithm 5, where,
sourceS[j] indicates the source node of super lightpath j,
sourceP[i] indicates the source node of path i,
capacity[j] indicates the remaining capacity of super lightpath j,
data[i] indicates the amount of data that the path i carries,
routeS[j] indicates the set of arcs that super lightpath j passes through,
routeP[i] indicates the set of arcs that path i passes through,
C indicates the capacity of a wavelength:

Algorithm .5 Assigning Paths to Super Lightpaths

```
Initialize capacity[j] to be C for all j
for all Path i do
  j ← 0
  while Path i is not assigned to super lightpath j do
    if j is not occupied before then
      sourceS[j] ← sourceP[i]
      capacity[j] ← capacity[j] - data[i]
      routeS[j] ← routeP[i]
    else
      if sourceP[i] = sourceS[j] and data[i] ≤ capacity[j] then
        if (routeP[i] ⊆ routeS[j] or routeS(j)⊆ routeP[i] then
          capacity[j] ← capacity[j] - data[i]
          routeS[j] ← argmax(|routeS[j]|, |routeP[i]|)
        end if
      else
        j ← j + 1
      end if
    end if
  end while
end for
```

5.3 Application of Tabu Search

Tabu Search is a well known type of improvement algorithm. This type of algorithms do not construct a solution from scratch, rather they take an initial feasible solution to reach better solutions by moving in the feasible region. The movement is a step by step procedure, where at each step one of the neighbor solutions is selected as the new solution. The solutions around the initial solution that can be reached via single step of the movement function are called *neighbor solutions*. The best of these neighbors is selected as the new solution even though it may be worse than the initial solution. After that, the neighbors of the new solution are scanned and again the best one is selected. This cycle is repeated for a certain number of times.

In order to prevent cycles for this movement procedure, moving backwards is forbidden, which means that after visiting a solution, it can not move back to this solution for several number of steps. This is prevented by maintaining a *tabu list*, which is the set of solutions that can not be visited at a specific step. The list is empty at the beginning of the moving procedure, then after each move the visited solution is added to the list. If the list is full and a new solution is visited, then the first solution of the list is erased and the new solution is added. Hence, the list is updated in a First In First Out manner. In some cases, a solution can be selected as the new solution even though it is in the tabu list, which is covered by *aspiration criteria*. Aspiration criteria determines the condition under which a solution in the tabu list can be selected as the new solution. One obvious case is that if the result of the solution in tabu list is better than the best solution so far.

The efficiency of the movement procedure depends on the movement function, the size of the tabu list and the aspiration criteria. Determination of these parameters solely depends on the structure of the problem under consideration.

For the problem studied in this thesis, an algorithm based on Tabu Search is developed, where the movement is facilitated by rerouting a super lightpath, that is, all the other super lightpaths are preserved in terms of routing and only one super lightpath is rerouted. Hence, two solutions are said to be neighbors if they differ by only one super lightpath routing. So a solution has as many neighbors as the number of super lightpaths available. The algorithm starts with an initial feasible solution. We use the results of the algorithms discussed in previous sections of this chapter as different initial solutions for this algorithm. Then, the algorithm has to decide which neighbor to move to at each step. In order to make that decision, it checks all neighbors and selects the one with the best objective value, which is minimum network cost for the problem under consideration. To do this, it reroutes all super lightpaths separately while preserving the super lightpaths other than the one that is being rerouted. More specifically, it reroutes first super lightpath and calculates the corresponding cost. Then, it resets the change on first super lightpath and reroutes second super lightpath and all others in this manner. After the cost values are gathered for each super lightpath rerouting, the one with the minimum cost is selected and the new solution is set to be the one with the new route of the corresponding super lightpath. Note that, the solution selected may not be better than the previous solution in terms of the cost, but this does not prevent moving. The one of the initial solution and the new solution with the better result is set to be the best solution visited so far and this solution is updated in further moves if the moved solution is better than this one. After moving to the new solution, the rerouted super lightpath is added to the tabu list. This completes a moving step. This moving step is repeated for a certain number of times with a little modification, which is caused by the tabu list. Because in the subsequent steps, while checking all neighbors, the ones gathered with rerouting the super lightpaths in the tabu list can not be selected as the new solution unless the resulting cost is smaller than the best solution so far. More formal explanation of the algorithm is presented in Algorithm 6, where TL is the set of super lightpaths that are in the tabu list, $S_{initial}$ is the initial solution, $C_{initial}$ is the resulting cost of initial solution and $\sharp repetitions$ is the number of movements steps to perform.

Rerouting is performed by an integer linear program, which is a reduced version of (ILP-1). The reduced ILP, like the original one, tries to minimize the total fiber and wavelength usage cost, but it considers only a single super lightpath

Algorithm .6 Application of Tabu Search

$C_{new} \leftarrow C_{initial}$
$S_{new} \leftarrow S_{initial}$
$TL \leftarrow \emptyset$
$counter \leftarrow 0$
while $counter \leq \sharp repetitions$ **do**
    $C_{max} \leftarrow \infty$
    for all Super lightpath slp **do**
        Reroute slp in the solution S_{new} to get solution S_{slp}
        Calculate the corresponding cost C_{slp}
        if $slp \notin TL$ **and** $C_{slp} < C_{max}$ **then**
            $C_{max} \leftarrow C_{slp}$
            $S_{max} \leftarrow S_{slp}$
            $slp^* \leftarrow slp$
        end if
        if $slp \in TL$ **and** $C_{slp} < C_{best}$ **then**
            $C_{max} \leftarrow C_{slp}$
            $S_{max} \leftarrow S_{slp}$
            $slp^* \leftarrow slp$
        end if
    end for
    $C_{new} \leftarrow C_{max}$
    $S_{new} \leftarrow S_{max}$
    if $C_{new} < C_{best}$ **then**
        $C_{best} \leftarrow C_{new}$
        $S_{best} \leftarrow S_{new}$
    end if
    if TL is full **then**
        Remove the entry that is inserted first
    end if
    $TL \leftarrow TL \cup \{slp^*\}$
    $counter \leftarrow counter + 1$
end while

routing, where the nodes that the super lightpath has to carry data is predetermined. That is, when a super lightpath is selected from a solution, the nodes, to which it carries data, are known. Since the content of the super lightpath is not changed during rerouting, the super lightpath will again carry data to the same nodes but through a different route. The ILP is updated at each step by the help of parameters, which are calculated as follows: after the super lightpath to reroute is selected, it is removed from the system, that is, the wavelengths that are used for it are freed. Then, available wavelengths that can be used without opening a new fiber on each edge $\{k, l\}$, A_{kl}, can be calculated. After that, we have to force the model to visit the nodes, to which it carries data. Hence, we define V_k for all $k \in V$, where $V_k = 1$ if the super lightpath carries data to node k in the original solution. V_k value for the source node of the super lightpath is a little different than the other nodes. V_s is equal to the number of nodes to visit, that is: $V_s = \sum_{k \in V : k \neq s} V_k$. Finally, we need another parameter to force the model not to use the same route as the original one, U_{kl}, which indicates whether the arc (k, l) is used by the selected super lightpath or not in the original solution. Hence the following parameters have to be defined:

A_{kl} : number of available wavelengths without opening a new fiber after removing the super lightpath to reroute from the system.

L_{kl} : cost of the edge $\{k, l\} \in E$

$$U_{kl} = \begin{cases} 1 & \text{if arc (k,l) is used by the super lightpath in the original solution} \\ 0 & \text{otherwise} \end{cases}$$

$$V_k = \begin{cases} 1 & \text{if node k must be visited by the super lightpath in the latter model} \\ \sum\limits_{l \in V : l \neq s} V_l & \text{if } k \text{ is the source node of the super lightpath to reroute} \\ 0 & \text{otherwise} \end{cases}$$

C : capacity of a wavelength.

W : number of wavelengths in a fiber.

Besides these parameters, decision variables regarding number of fibers to open on any arc, route and amount of flow has to be defined:

F_{kl} : number of fibers to open additionally on the edge $\{k, l\} \in E$

$$Y_{kl} = \begin{cases} 1 & \text{if the arc (k,l) is used by the new route} \\ 0 & \text{otherwise} \end{cases}$$

P_{kl} : amount of flow on the arc (k, l)

The model developed is presented below, where s represents the source node of the super lightpath to reroute.

(ILP-6)

$$Min \sum_{\{k,l\} \in E} F_{kl} \times L_{kl} + \alpha \times \sum_{(k,l) \in A} Y_{kl}$$

s.t.

(1) $Y_{kl} + Y_{lk} - W \times F_{kl} \leq A_{kl}, \quad \{k,l\} \in E$

(2) $\sum_{l:(k,l) \in A} Y_{kl} - \sum_{l:(l,k) \in A} Y_{lk} \leq 0, \quad k \in V : k \neq s$

(3) $\sum_{l:(s,l) \in A} Y_{sl} - \sum_{l:(l,s) \in A} Y_{ls} = 1$

(4) $\sum_{l:(k,l) \in A} P_{kl} - \sum_{l:(l,k) \in A} P_{lk} = V_k, \quad k \in V$

(5) $P_{kl} - C \times Y_{kl} \leq 0, \quad (k,l) \in A$

(6) $P_{kl} - Y_{kl} \geq 0, \quad (k,l) \in A$

(7) $\sum_{(k,l) \in A} Y_{kl} \times U_{kl} - \sum_{(k,l) \in A} U_{kl} \leq -1$

(8) F_{kl} and A_{kl} integer, P_{kl} binary

The objective function of the above model complies with the original one, because both incur fiber cost and the wavelength usage cost. The original model incorporates the whole network cost, on the other hand the model above considers only the cost incurred from the construction of a single super lightpath. Constraint (1) is the fiber constraint. If the number of wavelengths available (A_{kl}) on an edge is sufficient to reroute, then there will be no need to open a new fiber. 2nd and 3rd constraints are the conservation constraints in terms of binary arc usage variable (Y_{kl}). Constraint (4) is the flow constraint, which enforces the super lightpath to visit the mandatory nodes. Constraints (5) and (6) are used to construct the relation between the binary arc usage variable and the integer flow variable. 7th constraint ensures that the super lightpath does not use the same route as the original one. Finally, constraint (8) is the domain constraint.

5.4 Experimental Results

The network topologies, traffic patterns and the parameter settings (W=4, C=10, $\alpha = 1$) for the experimental studies performed in this chapter are the same as in Chapter 4.

In table 5.1 the results of LP relaxation based algorithms and the two-stage algorithm are presented. First column represents the topology-traffic cases. Column 3 represents the LP relaxation based algorithm described in Algorithm 2. In the next column, results of the same algorithm with the further improvement (threshold of 0,5) is presented. Columns 5 and 6 are for the results of Algorithm 3 with and without the further improvement of 0,5 threshold. In column 7, the results for the two-stage algorithm are available. The empty cells mean that a solution was not attainable within 24 hours for the corresponding topology-traffic cases. Column 8 represents the results of two-stage algorithm too, however this time the second stage is performed by the heuristic rather than the ILP. And, the last column is the best lower bound gathered from Chapter 4. The percentages given in the table are the gap between the corresponding result and the best lower bound. The gap is calculated as $100 \times \frac{(result)-(bestLB)}{(bestLB)}$.

Table 5.3 presents a summary of Table 5.1. We can see that, using threshold of 0,5 improves the performance of the LP relaxation based algorithms without increasing the CPU time requirements. It decreases the gap from 62% to 41% for Algorithm 2 and from 30% to 29% for Algorithm 3. This shows that arc usage variable selection procedure used in Algorithm 2 selects the variables with values smaller than 0,5 to set to 1 more frequently. However, Algorithm 3 selects at most $|V|$ variables at a pass, that is, it lets the flow to accumulate on selected arcs and this leads to bigger arc usage values. Hence, the effect of using 0,5 threshold is limited on Algorithm 3. As expected, Algorithm 3 provides better solutions compared to Algorithm 2, whereas it requires more CPU time.

The two-stage algorithm with the assignment performed by the (ILP-5) can not give solutions for every topology-traffic case within reasonable time. Hence, the heuristic, described in Algorithm 5, is used in order to perform the second stage. The results of the problem with the heuristic are the same as the problem with the ILP for the cases that the problem with the ILP is able to generate results. Furthermore, the average CPU time for the problem with the heuristic for all the cases is smaller than 1 second, whereas it is 10429 seconds for the problem with the ILP.

In Table 5.2, performance of Tabu Search algorithm is presented for three different initial solutions. 300 repetitions of tabu movements are performed for each initial solution for a tabu list of the size of 7. First initial solution used is the result of Algorithm 2 with 0,5 threshold, whose results are given in 3rd column. The result of Tabu Search and corresponding percent improvement are presented as well as the CPU time required for Tabu Search in 4th column. The percentage is calculated as $100 \times \frac{(tabu)-(initial)}{(initial)}$. The second initial solution used is the result of Algorithm 3 with 0,5 threshold. The results of the initial solution and the corresponding Tabu Search result is available in columns 5 and 6, respectively. The last initial solution used is the result of the 2-Stage algorithm with heuristic assignment. Columns 7 and 8 are for presenting initial solution results and corresponding Tabu Search results.

Table 5.4 presents a summary of Table 5.2 in the sense that it shows the average percent improvement and average CPU times of Tabu Search applied on three different initial solutions. We can see that Tabu Search has limited effect on the solutions of 2-Stage algorithm, which means that packing of traffics on super lightpaths are not performed within this algorithm as efficiently as the Algorithm 2 and 3. Tabu Search is based on rerouting of super lightpaths, i.e, it can not change the content of them. Therefore, the initial packing of super lightpaths is the limiting factor of the performance of Tabu Search.

Finally, in Table 5.5 the performance of our overall solution approach is presented. 2nd column shows the best lower bound found in Chapter 4 for the corresponding topology-traffic case. In the 3rd column the results of (ILP-1) is available. As it is seen, the optimal solutions can be gathered for only a small portion of the cases (5 out 22). Moreover, the necessary CPU times are excessive, which are presented in Table 5.6. The next column shows the best solution found in Chapter 5 for the corresponding topology-traffic case. And the last column represents the gap between the best solution and the best lower bound, where there does not exist optimal result. Otherwise, the gap is defined to be between the best result and the optimal result. Therefore, it is calculated as $100 \times \frac{(solution)-(bestLB)}{(bestLB)}$, where there is no optimal result and $100 \times \frac{(solution)-(optimal)}{(optimal)}$, otherwise. We can see that, the size of the gap varies from 3% to 30%, where the average size of the gap is 13%.

		Algorithm 2		**Algorithm 3**				
		No Half	**Half**	**No Half**	**Half**	**2-Stage 1**	**2-Stage 2**	**Best LB**
	Result	260	228	210	210	210	210	156
5-LD	Percent	67%	46%	35%	35%	35%	35%	
	Time(sec)	0,02	0,02	0,03	0,03	0,25	0,01	
	Result	266	296	290	290	244	244	206
5-HD	Percent	29%	44%	41%	41%	18%	18%	
	Time(sec)	0,04	0,04	0,08	0,09	0,12	0,01	
	Result	590	478	388	388	416	416	286,5
6-LD	Percent	106%	67%	35%	35%	45%	45%	
	Time(sec)	0,05	0,05	0,07	0,07	1,59	0,02	
	Result	694	598	552	552	524	524	406,8
6-MD	Percent	71%	47%	36%	36%	29%	29%	
	Time(sec)	0,11	0,13	0,2	0,22	93784	0,02	
	Result	728	722	690	690		730	591
6-HD	Percent	23%	22%	17%	17%		24%	
	Time(sec)	0,19	0,21	0,35	0,43		0,02	
	Result	568	556	486	484	454	454	404,82
6-XD	Percent	40%	37%	20%	20%	12%	12%	
	Time(sec)	0,11	0,11	0,2	0,2	1,05	0,02	
	Result	704	626	560	602	606	606	412,5
7-LD	Percent	71%	52%	36%	46%	47%	47%	
	Time(sec)	0,14	0,14	0,25	0,28	13,04	0,04	
	Result	870	804	744	744		788	567
7-MD	Percent	53%	42%	31%	31%		39%	
	Time(sec)	0,21	0,25	0,42	0,48		0,05	
	Result	1052	974	976	976		1010	840,4
7-HD	Percent	25%	16%	16%	16%		20%	
	Time(sec)	0,39	0,45	1,05	1,1		0,04	
	Result	958	826	762	694	696	696	556,15
7-XD	Percent	72%	49%	37%	25%	25%	25%	
	Time(sec)	0,27	0,29	0,51	0,59	9,93	0,04	
	Result	1078	912	860	824	758	758	545,4
8-LD	Percent	98%	67%	58%	51%	39%	39%	
	Time(sec)	0,29	0,32	0,41	0,43	17,65	0,08	
	Result	1230	1170	1038	1000		922	667,4
8-MD	Percent	84%	75%	56%	50%		38%	
	Time(sec)	0,61	0,66	1,03	1,18		0,09	
	Result	1396	1206	1202	1226		1232	925,78
8-HD	Percent	51%	30%	30%	32%		33%	
	Time(sec)	1,07	1	1,84	1,84		0,08	
	Result	1098	1074	980	980	906	906	664,31
8-XD	Percent	65%	62%	48%	48%	36%	36%	
	Time(sec)	0,65	0,7	1,18	1,44	32,61	0,09	
	Result	15812	10824	9672	9670		12414	7645,36
9-LD	Percent	107%	42%	27%	26%		62%	
	Time(sec)	0,59	0,56	1,41	1,46		0,13	
	Result	17894	14642	13984	14254		15370	12178,02
9-MD	Percent	47%	20%	15%	17%		26%	
	Time(sec)	1,09	1,07	2,96	3		0,13	
	Result	24320	19912	18760	18760		19978	17519,9
9-HD	Percent	39%	14%	7%	7%		14%	
	Time(sec)	1,9	1,86	6,91	6,43		0,14	
	Result	20026	16174	16070	14982		17374	12805,78
9-XD	Percent	56%	26%	25%	17%		36%	
	Time(sec)	1,22	1,4	3,97	4,01		0,13	
	Result	40992	33566	27302	28088		32074	20751,59
14-LD	Percent	98%	62%	32%	35%		55%	
	Time(sec)	6,92	7,39	23	23		0,93	
	Result	56640	43402	40794	40276		44512	32642,42
14-MD	Percent	74%	33%	25%	23%		36%	
	Time(sec)	21,77	19,43	77	78		0,92	
	Result	67582	55326	54512	53816		58600	49020,78
14-HD	Percent	38%	13%	11%	10%		20%	
	Time(sec)	42,18	33,22	158	170		0,95	
	Result	57088	47206	41988	42694		45388	35346,94
14-XD	Percent	62%	34%	19%	21%		28%	
	Time(sec)	24,15	25,74	101	128		0,93	

Table 5.1: Results for Algorithm 2, Algorithm 3 and 2-Stage Algorithm

		Algo. 2	**Tabu**	**Algo. 3**	**Tabu**	**2-Stage**	**Tabu**
5-LD	Result	228	184	210	184	210	208
	Percent		-19%		-12%		-1%
	Time(sec)		27		29		36
5-HD	Result	296	254	290	266	244	226
	Percent		-14%		-8%		-7%
	Time(sec)		55		53		61
6-LD	Result	478	368	388	336	416	392
	Percent		-23%		-13%		-6%
	Time(sec)		50		53		92
6-MD	Result	598	494	552	458	524	460
	Percent		-17%		-17%		-12%
	Time(sec)		98		96		105
6-HD	Result	722	656	690	628	730	664
	Percent		-9%		-9%		-9%
	Time(sec)		142		151		104
6-XD	Result	556	492	484	458	454	454
	Percent		-12%		-5%		0%
	Time(sec)		92		73		69
7-LD	Result	626	500	602	502	606	570
	Percent		-20%		-17%		-6%
	Time(sec)		95		128		108
7-MD	Result	804	668	744	648	788	700
	Percent		-17%		-13%		-11%
	Time(sec)		168		143		158
7-HD	Result	974	904	976	888	1010	926
	Percent		-7%		-9%		-8%
	Time(sec)		229		302		171
7-XD	Result	826	708	694	656	696	662
	Percent		-14%		-5%		-5%
	Time(sec)		195		160		149
8-LD	Result	912	714	824	682	758	704
	Percent		-22%		-17%		-7%
	Time(sec)		303		306		528
8-MD	Result	1170	906	1000	870	922	882
	Percent		-23%		-13%		-4%
	Time(sec)		594		613		355
8-HD	Result	1206	1036	1226	1064	1232	1044
	Percent		-14%		-13%		-15%
	Time(sec)		713		683		518
8-XD	Result	1074	828	980	802	906	854
	Percent		-23%		-18%		-6%
	Time(sec)		483		535		552
9-LD	Result	10824	9724	9670	8644	12414	10988
	Percent		-10%		-11%		-11%
	Time(sec)		231		251		359
9-MD	Result	14642	13632	14254	13284	15370	15100
	Percent		-7%		-7%		-2%
	Time(sec)		353		615		268
9-HD	Result	19912	19338	18760	18572	19978	20102
	Percent		-3%		-1%		1%
	Time(sec)		696		552		316
9-XD	Result	16174	14998	14982	14228	17374	16078
	Percent		-7%		-5%		-7%
	Time(sec)		305		372		584
14-LD	Result	33566	28250	28088	26094	32074	30020
	Percent		-16%		-7%		-6%
	Time(sec)		2374		2125		3156
14-MD	Result	43402	40466	40276	37520	44512	42686
	Percent		-7%		-7%		-4%
	Time(sec)		4406		4770		4094
14-HD	Result	55326	52856	53816	52002	58600	55972
	Percent		-4%		-3%		-4%
	Time(sec)		7181		6407		3779
14-XD	Result	47206	43030	42694	39440	45388	43426
	Percent		-9%		-8%		-4%
	Time(sec)		4061		5866		3349

Table 5.2: Performance of Tabu Search for three different initial solutions

	Algorithm 2		**Algorithm 3**			
	No Half	**Half**	**No Half**	**Half**	**2-Stage 1**	**2-Stage 2**
Percent	62%	41%	30%	29%	32%	33%
Time(sec)	5	4	17	19	10429	0,2

Table 5.3: Average gap and CPU times for Table 5.1

	Tabu (Algo.2)	**Tabu (Algo.3)**	**Tabu (2-Stage)**
Percent	-14%	-10%	-6%
Time(sec)	1039	1104	859

Table 5.4: Average percent improvements and CPU times for Tabu Search

	Best LB	**Optimal**	**Best Solution**	**Gap**
5-LD	156	156	184	18%
5-HD	206	220	226	3%
6-LD	286,5	308	336	9%
6-MD	406,8		458	13%
6-HD	591		628	6%
6-XD	404,82	426	454	7%
7-LD	412,5	426	500	17%
7-MD	567		648	14%
7-HD	840,4		888	6%
7-XD	556,15		656	18%
8-LD	545,4		682	25%
8-MD	667,4		870	30%
8-HD	925,78		1036	12%
8-XD	664,31		802	21%
9-LD	7645,36		8644	13%
9-MD	12178,02		13284	9%
9-HD	17519,9		18572	6%
9-XD	12805,78		14228	11%
14-LD	20751,59		26094	26%
14-MD	32642,42		37520	15%
14-HD	49020,78		52002	6%
14-XD	35346,94		39440	12%
Average Gap				**13%**

Table 5.5: Performance of overall solution approach

	5-LD	**5-HD**	**6-LD**	**6-XD**	**7-LD**
Result	156	220	308	426	426
Time (sec)	5	2721	406	74185	7115

Table 5.6: Optimal Results

Chapter 6

Conclusion

In this thesis, we studied the routing and wavelength assignment (RWA) problem associated with super lightpaths under static traffic. The problem is routing all traffic requirements so as to minimize the network cost, which is composed of fiber and wavelength usage costs. It includes deciding which outgoing traffic requirements of each node to be satisfied by using the same super lightpath and routing the super lightpath accordingly, i.e, a super lightpath has to visit all the nodes that it carries data to.

First, we developed an integer linear program (ILP) to solve the problem exactly. However, we proved that the problem is NP-Hard. Hence, we could not get optimal results for most of the topology-traffic cases. Therefore, we developed three heuristic approaches to get feasible solutions within reasonable time periods.

The first heuristic is an LP relaxation based algorithm, that is, it starts with the LP relaxation of the model and generates a feasible solution. LP relaxation solution for a super lightpath is a rooted tree, where the root node of the tree is the source node of the corresponding super lightpath. The algorithm tries to construct a path out of the tree by traveling through the tree. The travel starts from the source node and continues through one of the leaves of the tree, where at each branch, the arc with the maximum arc usage value is selected to move through. Consequently, the set of arcs that the algorithm visits forms a path, i.e, super lightpath. We used two different variations of this algorithm. The first one constructs all the super lightpaths at a single pass of the algorithm, whereas the second one constructs only one arc for only one of the super lightpaths that originates from the same node.

The second heuristic is a two-stage algorithm. In the first stage, multicommodity network flow problem is solved with the same network topology and traffic matrix as the original problem. Then, the resulting flow values are transformed into paths with an algorithm. In the second stage, the formed paths are assigned to the super lightpaths. The logic behind this algorithm is that a super lightpath can be thought as a combination of several paths where their sources are the same, their routes are subsets of the one with the longest route and the summation of amount of traffic that they carry is smaller than or equal to the wavelength capacity.

The third heuristic is the well known Tabu Search. This algorithm starts with an initial feasible solution and tries to reach better solutions by moving in the feasible region. We use the results of other heuristics as different initial solutions to this algorithm. The movement in the feasible region is facilitated by rerouting a single super lightpath, while conserving all other super lightpaths in terms of routing. The rerouting of a super lightpath is performed by an ILP, which is a reduced version of the original model. The reduction is due to the fact that the new ILP will deal with the resulting cost of routing a single super lightpath.

In order to evaluate the quality of the solutions gathered by the heuristics, we compared them with the lower bounds of the problem. One obvious lower bound is the LP relaxation of the solution, however generally it yields really weak lower bounds. Hence, we improved this lower bound by using a specific valid inequality. Specific valid inequality lifts the arc usage variable values by observing the traffic requirements between each source destination pair. In order to further improve this bound we used two most violated ST-Cut algorithms. First one is a step-by-step algorithm, where in each step the LP relaxation of the model is solved and S-T set configuration that yields the most violation in terms of arc usage values is determined by an ILP, then the corresponding cut is added to the model. The algorithm repeats this step till the ILP can not find any violated S-T set configuration. The second one is the same as the first one, except that the violation is calculated in terms of fiber usage values.

The proposed heuristics and algorithms to improve lower bound are tested on several test problems. In order to construct test problems six different networks (number of nodes 5, 6, 7, 8, 9 and 14) and four different traffic patterns (light, medium, heavy and mixed patterns) are generated. We gained some insights about the problem and the algorithms we used by observing the results. Then, the results of heuristics are compared with the gathered lower bounds and the gap between these two values are calculated, where there does not exist optimal result. Otherwise, the gap between the best solution and the optimal solution is calculated. Our experimentations show that the gap is around 13% on the average, which constitutes an upper bound on the deviation of our best solution from the optimal solution.

A further research can increase the performance of the Tabu Search. The setting we made allows Tabu Search to change the route of a super lightpath, however, the content of a super lightpath is not allowed to be changed. Therefore, if the initial packing of traffic to the super lightpaths is not well performed, then the quality of the solutions that Tabu Search can reach is limited. Hence, the movement function of Tabu Search can be developed so that it can also change the content of the super lightpaths. Furthermore, we assumed that each node has the wavelength conversion capability. A further research topic can be studying the problem with no or sparse wavelength conversion capabilities.

Bibliography

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows.* Prentice Hall, 1993.

[2] H. Z. B. Mukherjee, K. Zhu. Design of wdm mesh networks with sparse grooming capability. In *IEEE Global Telecommunications Conference (GLOBECOM'02)*, volume 3, pages 2696–2700, 17-21 Nov 2002.

[3] S. Balasubramanian, A. K. Somani, and A. E. Kamal. Sparsely hubbed light-trail grooming networks. In *14th International Conference on Computer Communications and Networks (ICCCN2005)*, pages 249–254, 17-19 Oct. 2005.

[4] D. P. Bertsekas and A. E. Ozdaglar. Routing and wavelength assignment in optical networks. *IEEE/ACM Transactions on Networking (TON)*, 11(2):259–272, April 2003.

[5] L. Calafato, E. Leonardi, M. Mellia, and F. Neri. Exploiting otdm grooming in dynamic wavelength routed networks. In *ONDM2004*, pages 585–598, 2004.

[6] D. Cavendish, A. Kolarov, and B. Sengupta. Routing and wavelength assignment in wdm mesh networks. In *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, volume 2, pages 1016–1022, 29 Nov. - 3 Dec. 2004.

[7] C. Chen and S. Banerjee. A new model for optimal routing and wavelength assignment in wavelength division multiplexed optical networks. In *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, volume 1, pages 164–171, 24-28 March 1996.

[8] I. Chlamtac, A. Farago, and T. Zhang. Efficient routing of lightpaths. In *MILCOM'96, IEEE Conference on Military Communications*, volume 1, pages 108–111, October 1996.

[9] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high-bandwidth optical wan's. *IEEE Transactions on Communications*, 40(7):1171–1172, July 1992.

[10] I. Chlamtac and A. Gumaste. Light-trails: a novel conceptual framework for conducting optical communications. In *Workshop on High Performance Switching and Routing (HPSR2003)*, pages 251–256, 24-27 June 2003.

[11] I. Chlamtac, A. Gumaste, and G. Kuper. Optimizing light-trail assignment to wdm networks for dynamic ip centric traffic. In *13th IEEE Local area networks and metro area networks, LANMAN Workshop*, pages 113–118, 25-28 April 2004.

[12] J. Choi, N. Golmie, F. Lapeyrere, F. Mouveaux, and D. Su. A functional classification of routing and wavelength assignment schemes in dwdm networks: static case. In *NRC200*, New Jersey, USA, April 14-15 2000.

[13] A. efficient approach to optimal wavelength routing in WDM optical networks. V. q. phung and d. habibi and h. n. nguyen and k. lo. In *12th IEEE International Conference on Networks (ICON 2004)*, volume 2, pages 600–604, 16-19 Nov 2004.

[14] J. Fang, W. He, and A. K. Somani. Optimal light trail design in wdm optical networks. In *IEEE Internationl Conference on Communications (ICC'04)*, pages 1699–1703, 2004.

[15] J. Fang and A. Somani. Enabling subwavelength level traffic grooming in survivable wdm optical network design. In *IEEE Global Telecommunications Conference (GLOBECOM '03)*, volume 5, pages 2761–2766, 2003.

[16] M. Frederick, A. Somani, and N. VanderHorn. Light trails: a sub-wavelength solution for optical networking. In *Workshop on High Performance Switching and Routing (HPSR2004)*, pages 175–179, 2004.

[17] M. Garey and D. S. Johnson. *Computers and Intractibility: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[18] D. Ho-Quang and M. Lee. A heuristic solution for high-performance lightpath set-up procedure in wavelength-routing optical networks. In *Proceedings of ICCT2003*, pages 306–311, 9-11 April 2003.

[19] J. Q. Hu and B. Leida. Traffic grooming, routing, and wavelength assignment in optical wdm mesh networks. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*, volume 1, pages 495–501, 7-11 March 2004.

[20] B. Jaumard, C. Meyer, and B. Thiongane. Ilp formulations and optimal solutions for the rwa problem. In *IEEE Global Telecommunications Conference (GLOBECOM '04)*, volume 3, pages 1918–1924, 29 Nov. - 2 Dec. 2004.

[21] R. M. Krishnaswamy and K. N. Sivarajan. Algorithms for routing and wavelength assignment based on solutions of lp relaxaion. *IEEE Communications Letters*, 5(10):435–437, October 2001.

[22] T. Lee, K. Lee, and S. Park. Optimal routing and wavelength assignment in wdm ring networks. *IEEE Journal on Selected Areas in Communications*, 18(10):2146–2154, October 2000.

[23] D. Li, Z. Sun, X. Jia, and S. Makki. Traffic grooming for minimizing wavelength usage in wdm networks. In *Eleventh International Conference on Computer Communications and Networks*, pages 460–465, 14-16 Oct 2002.

[24] T. L. Magnanti and P. Mirchandani. Shortest paths, single origin-destination network design and associated polyhedra. *Networks 23*, pages 103–121, 1993.

[25] M. Mellia, E. Leonardi, M. Feletig, R. Gaudino, and F. Neri. Exploiting otdm in wavelength routed networks. In *Optical Fiber Communication Conference and Exhibit (OFC2002)*, pages 605–606, 17-22 March 2002.

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

[27] R. Ramaswami and K. Sivarajan. *Optical Networks: A Practical Perspective*. Morgan Kaufmann Publishers, 2002.

[28] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, 1986.

[29] L. A. Wolsey. *Integer Programming*. John Wiley and Sons, 1998.