# Survivability and Impairment-aware Routing in Optical Networks

## An Algorithmic Study

# Survivability and Impairment-aware Routing in Optical Networks

## An Algorithmic Study

**Proefschrift**

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op vrijdag 29 april 2011 om 12:30 uur

door

Anteneh Ayalew BESHIR

ingenieur computer engineering
geboren te Shoa, Ethiopië.

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. P.F.A. Van Mieghem

Copromotor Dr.ir. F.A. Kuipers

Samenstelling promotiecommissie:

| | |
|---|---|
| Rector Magnificus, | Voorzitter |
| Prof.dr.ir. P.F.A. Van Mieghem, | Technische Universiteit Delft, promotor |
| Dr.ir. F.A. Kuipers, | Technische Universiteit Delft, copromotor |
| Prof.dr.ir. C. Roos, | Technische Universiteit Delft |
| Prof.dr. A. Orda, | Technion Israel Institute of Technology |
| Prof.dr.ir. P. Demeester, | Universiteit Gent |
| Prof.dr.ir. C.T.A.M. de Laat, | Universiteit van Amsterdam |
| Dr.ir. R. Nuijts, | SURFnet |
| Prof.dr.ir. N.H.G. Baken, | Technische Universiteit Delft, reservelid |

ISBN 978-94-6113-034-1
This work was supported by the GigaPort project, which is led by SURFnet and funded by the Dutch Ministry of Economic Affairs (grant number BSIK03020).

Keywords: Optical Networks, Wavelength Division Multiplexing, Survivable Routing, Impairment-aware Routing



Printed in the Netherlands

To my parents and my wife

# Contents

# Summary


Optical networks employing Wavelength Division Multiplexing (WDM) technology allow the multiplexing of several independent wavelength channels into a fiber. Since each wavelength channel operates independently at several Gb/s, WDM optical networks offer a tremendous transport capacity (which is in the order of several Tb/s), which makes them suitable candidates for future networks. A *lightpath* is made up of wavelength channels between the source and destination nodes to transfer a large amount of data. Routing in WDM networks involves assigning both paths and wavelengths, and is called *routing and wavelength assignment (RWA)*. In WDM optical networks, there are two vital RWA issues that have garnered a lot of interest from researchers as well as network operators.

1. Survivability: Lightpaths in WDM networks usually transport a tremendous amount of data. If a lightpath fails due to various natural or man-made disasters, the data loss can be costly. Hence, survivability, which is the ability to reconfigure and resume communication is indispensable.

2. Impairment-aware routing: As an optical signal traverses its path, it encounters noise and signal distortions along its way. These physical impairments cause bit errors, which may make the signal unrecognizable at the receiving end. In order to reverse the effect of physical impairments, the signal needs to be regenerated at intermediate nodes. Unlike traditional RWA, impairment-aware RWA, takes into account the effect of physical impairments. Impairment-aware RWA entails two important issues:

    (a) Impairment-aware path selection: how to find a feasible path from the source to the destination node?

    (b) Regenerators placement: how many regenerators are required and where to place them in the network?

The main focus of this thesis is to study various problems associated with survivability and impairment-aware RWA in WDM networks. Figure 1 shows an overview of the research work in this thesis. The thesis considers RWA both in intra-domain

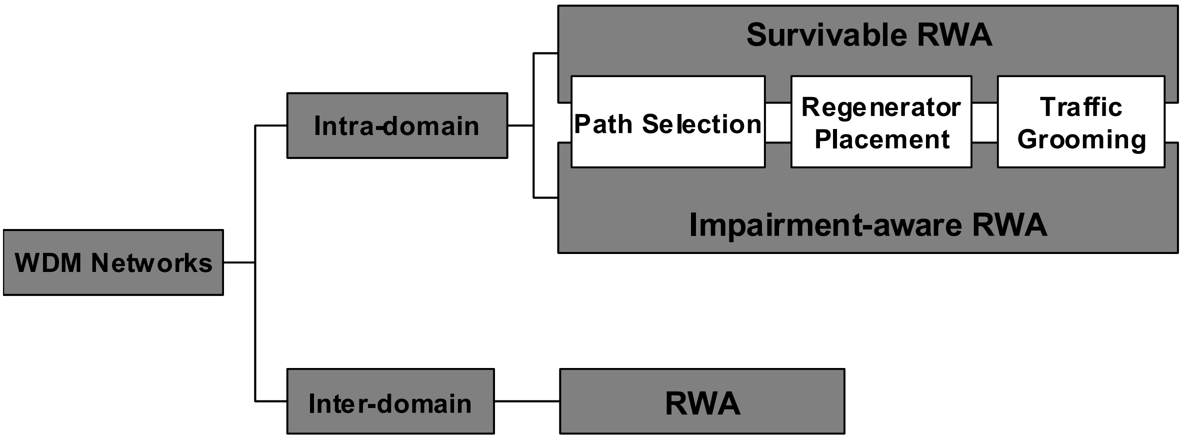



Figure 1: An overview of the research work in this thesis.

(i.e., within in a single domain) and inter-domain (i.e., across domains). Especially, for intra-domain networks, it makes a detailed study of survivable and impairment-aware RWA issues, separately or combined. As shown in Figure 1, the main topics are (1) path selection, i.e., finding survivable and/or impairment-aware lightpaths, (2) regenerator placement for unprotected and protected lightpaths, and (3) survivable and/or impairment-aware traffic grooming. Traffic grooming helps efficient utilization of available network capacity through the aggregation of several independent low-speed traffic streams onto high-speed lightpaths.

For the various problems studied in this thesis, the complexity of the problems is analyzed in detail, and accordingly exact, approximation or heuristic algorithms are proposed for solving them. In addition, a case study of survivable and impairment-aware routing is made on a realistic network that connects research and educational institutes in the Netherlands using data obtained from the network. The work done in this thesis will not only help us gain insight into the various problems in WDM networks, but it may also be applicable to corresponding problems in other types of networks, or even to problems in other areas. For example, impairment-aware routing relates to the gas station problem, where given a set of nodes (towns) with gas stations, the objective is to find a route from one town to another in such a way that a driver is not stranded between gas stations.

# Chapter 1

# Introduction

Optical networks employing wavelength-division multiplexing (WDM) technology are promising solutions to the ever-increasing demand for bandwidth. In wavelength-routed WDM networks, the enormous bandwidth of a fiber is divided into several non-overlapping wavelength channels that can transport data independently. Currently, the fastest wavelength channel supports a data rate of 100 Gb/s [15]. These wavelength channels make up *lightpaths*, which are optical connections that may span several fiber links without using routers.

Depending on the wavelength-conversion capability of their nodes, WDM networks can be classified as *wavelength-selective* or *wavelength-interchanging* [56]. In wavelength-selective networks, the nodes lack wavelength conversion capability. Therefore, a lightpath connection between a source and a destination must use the same wavelength in all links along its route. Whereas in wavelength-interchanging networks, the nodes have the capability to convert a wavelength at an incoming link to a different one at an outgoing link. Even though the absence of wavelength-continuity constraint in wavelength-interchanging networks increases the flexibility of the network, the high price of wavelength converters may make them less desirable.

In WDM optical networks, provisioning lightpaths involves not only routing, but also wavelength assignment, and this process is known as *routing and wavelength assignment (RWA)*. The RWA process in such networks should usually satisfy two important requirements: (1) Survivability, i.e., there should be a mechanism to restore communication after the failure of a lightpath, and (2) impairment-awareness, i.e., the quality of the optical signal, which degrades due to noise and signal distortions along the route of a lightpath, should not drop below a certain threshold. The main goal of this thesis is to study and provide algorithms for various problems pertinent to survivable and impairment-aware routing. Even though our study focuses on WDM optical networks, our algorithms can be extended to other types of (optical) networks as well.

## 1.1 Survivability

Due to the tremendous amount of data transported, *survivability*, which is the ability to reconfigure and reestablish communication upon failure, is indispensable in WDM networks. Hence, survivability of WDM networks has received a lot of attention from both researchers and network carriers [96]. WDM networks are usually employed as multi-layered networks, e.g., IP-over-WDM, SONET-over-WDM, etc. Thus, survivability can be provided either at the optical layer or by higher layers. However, the recovery time at higher layers may be in the order of seconds, while at the optical layer, it is usually only in the order of milliseconds. In addition, survivability at the optical layer is more efficient because of resource sharing and may provide survivability to higher layers that do not have inherent survivability capability [35][100].

When a component fails, all the lightpaths that are currently using this component will also fail. If the network is survivable, another lightpath which does not use the failed component will take over. The lightpath that carries traffic during normal operations is known as the *primary lightpath*, whereas the lightpath that is used to reroute traffic when the primary lightpath fails is called the *backup lightpath*.

### 1.1.1 Survivability Techniques

Depending on whether backup lightpaths are computed before or after a failure of the primary lightpath, survivability techniques can be broadly classified as restoration or protection techniques (see Figure 1.1) [75].

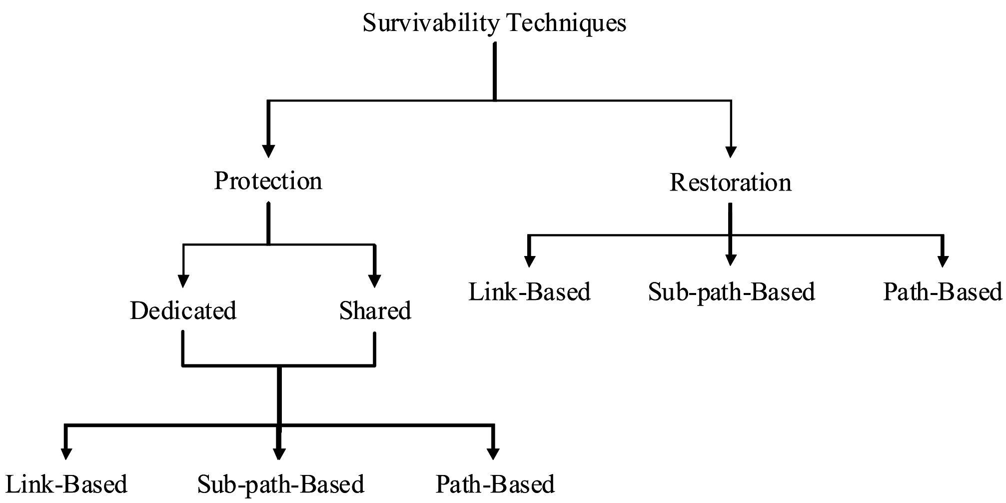



Figure 1.1: Survivability schemes in WDM networks [75].

**Protection scheme**: Protection is a proactive scheme, where backup lightpaths are precomputed and reserved in advance, and traffic is rerouted along the backup lightpath upon the failure of the primary lightpath.

**Restoration scheme**: Restoration is a reactive mechanism that handles a failure after it occurs. Thus, the backup lightpath is not known *a priori*. Instead, a backup lightpath is computed only after the failure in the primary lightpath is sensed.

In general, protection has a shorter recovery time since the backup lightpath is precomputed, but it is less efficient and less flexible. Restoration, on the other hand, provides increased flexibility and efficient resource utilization, but it may take a longer time for recovery and there is no guarantee that a backup lightpath will be found.

Depending on how rerouting is done after a failure in the primary lightpath, there are three categories of survivability techniques: path-based, link-based, and sub-path based.

**Path-based protection/restoration:** In path-based protection, a link- or node-disjoint backup lightpath is precomputed and takes over when the primary lightpath fails. In path-based restoration, a new path and wavelength is computed between the source and destination nodes of the failed lightpath using a (distributed) RWA algorithm. If the algorithm fails to find a backup lightpath, the request is blocked.

**Link-based protection/restoration:** In link-based protection, each link is pre-assigned a local route that is used when it fails, and in link-based restoration, the objective is to compute a detour between the two ends of the failed link for all lightpaths that are using the link. Since link-based protection/restoration requires signaling only between the two ends of the failed link, it has a smaller recovery time than path protection/restoration, which requires end-to-end signaling between the source and destination nodes. However, in link-based protection/restoration, the backup paths may be circuitous, and the backup lightpath is forced to use the same wavelength in wavelength-selective WDM networks since the rest of the primary lightpath is retained.

**Sub-path-based protection/restoration:** The sub-path based scheme is a compromise between path-based and link-based schemes. Thus, in sub-path-based protection, backup routes are precomputed for segments of the primary lightpath. In sub-path-based restoration, a detour of the segment containing the failed link is computed following a failure.

Depending on whether sharing of resources is allowed among backup lightpaths, protection schemes can be of two types: dedicated and shared.

**Dedicated protection:** In this scheme, wavelength channels are not shared among backup lightpaths, and are exclusively reserved for a given lightpath request.

**Shared protection:** In this scheme, backup lightpaths may share wavelength channels on some links as long as their primary lightpaths do not share links. The shared scheme provides a better resource utilization, however it is more complicated and requires more information, such as the shareability of each link.

Ramamurthy *et al.* [85] have shown that in terms of required capacity, path protection significantly outperforms link protection, and shared protection performs significantly better than dedicated protection. However, path protection is more susceptible to multiple link failures than link protection, and so is shared protection compared to dedicated protection. The choice of survivability techniques may depend on the following factors.

### Types of Traffic

Typically, the traffic demands can be static or dynamic [88].

- *Static lightpath establishment*: Under static traffic, requests are known *a priori* and traffic variations occur over long time scales. This is generally associated with design problems when network resources are allocated for a given input of traffic requests.
- *Dynamic lightpath establishment*: In this traffic scenario, the connection requests arrive and depart in a random fashion or follow a certain pattern that may not be known in advance. Thus, the traffic is generally not known *a priori*. Unlike static lightpath establishment, in dynamic lightpath establishment the main objective is to optimize the performance of an already provisioned network.

### Type of Network

The underlying network is another factor that determines the survivability mechanism. The most common topologies considered in the literature are ring and mesh topologies [118]. Rings are the typical choices for metropolitan area networks, while mesh topologies are commonly used in wide area networks. Survivability schemes in ring topologies have been widely studied due to the fact that they are relatively better understood and some of the schemes, such as embedded protection cycles (p-cycles), can be extended to mesh topologies [36][41].

### Wavelength conversion

The presence of wavelength converters generally improves the performance of (survivable) RWA algorithms. Since wavelength converters are costly, sparse wavelength conversion, where only a few nodes have wavelength conversion capability, is usually employed to reduce cost of wavelength conversion. Such an approach offers an improved wavelength utilization with reduced conversion capabilities [86]. However, the algorithms for the design of survivable WDM networks and survivable RWA algorithms in sparse wavelength conversion networks are generally more complicated [40][41]. Van Caenegem *et al.* [21] have suggested that wavelength conversion may not be necessary in survivable WDM networks with smaller number of wavelengths per fiber, since the benefit of wavelength conversion increases with the number of wavelengths per fiber.

#### Centralized vs. distributed

Survivability techniques can be centralized or distributed. Centralized survivability techniques generally require detailed information of all existing lightpaths as well as all node/link capacities, which may not be scalable. In addition, the centralized approach may lead to single point of failure. On the other hand, distributed survivability techniques do not take advantage of shareability information, thereby leading to inefficient use of resources [20].

#### Intra-domain vs. Inter-domain

As far as survivability in WDM optical networks is concerned, a majority of the work done in the literature is for single domain networks. However, WDM optical networks are being employed across domains, and this warrants a study of survivability in multi-domain networks. Unlike single domain networks, in multi-domain networks, there is limitation in the amount and type of information exchanged between domain due to scalability as well as privacy requirements. Therefore, it may not be possible to obtain a complete and global information, which may render most of the survivability techniques for single domain networks unusable in multi-domain networks [101].

### 1.1.2 Objectives

While providing survivability to a network, survivability techniques can be required to satisfy certain objectives. These objectives can be minimization of resource utilization [72], reduced blocking ratio [44], improved shareability of resources (minimizing the spare capacity) [51][79], recovery time [47], etc. The following is a list of the most common objectives [95]:

*Recovery time:* The recovery time (or restoration time) is defined as the down-time that the connection experiences due to a failure. It is important since it determines the amount of data and revenue losses.

*Capacity utilization*: Capacity utilization is defined as the measure of additional backup resources that have been reserved by the specific survivability scheme.

*Blocking ratio*: Blocking ratio can be defined as the ratio of the total number of failed connections to the total number of requests.

*Restoration guarantee*: The extent to which a protection/restoration mechanism can restore a failed connection is termed as the restoration guarantee. Dedicated protection mechanisms provide 100% restoration guarantee.

## 1.2 Impairment-aware Routing

WDM optical networks are widely used in long-haul and metro/regional networks, which usually cover a large distance. In transparent all-optical networks, the signal is transmitted in the optical domain from the source to the destination node, without any conversion to the electrical domain. If the signal is not regenerated at intermediate nodes, noise and signal distortions are accumulated along the physical path. The noise and signal distortions are known as *physical impairments*, and degrade the quality of the received signal. Especially for long distances and high bit rates, the signal degradation may lead to an unacceptable bit-error rate (BER). In such cases, it is necessary to regenerate the signal at intermediate nodes to overcome physical impairments.

Regeneration usually involves re-amplification, re-shaping, and re-timing, which are collectively known as 3R regeneration. Even though optical 3R regenerations have been demonstrated in laboratories, only electrical 3R regenerations are currently the most reliable and economically viable [91]. In other words, signal regeneration is achieved through optical to electrical and then back to optical (O-E-O) conversions, thereby disrupting the transparency of the signal. If signal regeneration is employed at each node, the network is called an *opaque network*; whereas, a network that uses sparse regeneration is known as a *translucent network*. Since regenerators are costly, the latter is preferred for practical implementations. In this thesis, we focus on translucent networks.

Physical impairments can be classified into two categories: linear and non-linear impairments [98]. Linear impairments are independent of signal power and affect wavelengths individually. Non-linear impairments generate dispersion on channels and crosstalk between channels. We shall present the main impairments listed in [98].

- *Polarization Mode Dispersion (PMD)* is a form of modal dispersion where two different polarizations of light in a waveguide travel at different speeds due to imperfections and asymmetries, causing random spreading of optical pulses. PMD is expressed in $ps/\sqrt{km}$, which means that its square value is additive with distance.

- *Amplifier Spontaneous Emission (ASE)* refers to the emission of radiation (photons) due to the presence of an electromagnetic field. ASE degrades the optical signal to noise ratio (OSNR) and is reflected in that measure. In practice, vendors generally provide bounds on the length of the transparent segment and number of spans in order to ensure an acceptable level of OSNR. Assuming the same output power at all amplifiers along a segment, the constraint on the number of spans $H$ is computed as

$$\sum_{j=1}^{H} n_{sp}(j)(\gamma(j)-1) \leq \frac{P_L}{hvB_oSNR_{\min}},$$

where $n_{sp}(j)$ and $\gamma(j)$ are the spontaneous emission factor and the amplifier gain of the $j$-th amplifier, respectively; $P_L$ is the average optical power, $h$ is Planck's constant, $v$ is the carrier frequency, and $B_o$ is the optical bandwidth.

- Other linear impairments, like *Polarization Dependent Loss (PDL)*, *Chromatic Dispersion*, *Crosstalk*, and *Effective Passband*, can be approximated by a domain-wide margin on the OSNR, plus in some cases a bound on the number of networking elements along the path [98].

- Incorporating non-linear impairments is much more complex and requires a detailed knowledge of the physical network. Strand and Chiu [98] suggested to trade-off accuracy for simplicity and to assume that the non-linear impairments are bounded and implicitly reflected in a maximum number of spans.

When considering impairment-aware routing, there are two major areas of research: (1) how to incorporate impairment-awareness in RWA algorithms (impairment-aware path selection), and (2) how many regenerators to place inside the network and where (regenerator placement).

### 1.2.1 Impairment-aware Path Selection

As mentioned earlier, in translucent optical networks, only some nodes are endowed with regeneration capacity. A threshold value is usually provided for each physical impairment. Any given lightpath should be regenerated before any of its impairment values reaches the respective threshold associated with it. Hence, this should be taken into account when assigning a path and a wavelength. This is known as impairment-aware path selection. In short, given a request between two nodes, impairment-aware path selection is an RWA process that assigns a lightpath (path and wavelength) to the request such that the impairment values of the segments of the lightpath between regenerator nodes (i.e., nodes with regeneration capacity) should not exceed their respective thresholds. Since a shortest path may not necessarily be a feasible path, impairment-aware path selection differs from the traditional RWA, where any (shortest) path between the source and destination nodes may suffice.

### 1.2.2 Regenerator Placement

The other issue associated with impairment-aware routing is regenerator placement. In impairment-aware path selection, it is assumed that the regenerators are already placed in the network. On the contrary, the regenerator placement issue is a design process that answers the questions, how many regenerators are needed for a given set of requests and where should these regenerators be placed. Thus, the main objectives in regenerator placement are minimizing the total number of regenerators and the total number of regenerator nodes.

## 1.3 Thesis Scope and Outline

The main focus of this thesis is to study different survivability techniques and impairment-awareness issues in WDM optical networks. We consider various routing and resource allocation problems related to these two important issues. For the various problems, we study their complexity and propose exact, approximation or heuristic algorithms for solving them. We show (and compare) the performances of algorithms analytically and/or through simulations. In the course of this thesis, we begin with networks where regeneration is not required (Chapters 2 and 3), which is suitable for scenarios when all nodes in the network are within a range that does not lead to an unacceptable signal quality. We then proceed to networks where the distance between nodes may warrant the use of regenerators at intermediate points (Chapters 4-7). We also consider inter-domain RWA, where the approaches for intra-domain routing and wavelength assignment may not be suitable (Chapter 8). However, for inter-domain routing, we have only studied unprotected and impairment-agnostic RWA, i.e., survivability and impairment-awareness in inter-domain networks is beyond the scope of this thesis work. We also make a case study (Chapter 9) of survivable impairment-aware RWA on a realistic network using actual data obtained from this network.

The following is an outline of the body of the thesis, which has nine chapters.

**Variants of the min-sum link-disjoint paths problem (P3):** The most commonly used objective in finding link-disjoint paths for survivability is minimizing the total cost of the primary and backup lightpaths (*min-sum*). The min-sum link-disjoint paths problem is polynomially solvable. However, there can be secondary objectives depending on additional requirements. Hence, in this chapter, we consider the effect of several secondary objectives on the complexity of the min-sum link-disjoint paths problem, and provide algorithms for solving these problems.

**On-line survivable routing and wavelength assignment (P4):** In practice, lightpath requests arrive over time and the decision to accept these requests, which may block future requests because of the limited amount of resources (i.e., wavelength channels), should be made without any knowledge of future requests. This is known as on-line routing, which is the opposite of off-line routing, where all requests are known beforehand. As one would expect, the performance of an off-line algorithm is often better than that of an on-line algorithm. The competitive ratio of an on-line algorithm is a measure of its performance against that of an off-line (but often non-implementable) algorithm. In this chapter, we study the on-line survivable routing and wavelength assignment (SRWA) problem. We provide constant and logarithmic competitive ratios for specific networks. For general networks, since it is not possible to find good competitive ratios, we propose wavelength rerouting, which is the process of changing the wavelengths of some of the existing lightpaths to accommodate new requests, so as to improve the performance of online algorithms.

**Impairment-aware path selection (P1):** As was mentioned earlier, lightpaths may need to be regenerated at intermediate nodes in order to restore the quality of the optical signal. In this chapter, we study the impairment-aware path selection problem, where given a lightpath request in a translucent network, the objective is to find a feasible path for the given request. We first prove that the problem is NP-complete. Then, we provide an exact algorithm and derive an efficient heuristic algorithm from it.

**Regenerator placement (P1** and **P5):** In this chapter, we deal with the regenerator placement problem for unprotected lightpath requests. We show that the problem is polynomially solvable if the objective is minimizing the total number of regenerators for a single impairment and there is no limitation on the number of wavelengths, while it becomes NP-hard if there is a secondary objective of minimizing the total number of regenerator nodes.

**Survivable regenerator placement (P5):** In this chapter, we continue with the study of the regenerator placement problem by considering the survivable regenerator placement problem. We consider two survivability techniques: dedicated and shared protection. We show that the problem is NP-hard in both cases. We also provide an approximation algorithm for the former, while giving an efficient heuristic algorithm for the latter.

**Survivable impairment-aware traffic grooming (P6):** Unlike the previous chapters, where we assume that each lightpath requires a full wavelength capacity, in this chapter we consider the case where several requests are aggregated in a single lightpath. This is suitable to scenarios where requests have much less bandwidth requirement than the capacity provided by an optical lightpath (which is several Gb/s). In such scenarios, the main cost is that of adding/dropping traffic and regeneration at nodes. Hence, we consider the problem of impairment-aware survivable traffic grooming, where given a set of requests with demands and a network capacity (i.e., the capacity of the wavelength channels), the problem is to minimize the total cost of adding/dropping traffic and regeneration.

**Inter-domain routing in optical networks (P2):** In this chapter, we focus on RWA algorithms of inter-domain routing protocols. We extend three inter-domain routing algorithms to accommodate the presence of wavelength converters at the border routers, and compare their performance.

**Case study (P7):** In this chapter, we continue with the study of survivable and impairment-aware routing with a case study of the SURFnet6 network, which connects research and educational institutes in the Netherlands. Using realistic data obtained from this network, we compare the performance of our proposed approach to a sequential approach, which is commonly used by practitioners.

**Conclusions:** Finally, we provide general conclusions of the thesis in Chapter 10.

# Chapter 2

# Variants of the Min-Sum Link-Disjoint Paths Problem

## 2.1 Introduction

Survivability is of paramount importance in networks, such as optical networks, that transport a large amount of traffic. In order to prevent single-link failures, which are the most prevalent types of failures, it is necessary to establish connections on link-disjoint primary and backup paths between the source and destination nodes. The primary path is used during normal operations, while the backup path takes over during the failure of the primary path.

There can be several objectives associated with finding link-disjoint paths. The most common and simpler one is the *min-sum* link-disjoint paths problem, which is finding a pair of link-disjoint paths whose combined cost is minimized. Depending on how frequently failures occur on the primary path, it may be desirable to minimize the cost of the primary (shorter) path (min-min problem) [108] or the backup (longer) path (min-max problem) [66]. In constrained routing, the costs or bandwidths of the primary and backup paths need to be bounded [54]. In load balancing, it may be necessary to find a pair of paths with the largest residual bandwidth so that heavily loaded links are avoided (shortest-widest problem) [89].

Among the aforementioned objectives, only the min-sum [99] and the shortest-widest [89] problems are polynomially solvable, while the others are NP-complete. In this chapter, we will investigate whether we can use these other objectives as secondary objectives to the min-sum link-disjoint problem. We show that the NP-complete secondary objectives turn the polynomially solvable min-sum problem to NP-complete min-sum problem variants. However, through simulations we show that due to the strongly reduced search space, exact algorithms can, in practice, solve the respective problem variants in a reasonable time.

The outline of this chapter is as follows. In Section 2.2, a formal definition and the complexity of each problem variant is presented. In Section 2.3, we provide algorithms for these problem variants. In Section 2.4, we present our simulation results, and in Section 2.5, we give conclusions.

## 2.2 Problem Definition

**Problem 2.1** *Given a graph $G(\mathcal{N}, \mathcal{L})$, where $|\mathcal{N}| = N$ and $|\mathcal{L}| = L$, a cost $c(l)$ and a bandwidth $B(l)$ associated with each link $l \in \mathcal{L}$, a source node $s$ and a destination node $d$, two bounds $\Delta_1 \geq 0$ and $\Delta_2 \geq 0$, find a pair of link-disjoint paths from $s$ to $d$ such that*

***Min-Sum Min-Min Link-Disjoint Paths Problem:*** *The total cost of the pair of link-disjoint paths is minimized and if there is a tie, the cost of the shorter path is minimized.*

***Min-Sum Min-Max Link-Disjoint Paths Problem:*** *The total cost of the pair of link-disjoint paths is minimized and if there is a tie, the cost of the longer path is minimized.*

***The Bounded Min-Sum Link-Disjoint Paths Problem:*** *The total cost of the pair of link-disjoint paths is minimized, and then the cost of the shorter path should be less than or equal to $\Delta_1$ and the cost of the longer path should be less than or equal to $\Delta_2$.*

***The Widest Min-Sum Link-Disjoint Paths Problem:*** *The total cost of the link-disjoint paths is minimized, and if there is a tie, the smallest bandwidth of all links in the two paths is maximized.*

The min-sum min-min link-disjoint paths problem is proven to be NP-complete by Yang *et al* [110].

**Theorem 2.1** *The min-sum min-max link-disjoint paths problem is NP-complete.*

To prove this theorem, we make use of the NP-complete partition problem [39], which is defined as follows.

**Problem 2.2** ***The partition problem****: Given a set of values $a_i \in A$, $a_i \geq 0$ for $i = 1, \ldots, n$, where $S = \sum_{i=1}^{n} a_i$. Find a subset $I \subseteq A$ such that $\sum_{a_i \in I} a_i = \sum_{a_i \in A \setminus I} a_i = \frac{S}{2}$.*

**Proof.** We will only provide a proof for undirected graphs. The directed case follows analogously. In Figure 2.1, the labels on the links represent their cost and all links without labels have zero cost. Let $x = 0$. Clearly, the shortest link-disjoint pair of paths from $s$ to $d$ have a total cost of $S$. Thus, for any link-disjoint paths pair $\{P_1, P_2\}$, the best possible value for the min-sum min-max problem in this network is when $c(P_1) = c(P_2) = \frac{S}{2}$. However, finding this pair of link-disjoint paths requires solving the NP-complete partition problem. ■

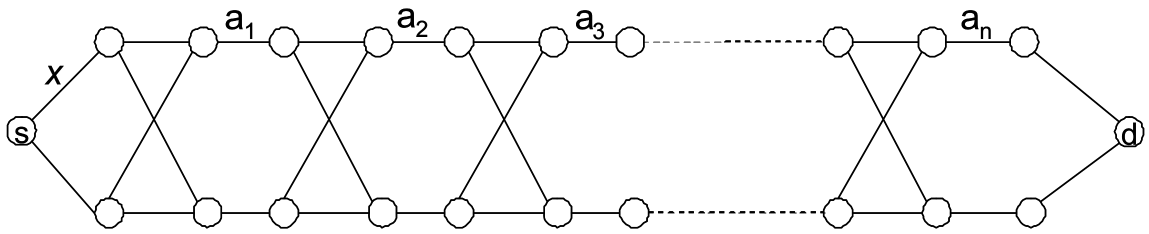



Figure 2.1: All links without labels have zero cost.

**Theorem 2.2** *The bounded min-sum problem link-disjoint paths problem is NP-complete.*

**Proof.** Here also, we use Figure 2.1 and the partition problem. Let $\Delta_1 = \frac{S}{2}$ and $\Delta_2 = x + \frac{S}{2}$. If $x = 0$, $\Delta_1 = \Delta_2$, otherwise $\Delta_1 \neq \Delta_2$. In either case, finding a pair of link-disjoint paths, where the shorter path is bounded by $\Delta_1$ and the longer path is bounded by $\Delta_2$ involves solving the NP-complete partition problem. ∎

The widest min-sum link-disjoint paths problem is not NP-compete and in the next section, we provide an exact polynomial-time algorithm for it.

## 2.3 Algorithms

The MIN-SUM+ algorithm given below is an outline of algorithms that can be used for exactly solving the three NP-complete variants of the min-sum problem. The algorithm basically goes through all the possible candidate pairs of link-disjoint paths, namely those with a total cost equal to that of the min-sum link-disjoint paths pair returned by algorithms such as Bhandari's algorithm [16]. As in Bhandari's algorithm, in Steps $1-3$, MIN-SUM+ finds the shortest path $p$ between $s$ and $d$, and modifies the graph in such a way that the links along the shortest path are redirected from $d$ to $s$ and their cost is set to the negative of their original cost. If a shortest path $q_1$ exists in the modified graph $G'$, $c(q_1)$ is used to identify the other candidate paths. This is due to the fact that any path $q_k$ with a cost greater than $c(q_1)$ will lead to a pair of link-disjoint paths whose total cost (which is equal to $c(q_k) + c(p)$) is higher than the total cost of the shortest pair. Hence, the **while** loop in Step $4b$ exits when $c(q_k) > c(q_1)$. In $G'$, all the shortest simple paths whose cost is equal to $c(q_1)$ can be obtained using such algorithms as the one given in [116]. Once all equal cost shortest paths are computed in $G'$, their corresponding links that overlap with $p$ in the original graph $G$ are removed to obtain the corresponding shortest pairs of link-disjoint paths in the **while** loop of Step $4d$. Among these link-disjoint paths, the pair that satisfies the corresponding objective of the different problem variants is chosen as a solution.

**Algorithm 2.1 MIN-SUM+(G,s,d)**

1. Find the shortest path $p$ between $s$ and $d$.
2. Graph $G'$ is obtained by directing each link $(u, v)$ of $p$ from $d$ to $s$, and setting the cost of the links on the shortest path as $cost(v, u) = -cost(u, v)$.
3. Find the shortest path $q_1$ in $G'$.
4. **if** $q_1$ exists:
   (a) Set $k := 1$ and $C := c(q_1)$
   (b) **while** $(c(q_k) = C)$
      i. Find the $(k + 1)$-th shortest simple path $q_{k+1}$.
      ii. Set $k := k + 1$
   (c) Set $K = k, k := 1, min\_len := INF$
   (d) **while** $(k \leq K)$
      i. In the original graph $G$, remove the interlacing links between $p$ and $q_k$ to obtain a pair of link-disjoint paths $\{q_{k1}, q_{k2}\}$
      ii. For the **Min-Sum Min-Min** problem:
      **if** $(min\_len > \min\{c(q_{k1}), c(q_{k2})\})$
         A. Set $min\_len := \min\{c(q_{k1}), c(q_{k2})\}$
         B. Set $P_1 := q_{k1}$ and $P_2 := q_{k2}$
      For the **Min-Sum Min-Max** problem:
      **if** $(min\_len > \max\{c(q_{k1}), c(q_{k2})\})$
         A. Set $min\_len := \max\{c(q_{k1}), c(q_{k2})\}$
         B. Set $P_1 := q_{k1}$ and $P_2 := q_{k2}$
      For the **Bounded Min-Sum** problem:
      **if** $(\min\{c(q_{k1}), c(q_{k2})\} \leq \Delta_1$ and $\max\{c(q_{k1}), c(q_{k2})\} \leq \Delta_2)$
         A. Set $P_1 := q_{k1}$ and $P_2 := q_{k2}$
         B. **return** $\{P_1, P_2\}$
      iii. Set $k := k + 1$
   (e) **return** $\{P_1, P_2\}$
5. **else return** no solution

The major operation in MIN-SUM+ is finding all possible shortest paths. For $k > 1$, finding each $k$-th shortest path using the algorithm in [116] takes $O(N(L + N \log N))$ time. Let $K$ be the total number of such paths. Thus, the total running time of MIN-SUM+ is $O(K \cdot N(L + N \log N))$. The size of $K$, which is dependent on the type of network and the distribution of the link costs, can in the worst-case grow exponentially. But by fixing $K$ to a given constant, and exiting the algorithm after at most $K$ link-disjoint paths are computed, heuristic algorithms can be obtained for the three NP-complete problem variants.

We also provide an outline of the WIDE-MIN-SUM algorithm, which is an exact algorithm for the widest min-sum problem. The algorithm begins by computing the shortest pair of link-disjoint paths in the original graph $G$. In each iteration $k$, a new graph $G_{k+1}$ is obtained from $G_k$ ($G_1$ is the original graph) by dropping all links with a bandwidth less or equal to that of the bottleneck link of the shortest link-disjoint paths in $G_k$. This process stops either when there are no link-disjoint paths in $G_k$ or when the total cost of the shortest link-disjoint paths in $G_k$ exceeds that of the shortest pair in the original graph. Finally, the pair with the highest bandwidth is returned. The WIDE-MIN-SUM algorithm is an exact algorithm because,

1. By dropping links with bandwidth less than that of a bottleneck link in $G_k$, only pairs of link-disjoint paths which use any of these links are affected. Hence, no better solution is dropped in the process.

2. If the shortest pair of link-disjoint paths in $G_k$ have a total cost higher than that of the shortest pair in the original graph, dropping more links from $G_k$ will not lead to a better result.

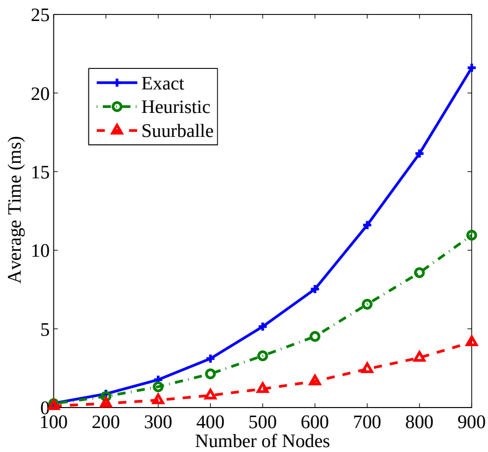
Since the major operation in WIDE-MIN-SUM is finding the shortest link-disjoint paths and in the worst-case $O(L)$ links are dropped before exiting the algorithm, the complexity of the algorithm is $O(L^2 + LN \log N)$.

## 2.4 Results and Discussion

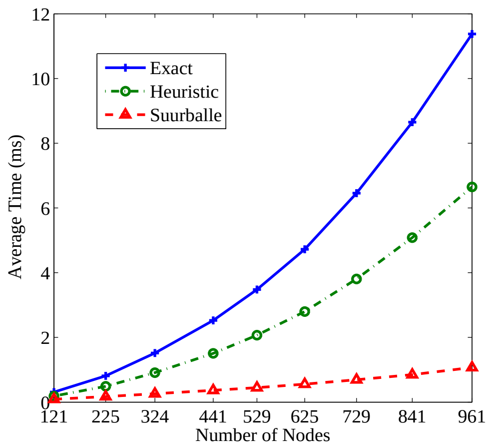
We present simulation results for random (with link density $p = 0.2$) and lattice networks comparing the exact algorithms (MIN-SUM+), heuristic algorithms (MIN-SUM+ with $K = 2$) and the min-sum Suurballe's algorithm [99]. The results we have provided are only for min-sum min-min and min-sum min-max problem variants, because they represent extreme cases of the bounded min-sum problem, where the primary or the backup bounds are tight, respectively. Since the exact algorithm goes through all the possible pairs of min-sum link-disjoint paths, its complexity depends on the total number of such pairs of paths. If there is high granularity in the link costs (e.g., fractional costs) the number of equal cost (min-sum) link-disjoint paths is likely to be small and if there is no granularity (e.g., equal link costs), the heuristic and Suurballe's algorithms will more likely find the optimal solution. Therefore, to increase the possibility of having more min-sum pairs of paths, we use integral link costs that are randomly generated in the range $[1, 100]$. In these simulation results, the number of nodes is varied, and for each number of nodes, we have considered 1000 networks, each network with 1000 randomly generated requests. It can be seen that the heuristic algorithms (with $K = 2$) perform close to the their respective exact algorithms. The exact algorithms also perform in a reasonable time (order of tens of ms) as shown in Figure 2.2 (similar results have been obtained for the min-sum min-max problem).

Table 2.1: The average number of times that the heuristic algorithms and Suurballe's algorithm fail to find the optimal solution out of 1000 requests.

| | Random Networks | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | N | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 |
| min-min | Suurballe | 3.9 | 6.67 | 7.78 | 8.1 | 7.83 | 7.35 | 6.69 | 5.73 | 5.19 |
| | Heuristic | 0.16 | 0.49 | 0.76 | 0.91 | 0.98 | 0.92 | 0.90 | 0.82 | 0.74 |
| min-max | Suurballe | 10.7 | 21.1 | 23.1 | 23.4 | 22.7 | 21.2 | 19.9 | 18.3 | 16.9 |
| | Heuristic | 1.05 | 5.36 | 7.46 | 8.84 | 9.82 | 10.1 | 10.3 | 10.2 | 10.0 |
| | Lattice Networks | | | | | | | | | |
| | N | 121 | 225 | 324 | 441 | 529 | 625 | 729 | 841 | 961 |
| min-min | Suurballe | 2.73 | 3.61 | 4.07 | 4.52 | 4.58 | 5.07 | 5.23 | 5.56 | 5.68 |
| | Heuristic | 0.04 | 0.1 | 0.12 | 0.14 | 0.14 | 0.2 | 0.22 | 0.22 | 0.27 |
| min-max | Suurballe | 5.08 | 6.22 | 6.74 | 7.41 | 7.74 | 7.78 | 8.09 | 8.29 | 8.48 |
| | Heuristic | 0.13 | 0.19 | 0.21 | 0.25 | 0.26 | 0.29 | 0.29 | 0.36 | 0.4 |



(a)



(b)

Figure 2.2: The average times (in ms) of the three algorithms for the min-sum min-min problem variant in (a) random networks, and (b) lattice networks.

## 2.5 Conclusions

In this chapter, we have considered the effect of having secondary objectives in the min-sum link-disjoint paths problem. Even though the min-sum link-disjoint path problem, which minimizes the total cost of the link-disjoint paths, is polynomially solvable, we have shown that NP-complete secondary objectives lead to NP-complete min-sum problems. For these problems, we have provided exact and heuristic algorithms. From simulations results, it can be inferred that our heuristic algorithms in each case outperform Suurballe's algorithm, and the results obtained are close to the corresponding exact algorithms. In addition, because of the reduced search space, the exact algorithms can solve the respective problems in a reasonable running time (in the order of seconds) for fairly large networks (in the order of hundreds of nodes). Therefore, it may also possible to use the exact algorithms for practical purposes.

# Chapter 3

# On-line Survivable Routing and Wavelength Assignment

## 3.1 Introduction

In Chapter 2, we considered how secondary objectives affect the complexity of the min-sum disjoint paths problem. In this chapter, we study the *survivable routing and wavelength assignment* (*SRWA*) problem, where given a set of lightpath requests, the problem is to assign link-disjoint primary and backup lightpaths to each request so that the total number of accepted requests is maximized or the blocking ratio is minimized. As in Chapter 2, since in reality not all the links fail at the same time, we assume the *single-link failure model*, where at most a single link fails at any given time. In addition, since wavelength converters are costly, we consider wavelength-selective WDM networks in this chapter, i.e., any lightpath connection between a source and a destination must have the same wavelength in all links along its route.

For a single request, the SRWA problem can be solved with Suurballe's algorithm [99], if the primary and backup lightpaths use the same wavelength (for different wavelengths, it is NP-complete [6]). But, in practice, lightpath requests arrive over time and the decision to accept or reject a request is made without any knowledge of future requests, yet maintaining the goal to maximize the total number of accepted requests. This version of the SRWA problem is called *on-line SRWA*.

An algorithm is said to be an on-line algorithm if, for any arbitrary input sequence $\sigma$, at any point in the sequence a decision is made based on the input seen so far and without any knowledge of the future. On the other hand, an off-line algorithm is assumed to know the whole input sequence. Thus, the performance of an on-line algorithm $A$ can at best be as good as an optimal, but usually non-implementable, off-line algorithm $OPT$. This performance metric is called *competitive ratio*, and is defined as follows.

**Definition 3.1** *An on-line algorithm A is said to be $\rho$-competitive if for any input sequence $\sigma$,*

$$\mathcal{B}(A,\sigma) \geq \frac{1}{\rho}\mathcal{B}(OPT,\sigma)$$

*where $\mathcal{B}(X,\sigma)$ is the number of accepted requests by algorithm $X$ for the input sequence $\sigma$. The smallest such $\rho$ is called the competitive ratio of the algorithm.*

Usually, constant and logarithmic competitive ratios are considered good, while linear and exponential competitive ratios are considered bad. The outline of this chapter is as follows. In Section 3.2, we provide algorithms for the on-line SRWA problem with constant and logarithmic competitive ratios for specific networks. In Section 3.3, we introduce rerouting of lightpaths to improve the practical performance of on-line routing. We discuss a related problem called the *minimum disruption link-disjoint paths* (*MDLDP*) problem and provide two 2−approximation algorithms for solving it. An algorithm is a *2-approximation* algorithm for MDLDP if for any request, the number of lightpaths rerouted by its solution is at most twice that of the optimal algorithm. In Sections 3.4 and 3.5, we employ these algorithms as heuristics to solve the on-line SRWA with rerouting problem for requests of infinite and finite duration, respectively. In Section 3.6, we consider shared on-line SRWA and provide a heuristic algorithm for it. Section 3.7 presents our conclusions.

## 3.2 On-line SRWA

The on-line survivable routing and wavelength assignment (SRWA) problem is defined as follows.

**Problem 3.1** ***On-line SRWA:*** *The physical optical network is modeled as an undirected graph $G(\mathcal{N},\mathcal{L})$, where $\mathcal{N}$ is a set of $N$ nodes and $\mathcal{L}$ is a set of $L$ links. Each fiber link has a set of $W$ wavelengths, $\mathcal{W}=\{\lambda_1,\lambda_2\ldots,\lambda_W\}$. A sequence of lightpath requests $\sigma$ arrive over time. Each request $f\in\sigma$ is represented by $(s_f,d_f)$, where $s_f,d_f\in\mathcal{N}$ are its source and destination nodes, respectively. The on-line SRWA problem is to allocate for each request link-disjoint primary and backup lightpaths such that (1) the same wavelength is used on all links of the primary and backup lightpaths, (2) no two lightpaths having the same wavelength can share a link, and (3) the decision to accept or reject a request is based only on the input sequence seen so far. The objective is to maximize the number of accepted requests.*

Before addressing the on-line SRWA problem, we consider the on-line SRWA problem without survivability (on-line RWA) and other related problems that have been studied in the literature.

**Problem 3.2** ***On-line Maximum Disjoint Paths (MDP) Problem:*** *Given are a graph $G(\mathcal{N}, \mathcal{L})$ and a sequence of requests. For each request $(s_f, d_f)$, find a path $P_f$ that connects $s_f$ and $d_f$ such that no two paths share the same link. The objective is to maximize the total number of accepted requests.*

The MDP problem is NP-complete [60]. Since lightpaths on the same wavelength are not allowed to share a link, the on-line MDP problem is equivalent to the on-line RWA problem with $W = 1$. Awerbuch *et al.* [10] have shown that if there is a $\rho$-competitive algorithm for the on-line MDP problem, then a $(\rho + 1)$-competitive algorithm can be obtained for the on-line RWA problem by employing the on-line MDP algorithm on each wavelength.

The on-line MDP problem has been widely studied in the literature. The $\Omega(N^a)$, where $a = \frac{2}{3}(1 - \log_4 3)$, lower bound given by Bartal *et al.* [13] for randomized on-line algorithms shows that it is not possible to find a good competitive ratio for general networks. In fact, most of the work done so far has been restricted to special networks such as lines [1] [9] [38], trees [9] [11], lattices [11] [61], tree of rings [5], etc. In the case of line and tree graphs, the path between a given source and destination pair is unique. Therefore, the on-line MDP problem is reduced to the path coloring problem.

Awerbuch *et al.* [9] dealt with non-preemptive randomized algorithms for line and tree networks. Their algorithms are based on a paradigm called "classify and randomly select," where requests are grouped in classes and the algorithm randomly chooses which class of inputs are served. For a line network, they provided an $\Omega(\log n)$ lower bound on the competitive ratio and an optimal $O(\log n)$-competitive algorithm. For an $N = n \times n$ lattice network, Kleinberg *et al.* [61] proposed an improved $O(\log N)$-competitive randomized algorithm. Anad *et al.* [5] dealt with the on-line MDP problem in a tree-of-rings. They provided an $O(\log D)$-competitive non-preemptive randomized algorithm, where $D$ is the minimum possible diameter of a tree resulting from the tree of rings by deleting one link from every ring.

**Problem 3.3** ***On-line $k$ Maximum Disjoint Paths (k-MDP) Problem:*** *Given are a graph $G(\mathcal{N}, \mathcal{L})$ and a sequence of requests. For each request $(s_f, d_f)$, find $k$ link-disjoint paths $P_{f1}, \ldots, P_{fk}$ that connect $s_f$ and $d_f$ such that no two paths of different requests share the same link. The objective is to maximize the total number of accepted requests.*

A simple upper-bound of any non-preemptive on-line algorithm for $k$-MDP is $O(\frac{L}{k})$. Suurballe's [99] algorithm ($k = 2$) has a competitive ratio equal to this upper-bound. For example in Figure 3.1, if the input sequence is $(s, d)$ followed by $(s, a_1)$, $(a_1, a_2)$,. . ., $(a_y, d)$, $(s, b_1)$, $(b_1, b_2)$,. . ., $(b_y, d)$ and all links have equal cost, the off-line algorithm accepts $O(N)$ requests (i.e., all except the first), but the on-line algorithm accepts only the first two requests. Since in this example $L = O(N)$, the competitive ratio is of the same order as the upper-bound.

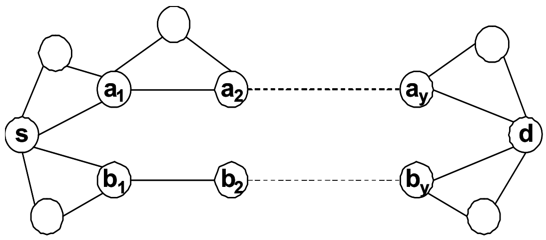



Figure 3.1: An example where Suurballe's algorithm attains the upper bound.

Using the same argument provided by Awerbuch *et al.* [10], a $(\rho + 1)$-competitive algorithm for the on-line SRWA problem ($W > 1$) can be derived from a $\rho$-competitive algorithm of the on-line 2-MDP problem. Hence, in the remainder of this section, we provide algorithms and corresponding competitive ratios for the on-line 2-MDP problem, which forms the basis for the on-line SRWA problem, in star-of-rings, tree-of-rings, and lattice networks. Even though these are simple networks, not only do they help us gain insight into the problem, but they are also used in real networks (e.g., the SURFnet5 network in the Netherlands resembles a star-of-rings[1]).

### 3.2.1 Star-of-rings network

**Algorithm 3.1** $Star\_Alg(G, s, d)$

- Accept a request if it is the first request so far that uses the ring(s) to which the source and destination nodes belong.
- Reject, otherwise.

$Star\_Alg(G, s, d)$ is 2-competitive if the number of rings is greater than 1. For a single ring, it is optimal. Figure 3.2 shows an example where $Star\_Alg(G, s, d)$ is 2-competitive for the input sequence $(a, b)$, $(b, c)$, $(a, e)$. In this example, the on-line algorithm accepts only the first request, while the off-line algorithm accepts the last two requests.

### 3.2.2 Tree-of-rings network

For tree-of-rings, we provide $Tree\_Alg(G, s, d)$. From [9], it follows that $Tree\_Alg(G, s, d)$ is $O(\log \Upsilon)$-competitive, where $\Upsilon$ is the number of rings.

[1] http://www.surfnet.nl/en

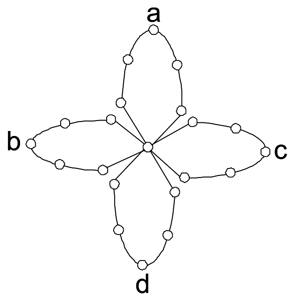



Figure 3.2: A star of rings containing four rings.

**Algorithm 3.2** $Tree\_Alg(G, s, d)$

- Replace each ring by a single link so that the whole tree of rings is substituted by the underlying tree topology.
- Each 2-MDP request in the tree of rings is equivalent to a corresponding MDP request in the underlying tree.
- Use the algorithm of Awerbuch *et al.* [9], which has $O(\log N)$ competitive ratio for a tree of $N$ nodes, to solve the on-line MDP problem.

### 3.2.3 Lattice network

The $O(\log N)$-competitive algorithm given by Kleinberg and Tardos [61] for the on-line MDP problem can, with a slight modification, be used for solving the on-line 2-MDP problem with an $O(\log N)$-competitive ratio. For the sake of completeness, we provide an outline of their algorithm. Given an $N = n \times n$ lattice network:

- Classify each request as either "short" or "long," depending on the shortest distance $dist(s_f, d_f)$ (in terms of hopcount) between its source and destination nodes. A request is said to be short if the distance $dist(s_f, d_f) < 16\gamma \log n$ for a given constant $\gamma > 1$; and long otherwise.
- Choose (randomly) to accept only short or only long calls.
- Create a "simulated network" whose vertices are subsquares of the original $n \times n$ lattice network and each of its links contain $O(\log n)$ links of the original network.
- Map the requests onto the simulated network (details in [61]).

- For long requests, a modified version of the AAP algorithm [8] is used to route the requests as shown in [61]. This algorithm can be modified to find a pair of link-disjoint paths instead of a single path.

- For short requests, the algorithm from [61] can also be modified to find a pair of link-disjoint paths.

## 3.3 On-line SRWA with Rerouting

In Section 3.2, we provided algorithms for the on-line 2-MDP problem in specific networks, which can be used to derive corresponding algorithms for the on-line SRWA problem. Unfortunately, it is not possible to attain a good competitive ratio for general networks [13]. In this section, we explore the idea of rerouting lightpaths to improve performance. Although rerouting does not improve the competitive ratio, we show through simulations that it can increase the acceptance rate considerably. In wavelength-selective WDM networks, a rerouting procedure may be path rerouting (i.e., changing the route of a lightpath while keeping the wavelength), wavelength rerouting (i.e., changing the wavelength while keeping the path) or a combination of both. Compared to path rerouting, wavelength rerouting does not need extra path computation (as it retains the same path), facilitates control and, if the rerouted lightpath is moved to a vacant route on another wavelength, it incurs less traffic disruption [65]. We therefore focus on wavelength rerouting.

Generally, the wavelength rerouting problem is NP-complete [65]. It consists of solving the three possible scenarios presented below. The second and the third scenarios make the problem hard to solve. Figure 3.3 shows the different scenarios. The labels on the links represent already existing lightpaths.

1. When the lightpaths to be rerouted are on the same wavelength, they can be moved to vacant wavelengths in parallel without any conflict (since they do not share links). For example, in Figure 3.3(a), a new lightpath from node 1 to 5 can be accepted on wavelength $\lambda_2$ by rerouting lightpath $p_3$ to $\lambda_1$ and $p_4$ to $\lambda_3$ in parallel.

2. When the lightpaths are on different wavelengths, moving to vacant wavelengths can be done sequentially while checking for conflicts. For example, in Figure 3.3(b), a new lightpath from node 1 to 5 can be accepted on $\lambda_1$ by first rerouting $p_4$ to $\lambda_3$ and then $p_1$ to $\lambda_2$.

3. Moving to a vacant wavelength may not be sufficient, and it may be necessary to swap the wavelengths of lightpaths. For example, in Figure 3.3(c), a new lightpath from node 1 to 4 can be accepted on $\lambda_2$ by swapping the wavelengths of $p_2$ and $p_3$.

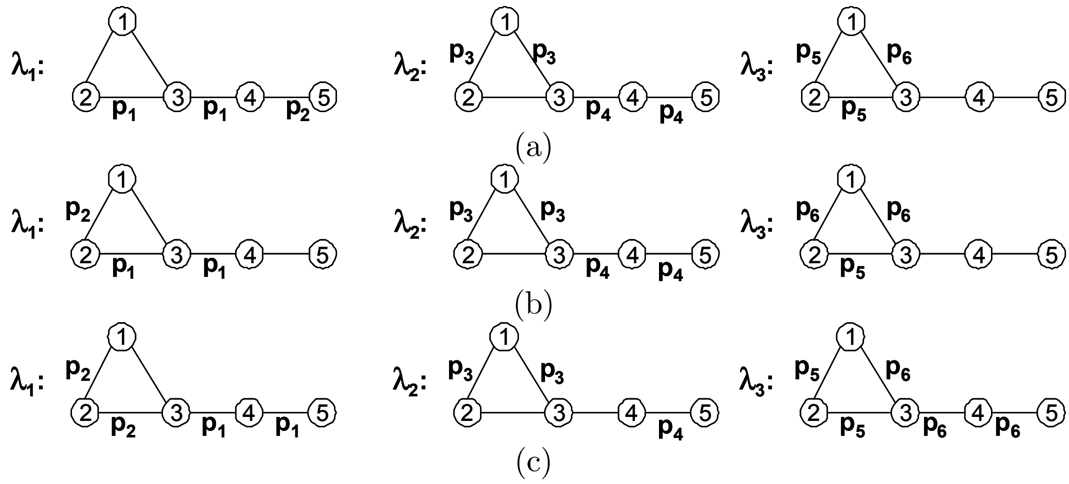



Figure 3.3: Different scenarios of wavelength rerouting: (a) moving to vacant, (b) sequential rerouting, and (c) swapping.

In the literature and the remainder of this chapter, the term *wavelength rerouting* is used to refer to the reduced problem, i.e., assigning a lightpath by moving existing lightpaths on the same wavelength to vacant wavelengths in parallel. Xue [109] has shown that this problem can be solved in $O(WN \log N + WL)$ time.

On-line SRWA with wavelength rerouting involves assigning link-disjoint primary and backup lightpaths for new requests by rerouting, if necessary, already existing lightpaths. When rerouting lightpaths, the number of rerouted lightpaths should be kept to a minimum. This leads us to consider the minimum-disruption link-disjoint paths (MDLDP) problem. The MDLDP problem is NP-complete when the primary and backup lightpaths use different wavelengths. However, it is polynomially solvable for the same wavelength [105]. We consider the polynomially-solvable version.

**Problem 3.4** ***Minimum Disruption Link-Disjoint Paths (MDLDP):*** *The physical optical network is modeled as an undirected graph $G(\mathcal{N}, \mathcal{L})$, where $N = |\mathcal{N}|$ and $L = |\mathcal{L}|$. Each fiber link has a set $\mathcal{W} = \{\lambda_1, \lambda_2, \ldots, \lambda_W\}$ of $W$ wavelengths. Given a request $f$, the MDLDP problem is to allocate on the same wavelength link-disjoint primary and backup lightpaths for request $f$, while minimizing the number of lightpaths to be rerouted.*

Wan and Liang [105] provided an $O(WL^5 \log N)$ exact algorithm for solving the MDLDP problem. We refer to this algorithm as $WLA$. $WLA$ has a very high running time and requires a large amount of memory. This makes it less suitable, especially in an on-line setting where the algorithm has to be invoked whenever a new request arrives. We propose two 2-approximation algorithms with considerably less running times and memory requirements.

### 3.3.1 2-Approximation Algorithms for MDLDP

We provide two 2-approximation algorithms for MDLDP: $MSA$ and $ESA$. $MSA$ is a modified version of Suurballe's algorithm [99] with a running time of $O(WN \log N + WL)$ and $ESA$ is an extended algorithm with a running time of $O(WN^2 \log N + WNL)$. This is a significant reduction from the $O(WL^5 \log N)$ running time of the exact $WLA$ algorithm with at most twice as much lightpaths being rerouted.

In our notation, we use $p$ to represent a lightpath and $P$ to represent any path. A lightpath on wavelength $\lambda_i$ is said to be *reroutable*, if and only if all of its links are free on at least one other wavelength $\lambda_j$. A path $P$ from $s$ to $d$ is said to *traverse* a lightpath $p$ if it shares at least one link with $p$. Let $\mathcal{P}_k$ be the set of lightpaths on wavelength $\lambda_k$; $\mathcal{P}_k^{'} \subseteq \mathcal{P}_k$ be the set of reroutable lightpaths on wavelength $\lambda_k$; $\mathcal{P}_k^{''} = \mathcal{P}_k \backslash \mathcal{P}_k^{'}$ be the set of non-reroutable lightpaths on wavelength $\lambda_k$; and $\mathcal{W}_{(u,v)}$ be the set of free wavelengths on fiber link $(u, v)$.

We identify $W$ subgraphs, $G_k = G(\mathcal{N}, \mathcal{L}_k)$, $\mathcal{L}_k = \{(u, v) \in \mathcal{L} \mid \lambda_k \in \mathcal{W}_{(u,v)}$ or $\exists\ p \in \mathcal{P}_k^{'}$ such that link $(u, v)$ belongs to lightpath $p\}$. The cost of a link $(u, v)$ in subgraph $G_k$ is $cost_k(u, v) = \epsilon$, if $(u, v)$ is a free link, where[2] $2N\epsilon < 1$; $cost_k(u, v) = 1$ otherwise. However, the cost $cost_k(P)$ of a path $P$ in subgraph $G_k$ is the sum of the costs of its free links and the number of *distinct* reroutable lightpaths traversed by $P$, i.e., multiple links belonging to a lightpath are counted only once. Thus, the shortest path between two nodes traverses the minimum number of reroutable lightpaths. Note that any lightpath that is traversed by the shortest path is encountered only once.

---

**Algorithm 3.3** $MSA(G, s, d)$

---

1. For each $G_k$, $k = 1, \ldots, W$
   (a) In graph $G_k$, find the shortest path from $s$ to $d$.
   (b) Graph $G_k^{'}$ is obtained by directing each link $(u, v)$ of the shortest path from $d$ to $s$, setting the cost of the free links on the shortest path as $cost_k(v, u) = -cost_k(u, v)$ and the cost of *all links* of lightpaths that are traversed by the shortest path to zero.
   (c) Find the shortest path from $s$ to $d$ in $G_k^{'}$.
   (d) If the shortest path exists in $G_k^{'}$, remove all the overlapping links between the two paths in $G_k$ to obtain the solution.
2. Choose the best solution among all wavelengths.

---

[2] Using such a cost, the longest possible link-disjoint paths made up of only free links have a total cost that is less than any link-disjoint pair of paths that cross a lightpath.

In Step $1a$ of the $MSA$ algorithm, we find the shortest path from $s$ to $d$ (using an algorithm such as the one given in [109]). In Step $1b$, the cost of all links belonging to lightpaths traversed by the shortest path is set to zero so that these links are preferred in the second path and the lightpaths are not counted twice. Similarly, the cost of free links on the shortest path is set to $-\epsilon$.

**Theorem 3.1** *$MSA$ is a 2-approximation algorithm for the MDLDP problem.*

**Proof.** Since the best solution is chosen after independently considering each wavelength, it suffices to consider only the wavelength that provides the best solution. Assume that for this wavelength, given a solution of $MSA$ that traverses a total of $K$ lightpaths, there is an optimal solution that traverses less than $\frac{K}{2}$ lightpaths, which would violate the claim of 2-approximation. Our intention is to prove that the assumption is wrong.

Let $\ell(P)$ represent the number of lightpaths traversed by a path $P$ and $\ell(\{P_1, P_2\})$ represent the number of distinct lightpaths traversed by paths $P_1$ and $P_2$, where $\ell(\{P_1, P_2\}) \leq \ell(P_1) + \ell(P_2)$.

Let $\{P_1^*, P_2^*\}$ be the optimal solution. In $MSA$, let $P_1$ be the first shortest path that is obtained in Step $1a$ and $P_2$ be the second shortest path that is obtained in Step $1c$.

Let $\mathcal{Q}$ be the set of alternating lightpaths of the optimal solution $\{P_1^*, P_2^*\}$, i.e., lightpaths with segments in both $P_1^*$ and $P_2^*$. Let $\mathcal{S}$ be the set of links of lightpaths $p \in \mathcal{Q}$.

$\ell(\{P_1^*, P_2^*\}) < \frac{K}{2}$ implies that $\ell(P_1^*) < \frac{K}{2}$ and $\ell(P_2^*) < \frac{K}{2}$. Hence, the first shortest path returned by $MSA$ must have $\ell(P_1) < \frac{K}{2}$. Since $\ell(\{P_1, P_2\}) = K$, the second shortest path returned by $MSA$ should have $\ell(P_2) > \frac{K}{2}$. But, $MSA$ can find a path $P_2$ from the set of links of $P_1^*$, $P_2^*$ and $\mathcal{S}$. If $P_1$ also contains any of these links, they are redirected in Step $1b$ of $MSA$ and are assigned a cost of zero. Since no new lightpaths are added $\ell(P_2) < \frac{K}{2}$, which is a contradiction. ■

The 2-approximation is attained in the worst-case when $\ell(P_1) = \ell(P_2) = \ell(\{P_1^*, P_2^*\})$ and $P_1$ and $P_2$ do not have common lightpaths as shown in Figure 3.4(a). $P_1 = \{s, 3, d\}$, $P_2 = \{s, 4, d\}$, $P_1^* = \{s, 1, 2, d\}$, and $P_2^* = \{s, 5, 6, d\}$; $\ell(\{P_1, P_2\}) = 2$ and $\ell(\{P_1^*, P_2^*\}) = 1$.

The example in Figure 3.4(a) can exactly be solved if $P_1$ leaves the source node through node 1 or node 5. We can achieve this by extending the $MSA$ algorithm so that it checks the shortest path through any given node $u \in \mathcal{N} \backslash \{s, d\}$. This is exactly what our extended algorithm $ESA$ does. As can be seen in Section 3.3.3, $ESA$ has a significantly improved performance in solving the MDLDP problem. But, it fails for cases like the one in Figure 3.4(b), where $P_1 = \{s, 1, 3, d\}$, $P_2 = \{s, 2, 3, 5, d\}$, $P_1^* = \{s, 1, 3, 4, d\}$, and $P_2^* = \{s, 2, 3, 5, d\}$; $\ell(\{P_1, P_2\}) = 3$ and $\ell(\{P_1^*, P_2^*\}) = 2$.

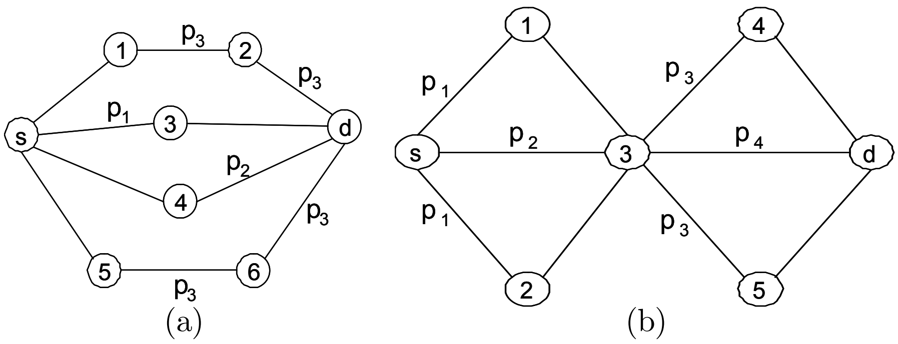



Figure 3.4: (a) A worst-case for MSA that leads to a 2-approximation and (b) an example where ESA fails.

---

**Algorithm 3.4** $ESA(G, s, d)$

---

1. For each $G_k$, $k = 1, \ldots, W$

   (a) For each node $u \in \mathcal{N} \backslash \{s, d\}$:

      i. In graph $G_k$, find the shortest path $P_{s-u}$ from $s$ to $u$.

      ii. Graph $G_k^{'}$ is obtained from $G_k$ by setting the cost of all links on $P_{s-u}$ and each link belonging to lightpaths on $P_{s-u}$ to infinity except for links of the lightpath (if any) in the last link of $P_{s-u}$. For the lightpath in the last link, all its links except the ones in $P_{s-u}$ will have a cost of zero.

      iii. In graph $G_k^{'}$, find the shortest path $P_{u-d}$ from $u$ to $d$. If $P_{s-u}$ and $P_{u-d}$ share nodes, go to Step 1$a$-$i$ if there are remaining nodes whose shortest paths have not been found, otherwise go to Step 1$b$. If $P_{s-u}$ and $P_{u-d}$ do not share nodes, the shortest path through $u$ is found by concatenating the two.

      iv. Graph $G_k^{''}$ is obtained from $G_k$ by directing each link $(u, v)$ along the shortest path from $d$ to $s$. The cost of free links on the shortest path is set to $cost_k(v, u) = -cost_k(u, v)$ and the cost of all links belonging to lightpaths on the shortest path is set to zero.

      v. In graph $G_k^{''}$, find the shortest path from $s$ to $d$.

      vi. If the shortest path exists, remove all the overlapping links.

   (b) Choose the best solution among all nodes.

2. Choose the best solution among all wavelengths.

---

In $ESA$, for each node $u \in \mathcal{N} \backslash \{s, d\}$, we find link-disjoint paths from $s$ to $d$, where the first path is forced to go through $u$. In Step 1$a$-$ii$ of the $ESA$ algorithm, the cost of all links on $P_{s-u}$ and all links belonging to lightpaths on $P_{s-u}$ (except those of the lightpath on the last link, if there is any) is set to infinity. This is to prevent the same links from being used again in $P_{u-d}$ and to make sure that any lightpath in $P_{s-d}$ is traversed in at most one segment. For the lightpath on the last link, since our interest is to find the shortest path from $s$ to $d$ through $u$, the lightpath can still be encountered on a segment *just* after node $u$. Therefore, its links, except those in $P_{s-u}$, will have a cost of zero. In Step 1$a$-$iii$, the shortest path from $u$ to $d$ is found. If $P_{s-u}$ and $P_{u-d}$ share nodes, then the algorithm does not proceed to finding the second shortest path. Instead, it skips to searching for the solutions of the remaining nodes. Once the path through $u$ is found by concatenating $P_{s-u}$ and $P_{u-d}$, the links on this path are directed from $d$ to $s$ in Step 1$a$-$iv$. In Step 1$b$, all the solutions are compared and the one that traverses the minimum number of lightpaths is chosen. In case of a tie, the one with the *smallest hopcount* is chosen. Since $ESA$ includes $MSA$, it is at worst a 2-approximation algorithm.

### 3.3.2 Reroutability Status Update Procedure

Once a lightpath request is accepted and its link-disjoint lightpaths are determined, it affects the reroutability of other lightpaths. These lightpaths include the rerouted lightpaths, and lightpaths that are using the same link, but on different wavelengths. In addition, the reroutability of the new lightpaths has to be identified. Once a request is accepted, its primary and backup lightpaths are treated independently, i.e., each can be rerouted to a different wavelength independently of the other. Hence, as in [73], for each lightpath, we dynamically keep track of such information as its hopcount, its wavelength, how many of its links are free on other wavelengths and to which other wavelengths it can be rerouted. This is done as follows.

1. When a new lightpath $p$ is assigned without rerouting other lightpaths on wavelength $\lambda_k$:

   - We create new reroutability status information for $p$, e.g., how many of its links are free on other wavelengths and the wavelengths it can be rerouted to. This takes $O(NW)$ time.
   - After checking whether $p$ is reroutable or not, we assign the costs of its links on wavelength $\lambda_k$. This takes $O(N)$ time.
   - In addition, the reroutability status information of lightpaths using the same fiber link, but on other wavelengths, should be updated. If $q$ is such a lightpath, the number of its links that are free on wavelength $\lambda_k$ is decremented by one for each link that $p$ and $q$ have in common. Thus, if $q$ was reroutable

to wavelength $\lambda_k$, it is not any more. Since, in the worst-case, there are $O(NW)$ such lightpaths, this takes $O(NW)$ time.

2. When a new lightpath $p$ is assigned by rerouting some lightpaths on wavelength $\lambda_k$:
   - All the aforementioned operations are performed.
   - If $q$ is a rerouted lightpath, the costs of its links on the new wavelength, and its reroutability status on $\lambda_k$ should be updated. This takes $O(N)$ time and in the worst-case $O(N)$ lightpaths are rerouted. Therefore, the total running time is $O(N^2)$.

3. When the holding time of lightpath $p$ expires:
   - All its links on wavelength $\lambda_k$ will be free links and their cost is updated accordingly. This takes $O(N)$.
   - For any lightpath $q$ that uses the same fiber link, but a different wavelength, the reroutability status information is updated. The number of its free links on wavelength $\lambda_k$ is increased by one and if this equals to the hopcount of $q$, then $q$ is reroutable to $\lambda_k$. This will take $O(NW)$ time.

The total running time of the reroutability update procedure is $O(N^2 + NW)$. We employ this procedure when solving the on-line SRWA problem using the MDLDP algorithms.

### 3.3.3 Simulation Study

We proceed to compare our 2-approximation algorithms ($MSA$ and $ESA$) with the exact algorithm ($WLA$) in solving the MDLDP problem. In order to simulate a wide range of possibilities, we generate dynamic traffic, where requests arrive according to a Poissonian distribution (arrival rate $a$) with exponential holding times of mean 1. For each request, we record the results of our algorithms in comparison to $WLA$. The *approximation ratio* represents the ratio of the number of lightpaths traversed by an approximation algorithm to the number of lightpaths traversed by $WLA$. It is averaged for all accepted requests over 20 iterations, each iteration with 5000 requests. The source and destination nodes are randomly selected with all nodes having equal probability of being selected.

We consider three networks: a USANET network [43][52] (Figure 3.5), an Erdös-Rényi random network ($N = 50$, link density $p = 0.2$, i.e., the average total number of links is $p \cdot \binom{N}{2}$), and a $7 \times 7$ lattice network, each with $W = 10$ wavelengths. In all our simulations, the approximation ratio attained by $ESA$ never exceeded 1.00004. The

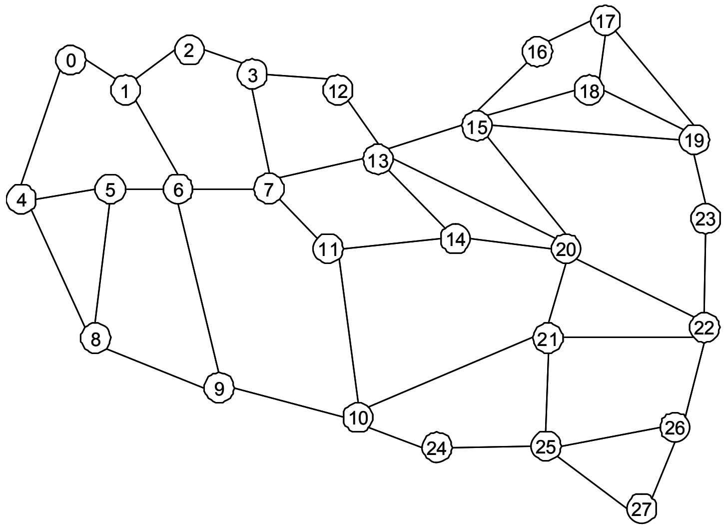



Figure 3.5: A 28-node USANET network topology.

approximation ratios of both $ESA$ and $MSA$ in comparison to the exact algorithm were much smaller than 2. Table 3.1 shows the average simulated approximation ratios of $MSA$, in terms of the number of lightpaths rerouted, when compared to $WLA$, which returns the exact number for a given request.

Table 3.1: Approximation ratios of the $MSA$ algorithm in the three networks for different arrival rates ($a$) for $W = 10$.

| USANET | | RANDOM | | LATTICE | |
|---|---|---|---|---|---|
| $a$ | Appr. ratio | $a$ | Appr. ratio | $a$ | Appr. ratio |
| 10 | 1.0105 | 20 | 1.0185 | 20 | 1.0208 |
| 15 | 1.0095 | 40 | 1.0209 | 30 | 1.0172 |
| 20 | 1.0055 | 60 | 1.0309 | 40 | 1.0147 |
| 25 | 1.0170 | 80 | 1.0326 | 50 | 1.0119 |
| 30 | 1.0128 | 110 | 1.0275 | 60 | 1.0093 |
| 35 | 1.0100 | 120 | 1.0169 | 70 | 1.0073 |

In Sections 3.4 and 3.5, we use the aforementioned MDLDP algorithms to heuristically solve infinite and finite duration on-line SRWA, respectively. For each case, we compare the performances of $MSA$, $ESA$ and $WLA$.

## 3.4 Infinite Duration On-line SRWA

In the infinite duration on-line SRWA problem, lightpaths stay indefinitely once they arrive. The off-line SRWA problem, where all the requests are known beforehand, can be described as a network flow problem. For this, we provide ILP formulations under two cases: *Case1*, when both the primary and backup lightpaths have to use the same wavelength and *Case2*, when they can use different wavelengths.

**Indices:**

| | |
|---|---|
| $f = 1, \dots, F$ | ID of requests ($F$ in total) |
| $\lambda = 1, \dots, W$ | ID of wavelengths |
| $\mathcal{N}(u)$ | Set of nodes adjacent to node $u$ |

**Variables (integers):**

$\gamma_{f,\lambda,u,v}$ is 1 (or $-1$ depending on the flow direction) if the primary or backup lightpaths of request $f$ use wavelength $\lambda$ on link $(u, v) \in \mathcal{L}$; 0 otherwise.

$x_{f,\lambda}$ *Case1* (*same wavelength*): is 1 if request $f$ is accepted and uses wavelength $\lambda$; 0 otherwise.
*Case2* (*different wavelengths*): is 0 if neither the primary nor the backup lightpaths of request $f$ are on wavelength $\lambda$; 1 if either the primary or the backup lightpath of request $f$ is on wavelength $\lambda$; 2 if both the primary and the backup lightpaths of request $f$ are on wavelength $\lambda$.

$y_f$ is 1 if request $f$ is accepted; 0 otherwise.

**Objective:**

Maximize the number of accepted requests.

$$Maximize : \sum_{f=1}^{F} y_f$$

**Constraints**

*Antisymmetry constraints*: Since the graph is undirected, the flow is in both directions.

$$\gamma_{f,\lambda,u,v} = -\gamma_{f,\lambda,v,u} \ \forall (u, v) \in \mathcal{L}; 1 \leq f \leq F; 1 \leq \lambda \leq W.$$

*Conservation constraints*: If a given node is not the source or destination of a given request, then any flow related to the request that enters the node has to leave the node.

$$\sum_{v \in \mathcal{N}(u)} \gamma_{f,\lambda,u,v} = 0$$

$$\forall u \in \mathcal{N} \backslash \{s_f, d_f\}; 1 \leq f \leq F; 1 \leq \lambda \leq W.$$

*Capacity constraints*: Only a single lightpath can use a given wavelength on a certain link.

$$\sum_{f=1}^{F} \gamma_{f,\lambda,u,v} \leq 1 \qquad \forall (u, v) \in \mathcal{L}; 1 \leq \lambda \leq W.$$

*Disjointedness constraints*: The primary and backup lightpaths of a request should be link-disjoint.

$$\sum_{\lambda=1}^{W} \gamma_{f,\lambda,u,v} \leq 1 \quad \forall (u,v) \in \mathcal{L}; 1 \leq f \leq F.$$

**Equations**

Lightpaths of a request on a given wavelength.

$$\sum_{v \in \mathcal{N}(s_f)} \gamma_{f,\lambda,s_f,v} = \beta x_{f,\lambda} \qquad 1 \leq f \leq F; 1 \leq \lambda \leq W.$$
$$\sum_{v \in \mathcal{N}(d_f)} \gamma_{f,\lambda,v,d_f} = \beta x_{f,\lambda} \qquad 1 \leq f \leq F; 1 \leq \lambda \leq W.$$

where $\beta = 2$ for *Case1* and $\beta = 1$ for *Case2*.

An accepted request has link-disjoint primary and backup lightpaths.

$$\sum_{\lambda=1}^{W} x_{f,\lambda} = \varphi y_f \qquad 1 \leq f \leq F.$$

where $\varphi = 1$ for *Case1* and $\varphi = 2$ for *Case2*.

Solving the given ILP formulation for large networks and high number of requests is not feasible. Therefore, we use the algorithms of the MDLDP problem to solve the on-line SRWA problem sequentially. Clearly, this approach will not guarantee an optimal solution. However for small networks, we show that the results obtained by these algorithms are close to the optimal off-line solution (given by the ILP for *Case2*), which does not need rerouting. Tables 3.2 and 3.3 show comparisons, in terms of the number of rejected requests, of our algorithms ($MSA$ and $ESA$), $WLA$ and without rerouting ($W/R$) against the optimal ILP formulation for small random networks with link density $p$ ($N = 10$ with 20 requests and $N = 12$ with 30 requests) and $W = 4$. We observe that rerouting performs better (though marginally, since the network is small and the number of requests are few) than without rerouting and our algorithms perform as good as (and at times better than) $WLA$.

Table 3.2: Number of rejected requests for $N = 10$ and 20 requests.

| $p$ | $W/R$ | $WLA$ | $MSA$ | $ESA$ | *Optimal* |
|---|---|---|---|---|---|
| 0.2 | 15 | 14 | 14 | 14 | 12 |
| 0.3 | 13 | 12 | 12 | 11 | 7 |
| 0.4 | 8 | 6 | 6 | 6 | 3 |
| 0.5 | 7 | 6 | 6 | 7 | 3 |
| 0.6 | 6 | 5 | 5 | 4 | 1 |

Table 3.3: Number of rejected requests for $N = 12$ and 30 requests.

| $p$ | $W/R$ | $WLA$ | $MSA$ | $ESA$ | *Optimal* |
|---|---|---|---|---|---|
| 0.2 | 23 | 22 | 22 | 22 | 21 |
| 0.3 | 25 | 25 | 25 | 25 | 24 |
| 0.4 | 12 | 11 | 11 | 10 | 6 |
| 0.5 | 5 | 3 | 3 | 3 | 2 |
| 0.6 | 4 | 3 | 3 | 3 | 0 |

## 3.5 Finite Duration On-line SRWA

Finite duration SRWA requests arrive to and depart from the network over time. Thus, any two lightpaths can share resources as long as they do not overlap in time. We use the algorithms of MDLDP as heuristics to solve the finite duration on-line SRWA problem.

We use the same scenarios as in Section 3.3.3 for our simulations. Figures 3.6-3.7 show comparisons of the performance of our algorithms ($MSA$ and $ESA$) with $WLA$ in terms of the percentage of rejections. The given results are (a) for different arrival rates with a constant number of nodes, and (b) for different number of nodes with a constant arrival rate. A comparison of these algorithms with the case of no rerouting ($W/R$) shows that rerouting of lightpaths decreases the percentage of rejections significantly. In addition, *we observe that both $MSA$ and $ESA$ perform similarly to $WLA$, which has much higher running time and memory requirements.* The need to have a small running time becomes more pronounced in an on-line setting, where the algorithm has to be invoked repeatedly whenever a request arrives.

## 3.6 Shared On-line SRWA

Until now, we have only considered dedicated survivable routing, where backup lightpaths do not share resources. This approach is more suitable for when a copy of the data signal is sent on both the primary and backup lightpaths. However, if the data signal is sent on the backup lightpath only after the failure of the primary lightpath, it is desirable to use shared survivable routing. In shared survivable routing, two backup lightpaths can share resources as long as their respective primary lightpaths do not fail simultaneously. Since we are considering single link failures, if two primary lightpaths do not have a common link, their backup lightpaths can share links. Here also, we employ wavelength rerouting to improve the acceptance rate under dynamic traffic. However, unlike the (reduced) wavelength rerouting problem in Section 3.3, the shared wavelength rerouting problem where backup lightpaths can possibly share links is not polynomially solvable as shown below.

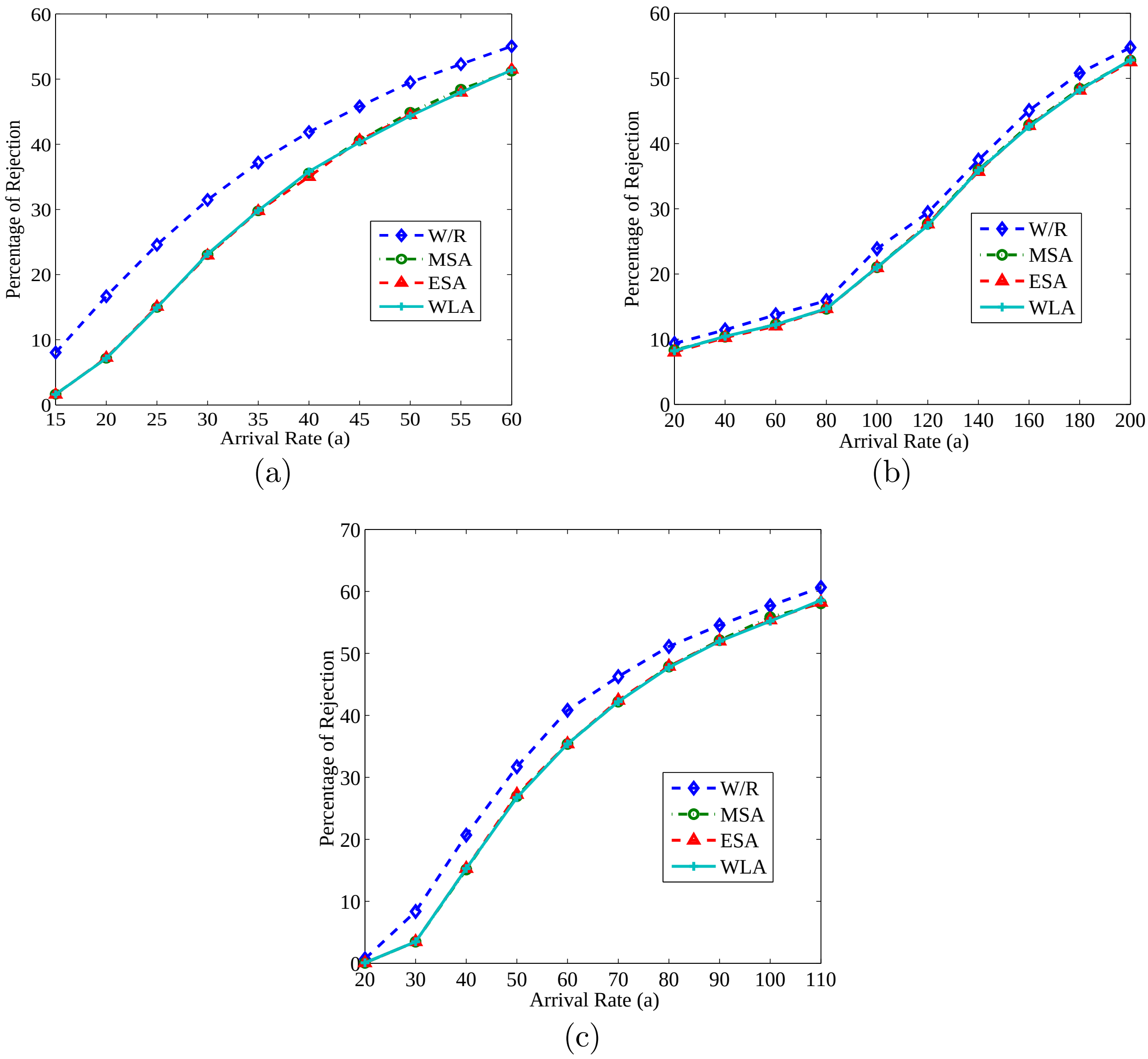



Figure 3.6: Rejection rates of $MSA$, $ESA$, $WLA$ and without rerouting for different arrival rates of (a) the USANET network ($W = 10$ and 5000 requests), (b) for random networks ($N = 50$, $p = 0.2$, $W = 10$, and 5000 requests), and (c) lattice networks ($N = 49$, $W = 10$, and 5000 requests).

**Problem 3.5** ***The Shared Wavelength Rerouting Problem:*** *In graph $G_k(\mathcal{N}_k, \mathcal{L}_k)$ of a given wavelength $\lambda_k$, given a request $f$, the shared wavelength rerouting problem is to find link-disjoint primary and backup lightpaths such that the primary lightpath is allowed to use only free links and links that belong to reroutable lightpaths, whereas the backup lightpath can additionally use links that belong to already existing backup lightpaths whose corresponding primary lightpaths do not share a link with its own primary*

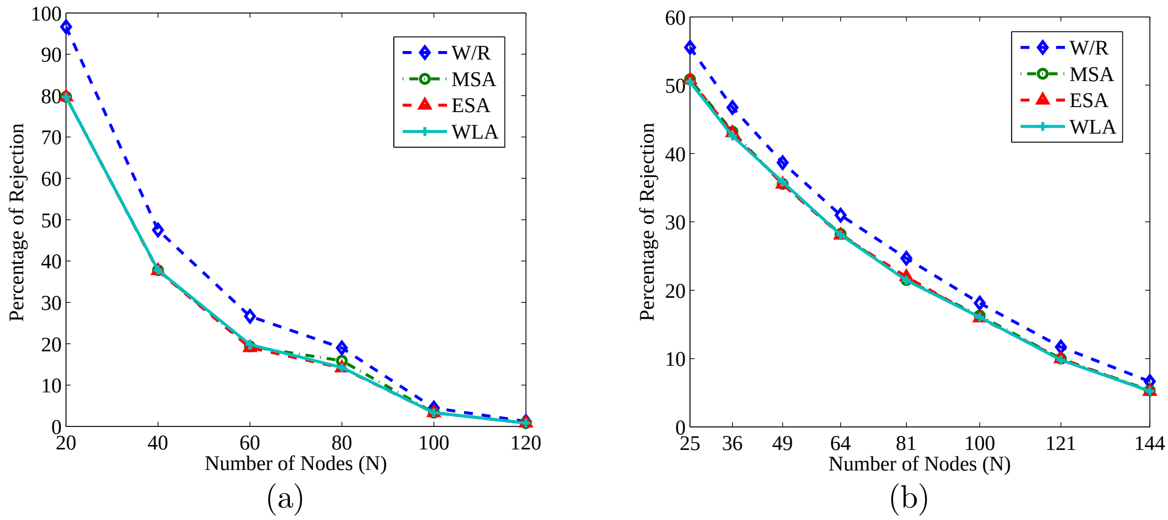



Figure 3.7: Rejection rate of $MSA$, $ESA$, $WLA$ and without rerouting for different number of nodes for random networks ($p = 0.2$, $W = 10$, $a = 100$, and 5000 requests), and (b) for lattice networks ($W = 10$, $a = 60$, and 5000 requests).

*lightpath.*

**Theorem 3.2** *The shared wavelength rerouting problem is NP-complete.*

**Proof.** The problem is in NP since the feasibility of a given solution can easily be verified.

We consider the following instance of the shared wavelength rerouting problem. This instance does not represent a worst-case scenario. In fact, it is a relatively simplified instance of the problem. In this instance, on the graph $G_k$ of wavelength $\lambda_k$, among the existing lightpaths:

- Only primary lightpaths are reroutable. Let $\mathcal{P}_k$ be the set of the reroutable primary lightpaths.
- Let $\mathcal{P}'_k$ be the set of all backup lightpaths on wavelength $\lambda_k$. For each backup lightpath in $\mathcal{P}'_k$, its corresponding primary lightpath (which can be on any wavelength), does not use the same link as a free link on $G_k$ or that of a lightpath in $\mathcal{P}_k$.

Now, let $\mathcal{L}_p \subseteq \mathcal{L}_k$ be the set of links that belong to lightpaths in $\mathcal{P}_k$, $\mathcal{L}_f \subseteq \mathcal{L}_k$ be the set of free links, and $\mathcal{L}_b \subseteq \mathcal{L}_k$ be the set of links that belong to the lightpaths in $\mathcal{P}'_k$. Color all links in $\mathcal{L}_p$ and $\mathcal{L}_f$ red, and color the links in $\mathcal{L}_b$ blue. Then, the problem is reduced to finding a pair of link-disjoint paths such that one of them uses only red-colored links, which is an NP-complete problem [89]. ■

Thus, for the shared on-line SRWA problem, we introduce a heuristic algorithm, which we call the Active Path First Rerouting ($APFR$) algorithm. This algorithm is based on an Active Path First (APF) approach, where the primary lightpath is computed first followed by the removal of its links before the backup lightpath is computed. This approach is suitable to shared survivable routing because once the primary lightpath is computed, it is easier to identify which existing backup lightpaths can share links with the new backup lightpath. However, since we are interested in minimizing the number of rerouted lightpaths, this approach fails in "trap" scenarios such as the one shown in Figure 3.8. In this example, the shortest path is $s-1-2-5-6-d$ and removing this path will disconnect the network. To avoid such scenarios, algorithm $APFR$ resorts to algorithm $MSA$ (which returns two link-disjoint paths with dedicated backup lightpath) if the APF fails to find a solution.

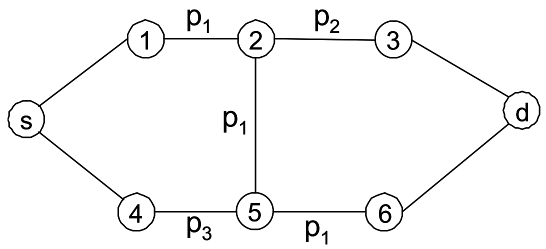



Figure 3.8: A trap scenario for an APF approach.

The major operations in $APFR$ are finding the shortest paths in Steps 1$a$ and 1$c$, finding a pair of link-disjoint paths in Step 1$e$, and identifying links that can be used by the new backup lightpath in Step 1$b$. The first two operations have a complexity of $O(L + N \log N)$. On any wavelength, let $K$ be the maximum number of backup lightpaths sharing any link. For each existing backup path, we have to check if its corresponding primary lightpath shares a link with the new primary lightpath. This costs a total of $O(KN^2)$ time and in the worst-case $O(L)$ links have backup lightpaths. Thus, the total complexity of Step 1$e$ is $O(KN^2L)$ and the total complexity of $APFR$ is $O(KWN^2L)$.

Figures 3.9 (a), (b) and (c) show comparison results of $APFR$, shared SRWA without rerouting(shared $W/R$), and $ESA$ (dedicated SRWA) in the USANET, random and lattice networks, respectively. These results show that (i) the rejection ratio is significantly reduced in shared routing, and (ii) wavelength rerouting coupled with shared routing further decreases the rejection ratio.

**Algorithm 3.5** $APFR(G, s, d)$

1. For each $G_k$, $k = 1, \ldots, W$:

   (a) Find the shortest path $P_1$ in graph $G_k$. Assign it to the primary lightpath.

   (b) Graph $G'_k$ is obtained from $G_k$ as follows,

      i. Remove all links on $P_1$.

      ii. Assign a cost of zero to all links that belong to lightpaths on $P_1$.

      iii. For all links that belong to backup lightpaths on wavelength $\lambda_k$ whose primary lightpaths do not share links with $P_1$, assign a cost of zero.

   (c) In graph $G'_k$, find the shortest path $P_2$.

   (d) If $P_2$ exists, go to the next wavelength.

   (e) Otherwise, find a pair of link-disjoint paths (i.e., with dedicated backup lightpath) in $G_k$ using $MSA$.

2. Choose the best solution (in terms of the number of rerouted lightpaths) among all wavelengths.

## 3.7 Conclusions

In WDM optical networks, where lightpaths carry a tremendous amount of data, survivability is of paramount importance. In practice, lightpath requests arrive over time and a decision on whether to accept or deny a request should be made without any knowledge of the future requests. Therefore, it is necessary to have an on-line solution scheme with good performance to deal with survivable routing and wavelength assignment (SRWA). The performance of an on-line algorithm is measured using a *competitive ratio*, which compares the blocking ratio of the on-line algorithm to that of a corresponding off-line algorithm (which knows the whole input sequence).

In this chapter, we have studied on-line SRWA and have provided constant and logarithmic competitive ratios (which are considered good competitive ratios) for special networks. For general networks, it is not possible to have algorithms with good competitive ratios. Since the competitive ratio reflects a worst-case performance, we considered lightpath rerouting, which generally improves the acceptance rate, but not the competitive ratio. To serve this purpose, we studied the Minimum Disruption Link-Disjoint Paths (MDLDP) problem, for which we provided two 2-approximation algorithms: $MSA$ and $ESA$. We have shown through simulations that these algorithms perform close to the best-known exact algorithm for MDLDP, which incurs a very high time-complexity.

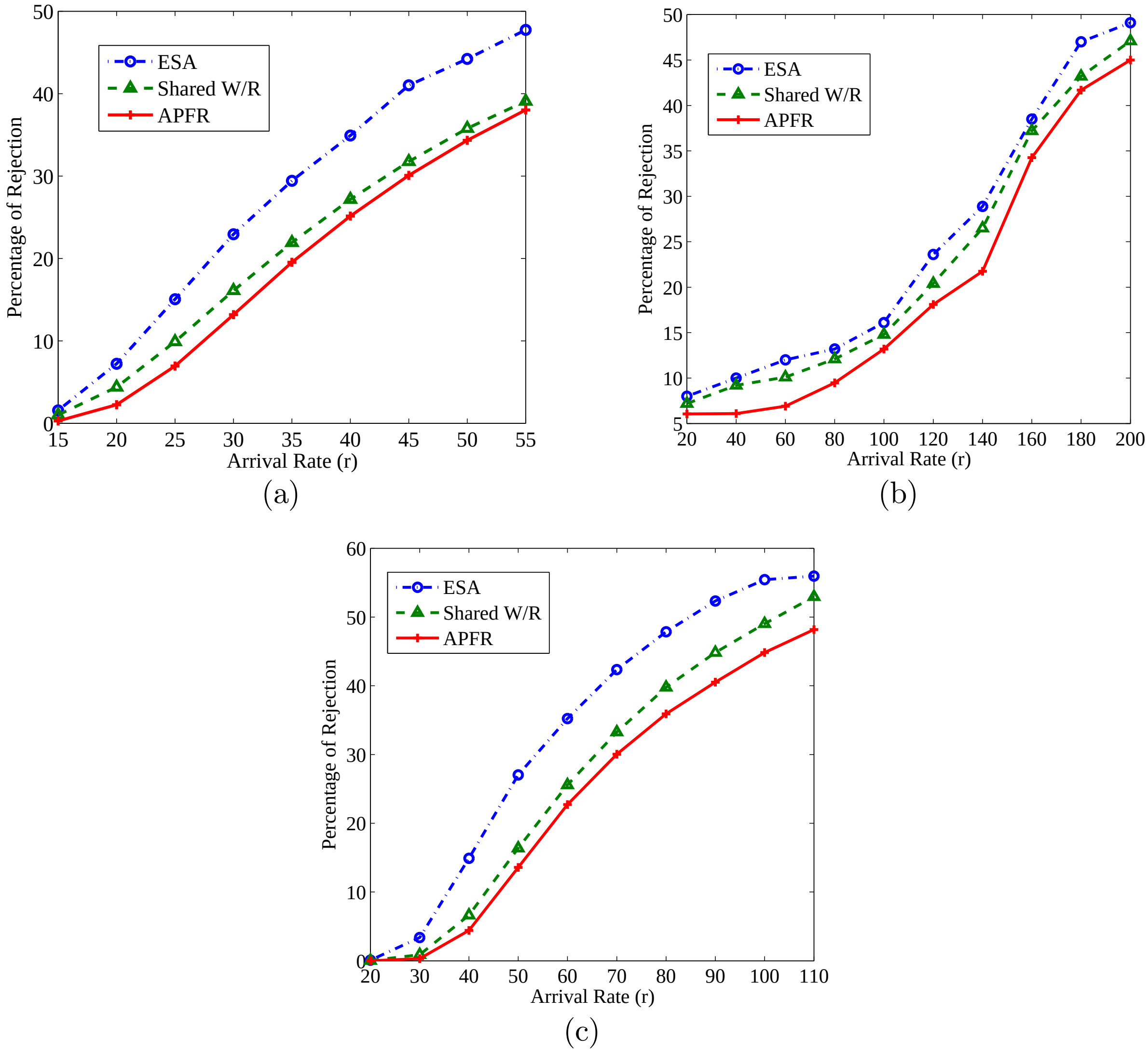



Figure 3.9: Rejection rates of $APFR$, $ESA$, and shared without rerouting for different arrival rates of (a) the USANET network ($W = 10$ and 5000 requests), (b) random networks ($N = 50$, $p = 0.2$, $W = 10$ and 5000 requests), and (c) lattice networks ($N = 49$, $W = 10$ and 5000 requests).

We subsequently applied all considered MDLDP algorithms as heuristics for infinite and finite duration on-line SRWA. For infinite duration SRWA, these algorithms performed close to the optimal off-line solution (for which we provided an ILP formulation). For finite duration SRWA, we considered Poissonian distributed input sequences with exponential holding times. In these scenarios, our algorithms performed as good as (and at times even better than) the exact algorithm of the MDLDP problem, but in a much faster time. These findings suggest that our algorithms are more suitable for dealing with the (on-line) SRWA problem.

We have also shown that unlike MDLDP, shared wavelength rerouting is NP-complete. Subsequently, we have provided an efficient heuristic algorithm for shared wavelength rerouting. Through simulations, we have shown that combined with sharing of resources among backup lightpaths, wavelength rerouting can significantly reduce the rejection ratio in on-line shared SRWA.

# Chapter 4

# Impairment-aware Path Selection

## 4.1 Introduction

In the previous chapters, we have assumed that all pair of nodes are within the impairment threshold(s), i.e., the quality of the signal does not degrade below an acceptable level. Such networks are known as transparent optical networks, and the signal is transmitted in the optical domain from the source to the destination, without any conversion to the electrical domain. However, optical networks are being widely deployed in long-haul and metro/regional networks, where lightpaths cover long distances. In such networks, if the signal is not regenerated at intermediate nodes, noise and signal distortions are accumulated along the physical path. The noise and signal distortions are known as *physical impairments* and degrade the quality of the received signal. Especially for long distances and high bit rates, the signal degradation may lead to an unacceptable bit error rate (BER).

The solution is to place regenerators at intermediate nodes. Since regenerators are costly, it is not practical to equip all nodes with regeneration capacity. Thus, in practice, only a few nodes have regeneration capacity, and such networks with sparse regeneration capacity are known as *translucent optical networks*. As far as impairment-aware routing is concerned, the two main problems in translucent optical networks are: (1) how to incorporate impairment awareness in the routing algorithms (impairment-aware path selection), and (2) how many regenerators to place inside the network and where (regenerator placement). We deal with the former in this chapter, while the latter is studied in the following chapters. In impairment-aware path selection, the segment between two nodes with regeneration capacity should have impairment value(s) that do not exceed the given threshold(s). In this chapter, we study how to incorporate impairment awareness in the RWA algorithms of translucent optical networks.

The outline of this chapter is as follows. In Section 4.2, we present a model of an optical transport network and discuss the key physical impairments that can be encountered in such a network. In Section 4.3, we briefly overview the work related to impairment-aware routing and regenerator placement. In Section 4.4, we propose impairment-aware path selection algorithms in translucent networks. We conclude in Section 4.7.

## 4.2 Impairments Model

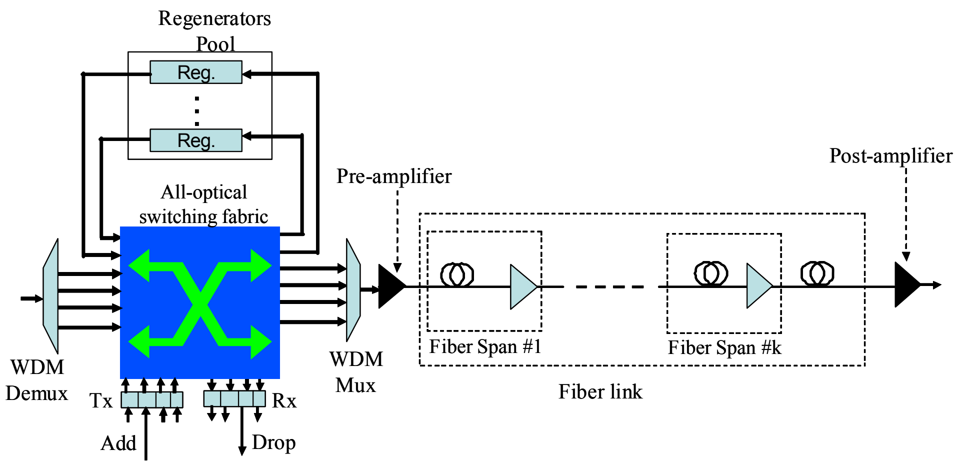



Figure 4.1: Network components that make up a translucent WDM OTN.

Figure 4.1 shows the components that make up a typical transmission system of a translucent optical transport network (OTN). In this model, a node is mainly composed of an all-optical switch, optional regenerators, transponders, multiplexers, demultiplexers, and pre- and post-amplifiers; whereas a fiber link is a WDM line system comprising of fibers and amplifiers. At a given node, transponders modulate electrical signals onto distinct wavelengths. These wavelengths are then multiplexed and the multiplexed signal is pre-amplified before being propagated through the WDM line. At the receiver end, the signal is post-amplified and de-multiplexed into individual signals. A link has a single fiber in each direction, each containing a number of wavelengths to be used by lightpaths. In Figure 4.1, only one direction is shown. We assume that there is no conversion at any of the nodes, which implies that a lightpath should use the same wavelength on all of its links.

At each node, there are add and drop ports for data to locally enter and leave the network. Each incoming signal is demultiplexed and switched inside a translucent node using an all-optical switching fabric, which can switch an optical signal from any input port to any other output port. In a translucent network, certain wavelengths may pass through the pool of regenerators so that the quality of their signals is restored through regeneration. We assume that the optical switches at the translucent nodes have enough ports to support incoming signals as well as regenerated signals. It is possible that some nodes do not have regenerators, thus providing only the service of locally adding and dropping of wavelengths. We define a *regeneration segment* of a lightpath to be a transparent segment (i.e., one or more links) between two regenerator nodes (including source and destination nodes) of the lightpath. A lightpath can be made up of multiple regeneration segments. There is no need for a lightpath to be regenerated at the source and destination nodes. After a transit signal is regenerated, its original physical features are restored. Thus, from a physical impairment point of view, the effect of physical impairments along the path followed to reach the regenerator node is completely removed. Each lightpath is assumed to require a single wavelength and each request represents a single lightpath (otherwise, each lightpath can be considered independently).

The major physical impairments can be well approximated by one or more link-based additive metric(s) and corresponding constraint(s) (for instance on length and number of spans) [84][98]. In the literature, different types of cost functions have been suggested for links and nodes to represent their physical impairments during the path-selection process. These include the distance of a link [120], a logical distance [45], a combination of distance and hopcount [23], a cost that is a function of the Four-Wave Mixing (FWM) crosstalk [71], the signal quality $Q$-factor [32] [120], an aggregated cost of monitored link information [83], and the noise variance [46]. Approaches dealing with multiple metrics explicitly have also been considered (e.g., [74], [84]). These metrics may represent measured or computed physical impairment values. Our work is independent of the impairment cost function used, and is applicable to single or multiple additive link metrics.

## 4.3 Related Work

There has been an increasing interest in dealing with physical impairments in optical networks. Most of the related work is either directed to (1) studying the problem of finding feasible paths that satisfy a given set of impairment constraints or (2) studying the optimal placement of regenerators in a network. We shall briefly discuss the work in both areas.

Azodolmolky *et al.* [12] have surveyed impairment-aware routing and wavelength assignment (RWA) algorithms. These impairment-aware algorithms commonly fall into two categories. In one category, the path and the wavelength of a lightpath are computed in the traditional way without taking into account the physical impairments, and subsequently the quality of the selected lightpath is tested against physical impairments [2], [49], [70], [78]. Then, new paths are computed if the candidate paths do not meet the physical impairment thresholds. In the second category, the physical impairment values are considered in the routing and/or wavelength assignment process [32], [46], [71], [81], [84], [92]. In these works, information pertaining to physical impairments is incorporated in finding a suitable path. However, most of them do not account for the presence of regenerator nodes during path computation. In order to achieve optimal use of regenerators, the physical impairment values of segments between regenerator nodes should be verified during the path computation process (as in our approach), and not after the path is finally computed. We provide a detailed study into the complexity of this problem and propose both exact and heuristic algorithms.

## 4.4 Impairment-aware Path Selection

In this section, we shall assume that a network is given with a scarce amount of regenerators in place and that requests arrive in an online fashion, i.e. without prior knowledge of when and between which nodes these requests are made. We define the impairment-aware routing problem as follows.

**Problem 4.1** ***The impairment-aware routing problem:*** *The physical optical network is modeled as a graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of $N$ nodes and $\mathcal{L}$ is the set of $L$ links. Associated with each fiber link $(u, v) \in \mathcal{L}$ are $m$ physical impairments $r_i(u, v)$, $i = 1, \ldots, m$. $\mathcal{N}_R \subseteq \mathcal{N}$ represents the set of $N_R$ nodes that have (spare) regeneration capacity. A request is represented by the tuple $(s, d, \vec{\Delta})$, where $s, d \in \mathcal{N}$ are the source and destination nodes of the request and $\vec{\Delta} = \{\Delta_1, \ldots, \Delta_m\}$ represents $m$ threshold values for the $m$ physical impairments. The impairment-aware routing problem is to find a route from source to destination that does not exceed any of the thresholds $\Delta_i$, $i = 1, \ldots, m$ on any of its regeneration segments.*

We illustrate this problem for $m = 1$ impairment using the example network in Figure 4.2(a) for a request $(s, d, 5)$. In this example, the shortest path from $s$ to $d$ goes via the direct link $(s, d)$, but this path violates the impairment threshold, i.e., $r(s, d) = 6 > \Delta$. The only feasible path is $s - t - d$, where $t$ is a regenerator node, because for the regeneration segments $P_{s\to t} = s - t$ and $P_{t\to d} = t - d$, it holds that $r(P_{s\to t}) = r(P_{t\to d}) = 5 \leq \Delta$.

Consider now the instance in Figure 4.2(b), where, given a request $(s, d, 5)$, there is a feasible *walk* $s{-}2{-}t{-}2{-}d$, but there is no feasible simple *path* for the impairment-aware routing problem. However, it is of interest to consider a variant of the impairment-aware routing problem where only simple paths are admitted as solutions. Such restrictions may be due to scarcity of resources (link or node capacities) or management considerations.

We first present a Polynomial-time Impairment-Aware Routing Algorithm ($PIARA$) for finding a path from source $s$ to destination $d$ subject to a single impairment threshold

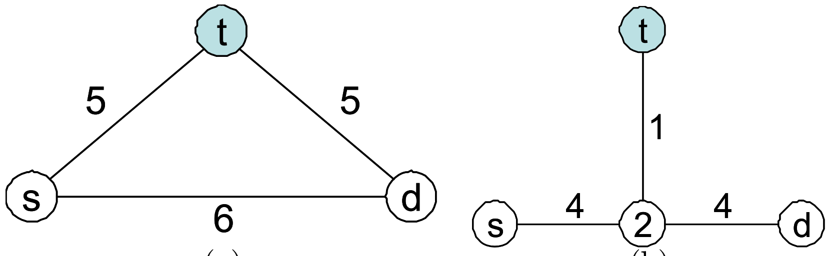



Figure 4.2: Example networks with request $(s, d, 5)$. (a) There is a feasible path through regeneration node $t$. (b) There is no feasible simple path.

$\Delta$ and for the case that loops are allowed (i.e., nodes and links may be revisited). $PIARA$ will serve as a module for our loopless impairment-aware routing algorithms.

**Algorithm 4.1** $PIARA(G, s, d, \mathcal{N}_R, \Delta)$

1. For each pair of nodes $u, v \in \mathcal{N}_R \cup \{s, d\}$, find the shortest (w.r.t. impairment) path $P^*_{u \to v}$.
2. Make a graph $G'$ consisting of nodes in $\mathcal{N}_R \cup \{s, d\}$. There is a link between nodes $u, v \in \mathcal{N}_R \cup \{s, d\}$ in $G'$ if $r(P^*_{u \to v}) \leq \Delta$.
3. Assign a cost to each link $(u, v)$ in $G'$ (e.g., a cost equal to $r(P^*_{u \to v})$).
4. Find a (shortest) path from $s$ to $d$ in $G'$ and substitute the links of the path in $G'$ with the corresponding subpaths in $G$.

By replacing the shortest path algorithm in Step 1 of $PIARA$ with a multi-constrained path algorithm (like SAMCRA [104]), we can deal with multiple impairments. However, since multi-constrained path selection is a (weakly) NP-complete problem [62], $PIARA$ will no longer be of polynomial complexity. $PIARA$ assumes that link or node capacities are not confining, even when traversed multiple times. When link or node capacities are confining we may need to find loop-free paths, which is considered in the remainder of this section. Although it is clear that the problem is NP-complete for $m > 1$ impairments, we shall demonstrate in the following that the problem is NP-complete for $m = 1$ as well.

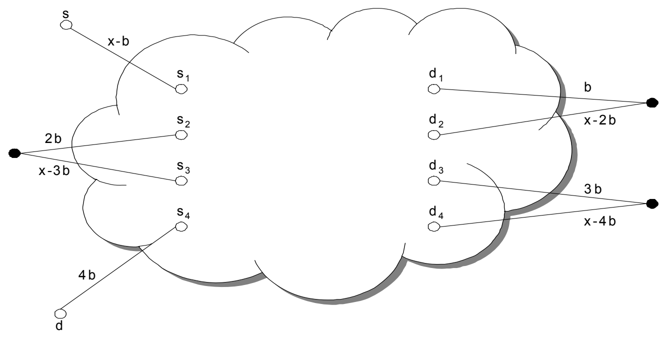



Figure 4.3: Transformation of an MLBDP instance to an impairment-aware routing instance. Black nodes are regenerator nodes.

**Theorem 4.1** *The impairment-aware loopless routing problem is strongly NP-complete*[1]*.*

To prove that the problem is strongly NP-complete, we shall use the Maximum Length-Bounded Disjoint Paths (MLBDP) problem [39], which is defined as follows.

**Problem 4.2** ***The maximum length-bounded disjoint paths problem:*** *Given an undirected graph $G$, source $s$ and destination $d$, and positive integers $b$ and $K$, does $G$ contain $K$ or more mutually node-disjoint paths from $s$ to $d$, none involving more than $b$ links?*

The MLBDP problem was proven to be NP-complete for $b \geq 5$ by Itai *et al.* in [54] and later proven to be APX-hard[2] for $b \geq 5$ by Bley in [18].

**Proof.** When we are given a path it is easy to verify whether it obeys the threshold $\Delta$ or not. The problem is therefore in NP. We shall provide a reduction to the MLBDP problem to prove strong NP-completeness.

Any instance of the MLBDP problem can be transformed in polynomial time to an impairment-aware routing instance as follows. The source node is split into $K$ source nodes $s_1, ..., s_K$ and the destination node is split into $K$ destination nodes $d_1, ..., d_K$. Each of these source (destination) nodes is connected to the same nodes as the original source (destination) node. So far all links have a weight of 1. We add a new source and connect it to $s_1$ with a link of weight $x - b$. For each pair of source nodes $(s_{2i}, s_{2i+1})$, for $i = 1, ..., \lfloor \frac{K-1}{2} \rfloor$, we add a new regenerator node and link it to $s_{2i}$ with weight $2ib$ and

[1] "Strongly NP-complete" indicates that the problem remains NP-complete even if the link weights are bounded by a polynomial in the length of the input. Unlike weakly NP-complete problems, these problems do not admit pseudo-polynomial time solutions.

[2] APX-hard problems can be approximated within some constant factor, but not every constant factor (as with polynomial-time approximation schemes), unless P=NP.

to $s_{2i+1}$ with weight $x - (2i+1)b$. For each pair of destination nodes $(d_{2i-1}, d_{2i})$, for $i = 1, ..., \lceil \frac{K-1}{2} \rceil$, we add a new regenerator node and link it to $d_{2i-1}$ with weight $(2i-1)b$ and to $d_{2i}$ with weight $x - 2ib$. The last node (either $s_K$ or $d_K$) is connected to a new destination node through a link with weight $Kb$. Figure 4.3 visualizes this construction for $K = 4$. If we choose $\Delta = x + b$ and $x > 2Kb$, then solving the impairment-aware routing problem in the new graph provides a solution to the MLBDP problem. Moreover, since $Kb \leq 2(N-1)$, we have that $\Delta = O(N)$, which on its turn means that the impairment-aware routing problem is strongly NP-complete. ■

Since good approximation schemes are unlikely to exist, as indicated by Theorem 4.1, we focus in the following sections on exact and heuristic solutions.

### 4.4.1 Problem variants

Depending on how regenerator nodes are used, associating an objective with solving the impairment-aware routing problem can lead to several problem variants. We shall first focus on the case of $m = 1$ impairment, after which we present our algorithms for the general case of $m \geq 1$ impairments.

**Variant 1:** Find the shortest (in terms of physical impairment) feasible path. Regenerators can be used at no extra cost.

**Variant 2:** Given that each used regenerator has a cost of usage that will be added to the total path length, find the shortest feasible path.

**Variant 3:** Find a feasible path that uses the fewest number of regenerators. In case of a tie, the one with shortest length is returned.

Problem variants 2 and 3 can be transformed into problem variant 1 by splitting each regenerator node in the input graph $G$ into four nodes as shown in Figure 4.4(a) for undirected networks and Figure 4.4(b) for directed networks. In these figures, the link weight $x$ equals the cost of using the given regenerator in problem variant 2, while $x = \Delta$ in problem variant 3. We will focus on solving problem variant 1.

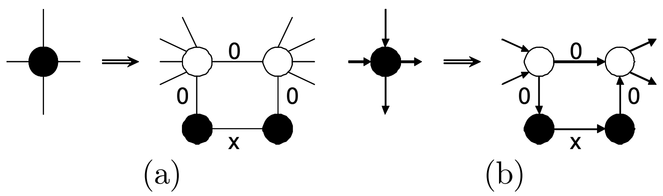



Figure 4.4: Regenerator node splitting in (a) an undirected network, and (b) a directed network. Black nodes are regenerator nodes.

In solving the impairment-aware shortest path routing problem, we have to take into account two parameters during the search process:

1. The total length $r(P)$ of a (sub)path $P$ accumulated since the source node.

2. The length $r'(P)$ since the last used regenerator node (or the source node) along a (sub)path $P$.

The fact that $r'(P)$ does not reflect an end-to-end property prevents a simple adoption of multi-parameter algorithms like SAMCRA [104]. Two search-space reducing techniques that are used in SAMCRA are the concept of *non-dominance* (or *Pareto optimality*) and the concept of *look-ahead* (or $A^*$). We will demonstrate that, while the concept of non-dominance cannot be used, we can apply the look-ahead concept with some modifications.

### Non-dominance

When solving multi-constrained routing problems, at any intermediate node, it does not make sense to consider a (sub)path that has worse weights (i.e., higher or equal in every metric) than another (sub)path. Such paths are said to be *dominated* and are discarded, thereby reducing the search space. This non-dominance technique fails in impairment-aware routing as shown in Figure 4.5. In this example, the request is $(s, d, 9)$ and $t$ is the only regenerator node. At node 3, the subpath $P_1 = s - 3$ with $r(P_1) = r'(P_1) = 8$ is dominated by the subpath $P_2 = s - 1 - 2 - 3$ with $r(P_2) = r'(P_2) = 7$. However, $P_1$ cannot be discarded since it is part of the only feasible path $s - 3 - t - 2 - 1 - d$.

Assuming non-negative link weights, the non-dominance principle prevents loops along a path. In its absence, we will have to check for loops explicitly.

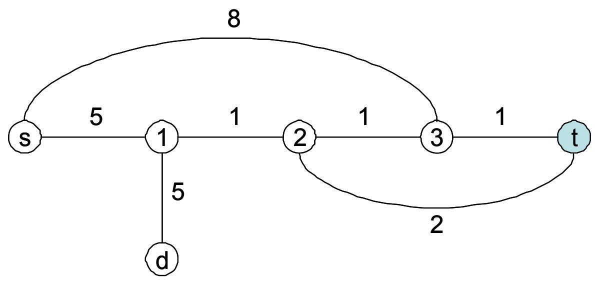



Figure 4.5: An example wherein the concept of non-dominance fails for impairment-aware routing. The request is $(s, d, 9)$ and node $t$ is a regenerator.

### Look-ahead

Look-ahead refers to finding lower bounds on the weights of the remaining subpath towards the destination in order to predict whether the current subpath will exceed any of the constraints. For multi-constrained routing, this information is built by computing for each metric, the shortest paths tree rooted at the destination node to each node in the network. For the impairment-aware routing problem, we employ two types of look-ahead values for each node, i.e., the length of the node to its nearest regenerator node and the shortest length of the node to the destination node. The former is used to calculate whether the current segment of the given subpath will lead to a length higher than $\Delta$, while the latter is used to assess whether the lower bound on the end-to-end length of the given subpath exceeds $(N_R + 1)\Delta$, since any feasible path can use a maximum of $N_R$ regenerators, where $N_R$ is the total number of regenerator nodes in the network.

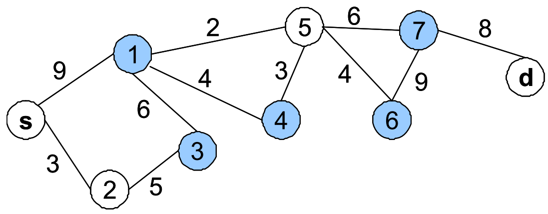
### Pruning

In a given graph, it may be necessary to check all paths between the source and destination nodes before concluding that a feasible path does not exist. This, in the worst-case, can require checking a factorial number of paths, which could take an extremely long time. In order to facilitate this process, we employ a graph pruning approach. In our simulations, we have observed that pruning drastically reduced the amount of time required in the worst-case, while slightly increasing the average time of the exact algorithm. The approach is based on the observation that two regenerator nodes that are directly connected by a link whose physical impairment is less than the threshold $\Delta$ can be merged to form a "super regeneration node". The supernode replaces the two nodes in the graph as follows:

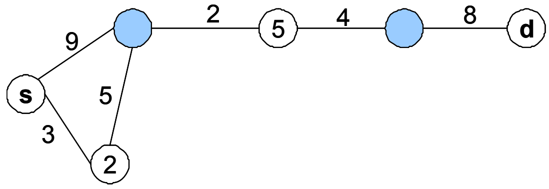
- The supernode inherits all the links of either nodes except the link between them.
- In order to maintain a simple graph, if both nodes have the same neighbor, only the link with the smaller physical impairment is inherited by the supernode.

This process can be recursively continued until all the nodes that are reachable from each other using only regenerator nodes form a supernode. It can be done by randomly choosing a regenerator node as a root node and finding the shortest "regenerator-nodes-path-tree", which contains regenerator nodes that are reachable from the root node using only regenerator nodes. We define a "cluster" as the maximal set of regenerator nodes that are reachable from each other using only regenerator nodes. Thus, a regenerator node can belong to exactly one cluster. The example in Figure 4.6(a) has two clusters: one containing nodes 1, 3 and 4, and another containing nodes 6 and 7. Figure 4.6(b) shows the pruned graph.

Note that the pruning is intended to check for the presence of a feasible path. However, the path obtained in the pruned graph is not necessarily the optimal path. In addition, if a feasible path does not exist in the pruned graph, there is no feasible path in the original graph and vice versa.




(a)




(b)

Figure 4.6: An example for graph prunning where the request is $(s, d, 10)$. The shaded nodes in the graph are the regenerator nodes. (a) The original graph with two clusters. (b) The pruned graph with two supernodes.

### 4.4.2 Exact Impairment-Aware Routing Algorithm ($EIARA$)

As in Dijkstra's algorithm, our Exact Impairment-Aware Routing Algorithm ($EIARA$) records information pertaining to subpaths leading up to intermediate nodes in the path selection process. However, for each intermediate node, unlike Dijkstra's algorithm, which stores only a single subpath, $EIARA$ maintains a list of several feasible subpaths with their corresponding impairment values and sets of regenerator nodes. We now describe in detail how $EIARA$ works.

In order to prevent, if possible, the more expensive operations in the latter parts of the code, $EIARA$ calls algorithm $PIARA$ for each impairment metric $i$ in Lines 1-12. If $PIARA$ fails to find a path for metric $i$, then $EIARA$ exits in Line 4. However, if $PIARA$ returns a path, this path is returned as a feasible solution only if it does not contain loops and satisfies all the other constraints. Hence, the flag *loop* is set to *true* if the path contains loops in Line 6, and *infeasible* is set to *true* if the path fails to satisfy any of the other constraints in Line 9.

For each node $u$, $Rnear_i[u]$ and $Rbound_i[u]$ are computed in Lines 14-18, which represent the look-ahead lengths of node $u$ to its nearest regenerator node (or destination node) and the destination node, respectively for metric $i$. The shortest path between two nodes $u$ and $v$ of metric $i$ is denoted as $P^i_{u \to v}$. In addition, $counter[u]$, which represents the number of subpaths maintained for each node $u$, is set to zero. In Line 20, the queue $Q$ that stores all the computed subpaths in the network is initialized to

**Algorithm 4.2** $EIARA(G, s, d, \mathcal{N}_R, \vec{\Delta})$

```
for i = 1, ..., m do
    P_i ← PIARA(G, s, d, 𝒩_R, Δ_i)
    if P_i = NULL then
        STOP → return no path found!
    else
        loop ← CHECK_LOOP(P_i)
        if (not(loop)) then
            for j = 1, ..., m do
                infeasible ← CHECK_CONS(P_i, Δ_j)
                if (infeasible)  then
                    goto Step 1
            STOP → return (P_i)

for each u ∈ 𝒩 do
    counter[u] ← 0
    for i = 1, ..., m do
        Rnear_i[u] ← min_{v∈𝒩_R∪{d}} {r_i(P^i_{u→v})}
        Rbound_i[u] ← r_i(P^i_{u→d})

queue Q ← ∅
counter[s] ← counter[s] + 1
INSERT(Q, s, counter[s], 0⃗)

while (Q ≠ ∅) do
    P[u, k] ← EXTRACT-MIN(Q)
    if (u = d) then
            STOP→ return P[u, k]
    else
        for each v ∈ adj[u] do
            infeasible ←BACKTRACK(P[u, k], v)
            for i = 1, ..., m do
                RL'_i ← r'_i(P[u, k]) + r_i(u, v) + Rnear_i[v]
                RL_i ← r_i(P[u, k]) + r_i(u, v) + Rbound_i[v]
                if (RL'_i > Δ_i or RL_i > (N_R + 1) ∗ Δ_i) then
                    infeasible ← 1
            if (not(infeasible)) then
                counter[v] ← counter[v] + 1
                for i = 1, ..., m do
                    r_i[v[counter[v]]] ← r_i(P[u, k]) + r_i(u, v)
                    r'_i[v[counter[v]]] ← r'_i(P[u, k]) + r_i(u, v)
                    if (v ∈ 𝒩_R) then
                        r'_i[v[counter[v]]] ← 0
                π[v[counter[v]]] ← u[k]
                INSERT(Q, v, counter[v], RL⃗)
```

an empty set. The path counter of the source node $s$ (i.e., $counter[s]$) is incremented in Line 21, and in Line 22 the subpath that contains only node $s$ is inserted into the queue with a value of 0 for all impairment metrics.

Lines 24-44 search for the solution as long as the queue $Q$ is not empty (otherwise, there is no feasible path). In Line 25, EXTRACT-MIN extracts the best subpath (e.g., the one with the smallest $\max_i\{r_i(P)\}$) in $Q$. Let the extracted subpath be the $k$-th subpath of a node $u$, which is denoted as $P[u,k]$. If node $u$ is the destination node, then subpath $P[u,k]$ is returned as the solution by concatenating the predecessor list $\pi$ in Line 27. If node $u$ is not the destination node, each node adjacent to node $u$ is considered in Lines 29-44. In Line 30, the function BACKTRACK returns *true* if adjacent node $v$ has already been encountered along this subpath, and *false* otherwise. In Line 32, $RL_i'$, which is the predicted length from the last regenerator node along the current subpath to the nearest regenerator node of node $v$ in metric $i$ is computed. In Line 33, $RL_i$, which is the predicted end-to-end length (i.e., source to destination node) in metric $i$ of the current subpath is computed. If a cycle is not detected along the current subpath, and the values of $RL_i'$ and $RL_i$ do not exceed $\Delta_i$ and $(N_R+1)\Delta_i$, respectively, the path counter of node $v$ is incremented in Line 37. The corresponding information associated with the new subpath, i.e. $r_i$ (the length of the subpath in metric $i$), $r_i'$ (the length since the last regenerator node in metric $i$), and $\pi$ (the predecessor list) are assigned in Lines 38-43. If node $v$ is a regenerator node, the length since the last regenerator node along the current subpath is set to zero in Line 42. This does not necessarily mean that this regenerator node will be used along the current subpath. Instead, after *EIARA* finds the final solution, it identifies the regenerator nodes where regeneration is absolutely necessary. This can be accomplished by only using regenerators that are farthest, but within $\Delta_i$ for each $i=1,...,m$, from the previous regenerator or source. Finally, the subpath is inserted into $Q$ in Line 44. Since *EIARA* is essentially a brute-force approach that only prunes paths from the search space (via the look-ahead concept) that are provably infeasible, *EIARA* is guaranteed to be exact.

The complexity of *EIARA* can be computed as follows (disregarding $O(1)$ operations). Lines 1-12 have a complexity of $O(mN_RN\log N + mN_RL)$ and the operations in Lines 14-18 have the same complexity. Let $k_{\max}$ be the maximum number of subpaths that are computed for any intermediate node. Then, the queue $Q$ contains at most $k_{\max}N$ subpaths. When using a Fibonacci or Relaxed Heap to structure the queue, selecting the best subpath takes at most $O(\log(k_{\max}N))$ time [30]. Since each node can be selected at most $k_{\max}$ times from the queue, the EXTRACT-MIN function in Line 25 takes at most $O(k_{\max}N\log(k_{\max}N))$ time. Constructing the path in Line 27 takes at most $O(N)$ time. The **for** loop starting in Line 29 is invoked at most $k_{\max}$ times for each side of each link in the graph, resulting in $O(k_{\max}L)$ time. The BACKTRACK function in Line 30 takes $O(N)$ time and the **for** loop in Line 31 takes $O(m)$ time. Thus, the total running time of Lines 29-44 is $O(k_{\max}N\log(k_{\max}N) + k_{\max}LN + k_{\max}Lm))$. Combining the running times of all the operations in *EIARA* results in the following computational complexity:

$$C_{EIARA} = O(mN_R N \log N + k_{\max} N \log(k_{\max} N) + k_{\max} LN)$$

### 4.4.3 Heuristics

In this section, we provide two heuristics. Our first heuristic is named $TIARA$, i.e., Tunable Impairment-Aware Routing Algorithm, and it is identical to $EIARA$ except that the maximum number of subpaths $k_{\max}$ that can be computed for any node is now bounded by a fixed $k$ that is part of the input[3]. If $k = 1$, as set in the simulations, the complexity of $TIARA$ is $O(mN_R N \log N + mN_R L + NL)$. The second heuristic is called the Loop Avoidance Heuristic $LAH$.

---

**Algorithm 4.3** $LAH(G, s, d, \mathcal{N}_R, \vec{\Delta})$

---

1. Create graph $G'(\mathcal{N}', \mathcal{L}')$ such that $\mathcal{N}' = \mathcal{N}_R \cup \{s, d\}$ and $\mathcal{L}' = \emptyset$.
2. For each impairment $i$:
   (a) For each pair of nodes $u, v \in \mathcal{N}_R \cup \{s, d\}$, $u \neq v$, and link $(u, v) \notin \mathcal{L}'$:
      i. Find the shortest path $P^i_{u \to v}$ in graph $G$ using metric $i$ as the cost.
      ii. For each impairment metric $j$, if $r_j(P^i_{u \to v}) > \Delta_j$ go to Step 2 for the next metric.
      iii. Set $P^*_{u \to v} = P^i_{u \to v}$ and add link $(u, v)$ to $\mathcal{L}'$.
   (b) Set the cost of link $(u, v)$ in $G'$ equal to the number of other such paths that $P^*_{u \to v}$ shares a segment (i.e., a link or more) with.
   (c) Find the shortest path from $s$ to $d$ in $G'$ and substitute each link $(u, v)$ in the shortest path with the corresponding subpath $P^*_{u \to v}$ in $G$ to obtain the solution.
   (d) Return the path if it exists and is loop-free, else go to Step 2 for the next metric.

---

Algorithm $LAH(G, s, d, \mathcal{N}_R, \vec{\Delta})$, as in algorithm $PIARA(G, s, d, \mathcal{N}_R, \Delta)$ given earlier, computes the shortest paths between the regenerator nodes (including $s$ and $d$) in creating graph $G'$. The difference is that $LAH$ tries to avoid loops by assigning link weights in $G'$ that reflect the "criticality" of links, which in this case relates to the number of paths associated with a link. Other measures of criticality could also be used.

[3] The multi-constrained path selection heuristic TAMCRA is analogously derived from its exact counterpart SAMCRA [104].

Since $LAH$ employs $PIARA$ for $m \geq 1$ impairments, it has $O(mN_RN \log N + mN_RL)$ time complexity.

### 4.4.4 Simulation Results

In this subsection, we tested the algorithms under a wide range of instances. Specifically, we present case-by-case comparisons of the three algorithms, where we create thousands of graphs from a particular class, and for each graph we run the algorithms for a single request. The performance metrics that we use to compare the algorithms are the *success ratio* (i.e., the ratio of requests with feasible paths to the total number of requests) and the *average time* an algorithm takes to find a feasible path (only requests for which feasible paths are found by all the three algorithms are considered).

Table 4.1: Simulation parameters for a given carrier's backbone network.

| Parameter | Value |
|---|---|
| $OSNR_{\min}$ | 20 dB |
| Average optical power | 4 dBm |
| ASE factor ($n_{sp}$) | 2.5 |
| Average span length | 75 km |
| Fiber Loss | 0.2 dB/km |
| $hvB_0$ | −58 dBm |
| Amplifier gain ($\gamma$) | 15 dB |
| Bit rate | 10 Gb/s |

We first give simulation results on a carrier's backbone network that has been used in other works [84], [119]. The network has 28 nodes and 43 links. The source and destination nodes of each request are randomly selected. Similarly, the regenerator nodes are randomly chosen. Table 4.1 shows the parameters used in this simulation, which are similar to those suggested in [84] and [97]. As in [84], we assume that ASE is the dominant physical impairment. We compare the performance of our three algorithms and a $K$-shortest path approach that has been used in [70], where at most $K$ shortest paths are computed and the path that fits first is selected. In Figure 4.7, it can be seen that the $K$-shortest paths approach performs poorly for small values of $K$. The success ratio of the $K$-shortest paths approach becomes comparable to those of our algorithms only for very large values of $K$ (e.g., $K = 100$), making it less suitable for impairment-aware path selection in translucent optical networks.

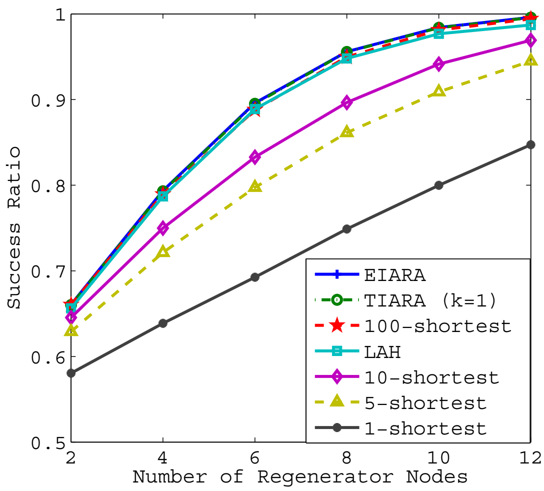



Figure 4.7: Case-by-case comparison of the success ratios of our three algorithms and a $K$-shortest path approach.

We have also performed extensive simulations on random and lattice networks. The link weights were uniformly distributed within a scaled and normalized range of $(0, 1]$. In practice, we expect a more positive correlation between links individually and also between the $m$ impairment values of a link, which would simplify the problem. Also, without loss of generality, we assume that the threshold values are equal, i.e., $\Delta_i = \Delta$ for $i = 1, ..., m$. For each request, the source and destination nodes are randomly assigned. In addition, the regenerator nodes are placed randomly in such a way that there are no two adjacent nodes with regeneration capacity. If two regenerator nodes are directly connected by a link with feasible link weights, then they can be merged to form a "super regenerator node" without affecting the feasibility of any path. Such preprocessing of the graph could reduce the network size and make the problem relatively easier, which is avoided in our set-up.

We have observed that the results for the lattice networks have been worse than for the random graphs, when the network size and link-weight distribution are the same. The reason is that a larger expected hopcount in lattice networks increases the probability that the impairment thresholds will be violated. We shall only present the results for the lattice networks. We have also observed that $EIARA$ is generally fast when a feasible path exists, but in some cases it can take a long time to decide that no solution exists. As noted earlier, we only present the computation times for requests that are accepted by *all* the three algorithms.

### Different impairment thresholds

Figure 4.8 shows the success ratios of the three algorithms for lattice networks of $N = 49$ by varying the impairment threshold $\Delta$. It can be seen that an increase in the impairment threshold leads to a relaxation of the problem, and consequently increases the probability that a feasible path exists (and is found). In fact, after $\Delta = 2.4$, both $EIARA$ and $TIARA$ find feasible paths for all requests.

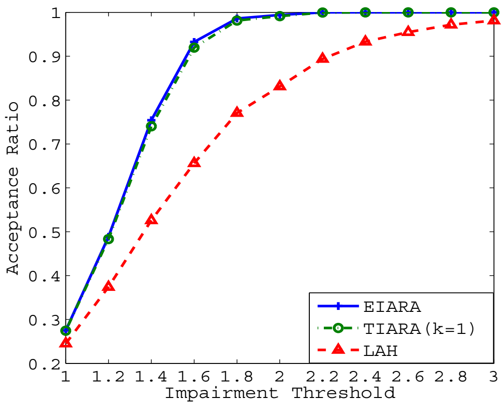
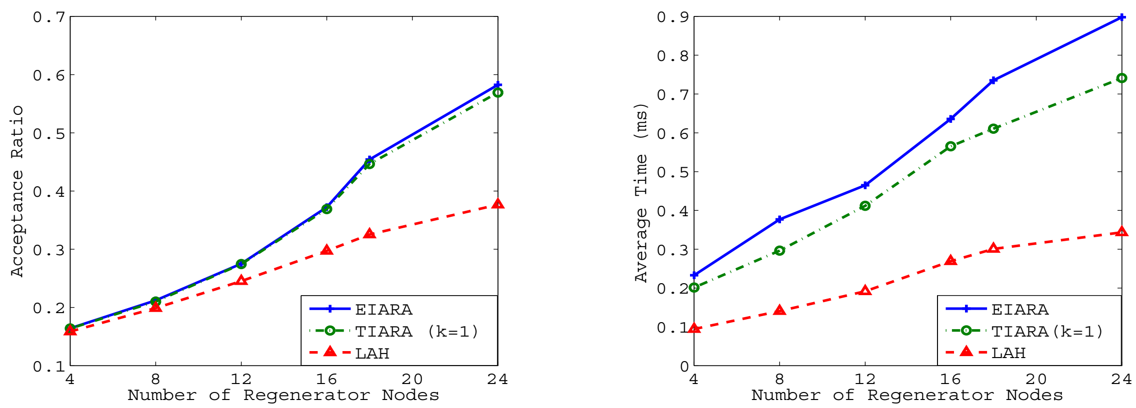



Figure 4.8: Case-by-case comparison of the success ratio for lattice networks ($N = 49$) and different impairment thresholds ($\Delta$). (Number of networks = 10000, $N_R = 12$, and $m = 2$).

### Different number of regenerator nodes

Figures 4.9 (a) and (b) show the success ratios and average times of the three algorithms for lattice networks of $N = 49$ by varying the number of regenerator nodes $N_R$. Since increasing the number of regenerator nodes improves the possibility of finding a feasible path, the success ratio grows fast with the number of regenerator nodes. The average time of finding a feasible path also increases with the number of regenerator nodes. This is because, in $EIARA$ and $TIARA$, the number of feasible subpaths that are considered in the path computation process increases, while in $LAH$, the number of links in $\mathcal{L}'$ increases.

In general, the simulations show that although $LAH$ is somewhat faster than $TIARA$, $TIARA$ is also fast and always outperforms (often considerably) $LAH$. Moreover, the quality of $TIARA$'s solutions are quite close to the exact solutions of $EIARA$. Hence, $TIARA$ is our preferred choice, offering close-to-optimal performance within a reasonable computational complexity.




(a) (b)

Figure 4.9: Case-by-case comparison of the success ratio for lattice networks ($N = 49$) (a) for different number of regenerator nodes (R). (Number of networks =10000, $\Delta = 1$, and $m = 2$), and (b) different number of regenerator nodes ($N_R$) (Number of networks = 10000, $\Delta = 1$, and $m = 2$).

## 4.5 Convertor/Regenerator Minimization Problem

Like some related work, in the previous sections, we have assumed that regenerators do not perform wavelength conversion. However, regenerator devices are generally capable of performing both regeneration and wavelength conversion simultaneously. In this section, we extend the impairment-aware path selection problem by considering how to find a feasible lightpath for a given request while minimizing the number of required wavelength conversions as well as regenerations. For ease of presentation, we focus on the case of $m = 1$ impairment, but our algorithms apply to the general case of $m \geq 1$ impairments by replacing the shortest path computations with multi-constrained path computations (using for instance SAMCRA [104] instead of Dijkstra's algorithm).

**Problem 4.3** ***The conversion/regeneration minimization problem (CRMP):*** *In addition to the input of the impairment-aware path selection problem, associated with each fiber link $(u, v) \in \mathcal{L}$ is a set of at most $W$ unused wavelengths $\mathcal{W}(u, v)$. The conversion/regeneration minimization problem (CRMP) is to find a feasible path from $s$ to $d$ that requires the minimum number of wavelength conversions/regenerations, such that the impairment threshold as well as the wavelength continuity constraint is satisfied between converter/regenerator nodes.*

Since the impairment-aware path selection problem (i.e., without the presence of wavelength converters) is strongly NP-complete for general topologies, the CRMP problem is also strongly NP-complete. However, in this section we show that for specific topologies, the CRMP problem is polynomially solvable.

### 4.5.1 Line Topology

The line topology is suitable for fixed path routing techniques, which may be necessary for such cases as when paths are pre-selected according to specific policies other than resource optimization, or when a group of feasible paths are pre-computed and the problem is to choose the best among them, or when the path between the source and destination nodes is fixed in a given topology (e.g., a tree topology). As can be seen in Figure 4.10, if conversion and regeneration are considered independently, an optimal solution may not be found. For example, if regeneration is performed at node $b$ in Figure 4.10(a) and at node $a$ in Figure 4.10(b), then wavelength conversion is needed at node $a$ for Figure 4.10(a) and at node $b$ for Figure 4.10(b). In the optimal solution, both wavelength conversion and regeneration can be done only at node $a$ for the former, and only at node $b$ for the latter.

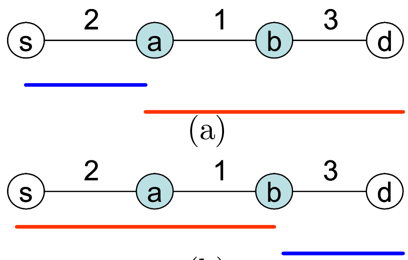



Figure 4.10: A line topology with only two available wavelengths and $\Delta = 5$. Nodes $a$ and $b$ are regenerator/converter nodes.

We now provide an exact algorithm, which we call the Line Conversion/Regeneration Minimization ($LCRM$) algorithm, for solving the $CRMP$ problem in a line topology.

The $LCRM$ algorithm constructs a graph $G'$ by connecting only reachable converter/regenerator nodes (including the source and destination nodes). Any two converter/regenerator nodes are said to be *reachable* if the distance between them is less than the impairment threshold, and there is at least one continuous wavelength between them. The internal nodes of the shortest path between the source and destination nodes in the new graph will be the conversion/regeneration nodes for the request. The main operations in $LCRM$ are the construction of $G'$ and finding the shortest path in $G'$. The former takes $O(N^2)$ time, while the latter takes $O(L' + N_R \log N_R)$ time. Hence, $LCRM$ has $O(N^2)$ complexity.

**Theorem 4.2** *The $LCRM$ algorithm optimally solves the $CRMP$ problem in a line topology.*

**Algorithm 4.4** $LCRM(G, \Delta)$

1. Create a new graph $G'(\mathcal{N}', \mathcal{L}')$, where $\mathcal{N}' = \mathcal{N}_R \cup \{s, d\}$ and $\mathcal{L}' = \{\}$.
2. $\mathcal{L}'$ is obtained in such a way that for each pair of nodes $u, v \in \mathcal{N}'$, there is a link between them they are *reachable*. Let $\mathcal{W}'(u, v)$ represent the set of wavelengths that are available to all links between $u$ and $v$.
3. Assign a cost of 1 to each link in $G'$.
4. Find the shortest path $P'_{s-d}$ from $s$ to $d$ in $G'$. For each link $(u, v)$ in $P'_{s-d}$, assign a wavelength randomly chosen from $\mathcal{W}'(u, v)$.
5. Return $P'_{s-d}$ (and the total number of converters/regenerators is one less than its length), if it exists. Otherwise, return no solution.

**Proof.** The path returned by $LCRM$ is feasible since any pair of consecutive converter/regenerator nodes (including $s$ and $d$) are reachable from each other. Thus, what remains to show is that the number of conversions/regenerations is optimal. We provide proof by contradiction. Let us assume that it is possible to establish a feasible path between $s$ and $d$ with less number of conversions/regenerations. Hence, in graph $G'$, there are links between any two consecutive converter/regenerator nodes (including $s$ and $d$) of this solution. This, in turn, implies that $P'_{s \to d}$ is not the shortest path in $G'$, which is a contradiction. ■

### 4.5.2 Directed Acyclic Graphs

The $LCRM$ algorithm can be extended to optimally solve the CRMP problem in directed acyclic graphs (DAGs). We call this algorithm DAG Conversion/Regeneration Minimization ($DCRM$).

Unlike the line topology, in a DAG there can be several paths between a pair of converter/regenerator nodes on different wavelengths. Hence, in Step 1, we construct a separate graph for each wavelength. Then, a new graph $G'$ is created in Step 2, and a link is added between a pair of converter/regenerator nodes in Step 3 if they are reachable in any of the wavelength graphs constructed in Step 1. Creating the wavelength graphs in Step 1 has a total of $O(WL)$ complexity. Then, constructing the graph $G'$ in Step 3 requires a total of $O(WN_R^2(L + N \log N))$ time and finding the shortest path in this graph takes $O(L' + N_R \log N_R)$ time. Hence, the complexity of the $DCRM$ algorithm is $O(WN_R^2(L + N \log N))$.

**Theorem 4.3** *The $DCRM$ algorithm optimally solves the CRMP problem in a DAG.*

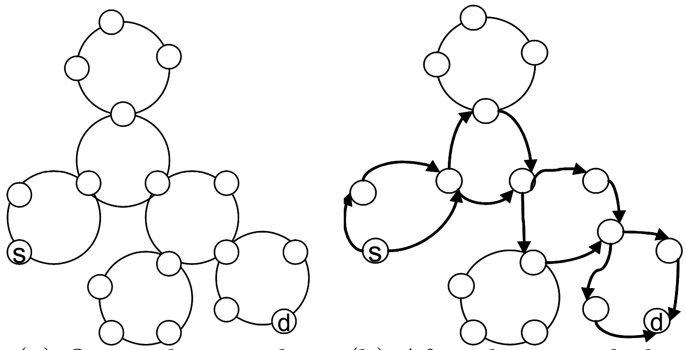
**Algorithm 4.5** $DCRM(G, \Delta)$

1. For each $\lambda = 1, \ldots, W$, construct graph $G_\lambda(\mathcal{N}, \mathcal{L}_\lambda)$, where $\mathcal{L}_\lambda = \{(u, v) \in \mathcal{L} | \lambda \in \mathcal{W}(u, v)\}$.
2. Create a new graph $G'(\mathcal{N}', \mathcal{L}')$, where $\mathcal{N}' = \mathcal{N}_R \cup \{s, d\}$ and $\mathcal{L}' = \{\}$.
3. For each pair of nodes $u, v \in \mathcal{N}'$:
   (a) For $\lambda = 1, ...W$,
      i. If the distance between $u$ and $v$ in $G_\lambda$ is less than $\Delta$,
         A. Add link $(u, v)$ to $\mathcal{L}'$ and assign wavelength $\lambda$ to it.
         B. Go to Step 3, until all pairs of nodes in $\mathcal{N}'$ are considered.
      ii. Else, go to Step $3a$ until all wavelengths are considered.
4. Assign a cost of 1 to each link in $G'$. Find the shortest path $P'_{s \to d}$ from $s$ to $d$ in $G'$.
5. If $P'_{s \to d}$ exists, obtain the solution $P_{s \to d}$ as follows:
   (a) For each link $(u, v)$ in $P'_{s \to d}$:
      i. Replace it with the corresponding path in $G_\lambda$, where $\lambda$ is the wavelength assigned to link $(u, v)$ in $G'$.
      ii. Assign wavelength $\lambda$ to each link in the corresponding path in $G_\lambda$.
6. Else, return no solution.

**Proof.** The proof for $LCRM$ can be reused to show that the optimal number of conversions/regenerations is equal to one less than the length of path $P'_{s \to d}$ obtained in Step 5, if it exists. What remains to show is that $P_{s \to d}$, which is the path constructed from $P'_{s \to d}$, is a simple path, i.e., it does not contain loops. However, if $P_{s \to d}$ contains a loop, then there is a cycle in graph $G$, which is not possible since graph $G$ is a DAG. ■

### 4.5.3 Ring Structures

In an undirected ring topology, we have two possible paths, one in the clockwise direction and the other in the anti-clockwise direction. Without loss of generality, we can transform the ring to a DAG. This can be done by directing all the links in the network from the source node to the destination node. This approach of directing all links from the source to the destination before applying the $DCRM$ algorithm can also be extended to the tree-of-rings topology as shown in Figure 4.11 or other simple structures like star topologies. It suffices to direct and use only links that belong to rings/lines that are along the path from $s$ to $d$ in the underlying tree structure, which is obtained by contracting rings to nodes.




(a) Original network. (b) After directing links.

Figure 4.11: A transformation in a tree-of-rings topology.

## 4.6 Survivable Impairment-aware Path Selection

**Problem 4.4** ***The survivable impairment-aware path selection problem:*** *Given the input of the impairment-aware path selection problem, the survivable impairment-aware path selection problem is to find a pair of link-disjoint paths whose regenerator segments do not exceed the threshold* $\Delta$*.*

Since the impairment-aware routing problem (without survivability) is strongly NP-complete, the survivable impairment-aware routing problem is also strongly NP-complete. We now show how the exact algorithm for the impairment-aware routing problem can be modified to handle the survivable impairment-aware problem. We first create a new graph, which represents two duplicates of the input graph connected by a *directed* link as shown in Figure 4.12. The following is an outline of the exact algorithm for the survivable impairment-aware path selection problem, which we call $ESIRA$.

### Outline of the $ESIRA$ algorithm

1. Create a new graph $G''(\mathcal{N}'', \mathcal{L}'')$ such that $\mathcal{N}'' = \mathcal{N} \cup \mathcal{N}'$, where $\mathcal{N}' = \{u' | u \in \mathcal{N}\}$; and $\mathcal{L}'' = \mathcal{L} \cup \mathcal{L}' \cup \{(d, s')\}$, where $\mathcal{L}' = \{(u', v') | (u, v) \in \mathcal{L}\}$.

2. Let the set of regenerators in the new graph be $\mathcal{N}''_R = \mathcal{N}_R \cup \mathcal{N}'_R \cup \{d, s'\}$.

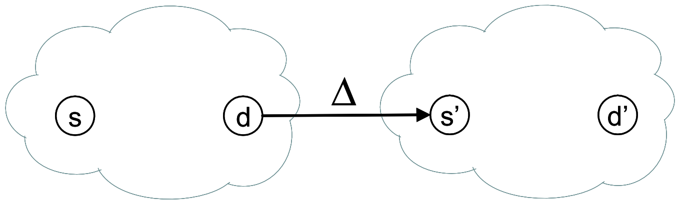



Figure 4.12: A new graph is created from the original graph before using a modified $EIARA$ for solving the survivable impairment-aware routing problem.

3. Assign a cost of $\Delta$ to link $(d, s')$.
4. Use a modified $EIARA$ algorithm to find a feasible path from $s$ to $d'$, where checking for loops is modified as follows:
   (a) Let $a$ be the current node and $b$ be its adjacent node that is being considered.
   (b) If $a, b \in \mathcal{N}$
      i. Check if node $b$ is encountered before.
   (c) If $a, b \in \mathcal{N}'$. Let $u' = a$ and $v' = b$.
      i. Check if node $v'$ is encountered before.
      ii. Check if link $(u, v)$ or $(v, u)$ is encountered before.

$ESIRA$, just like $EIARA$, is a brute-force approach that discards only unfeasible paths from the search list. In addition, Step 4$c$ prevents the backup path from sharing a link with the primary path. A heuristic $TSIRA$ can be derived from $ESIRA$, the same way $TIARA$ is derived from $EIARA$.

## 4.7 Conclusions

In optical networks, physical impairments, such as noise and signal distortions, degrade the quality of the signal. These impairments become more severe with distance and bit rate, unless the signal is regenerated timely. Since regenerators are costly, they are generally sparsely deployed, and such networks are called translucent networks. In this chapter, we have studied the problem of selecting a path that meets one or more impairment constraints.

We have shown that the impairment-aware path selection problem is NP-complete, even for the case of a single impairment. Subsequently, we have provided an exact algorithm $EIARA$ and heuristic algorithms. Through simulations we have demonstrated that our heuristic $TIARA$, which is derived from $EIARA$, is computationally efficient and offers close-to-optimal solutions. In addition, we have shown that both $EIARA$ and $TIARA$ can be easily extended to solve the survivable impairment-aware path selection problem.

We have also considered the case when regenerators can also be interchangeably used as wavelength converters. In this scenario, the problem will be to find a feasible path with the minimum number of converters/regenerators, i.e., the converter/regenerator minimization problem. We have provided exact algorithms to solve the problem in line topology and directed acyclic graphs (DAGs).

# Chapter 5

# Regenerator Placement

## 5.1 Introduction

In Chapter 4, we considered on-line requests arriving to a network with regenerators in place. In this and the following chapters, we shall consider the off-line case, where all the source and destination nodes are known in advance. In practice, signals are regenerated per wavelength (and not per fiber), with optoelectronic regenerators. The main costs in deploying optoelectronic regenerators are equipment cost (CAPEX) and power consumption (OPEX), which both are directly proportional to the total number of regenerators in the network. Since regenerators are costly, it is desirable to minimize the number of regenerators in the network. Hence, our focus in this chapter is the *regenerator placement problem* for unprotected lightpaths, where, given a set of requests, our objective is to allocate a feasible lightpath for each request while primarily minimizing the total number of regenerators needed (the survivable regenerator placement problem is studied in the next chapter). This approach differs from related work (specified in Section 5.3), where the objective is to minimize the total number of *regenerator nodes*. A regenerator node is a node that contains at least one regenerator. We argue that all nodes in a network are already regularly maintained, and minimizing the number of regenerator nodes due to maintenance costs is therefore of less importance than minimizing the number of regenerators. We define a *regeneration segment* of a lightpath to be a transparent (non-regenerated) segment (i.e., one or more links) between two consecutive regenerator nodes (plus source and destination) of the lightpath. The source and destination nodes are not considered as regenerator nodes of a lightpath. Each request is assumed to represent a single lightpath request (otherwise, each lightpath of a request is considered separately) and a lightpath has a capacity requirement of a single wavelength. A regenerator is usually capable of converting wavelengths. In absence of wavelength conversion, a lightpath has to use the same wavelength in all links along its path.

We consider regenerator placement under two scenarios:

(1) There is no limitation on the number of available wavelengths. By considering this scenario, we can analyze the complexity of the regenerator placement problem independently from the NP-hard Routing and Wavelength Assignment (RWA) problem. This scenario appears when enough fibers are already laid out to accommodate all the requests, and the main cost is associated with the regeneration capacity in the network.

(2) The number of used wavelengths needs to be limited. Since regenerators can also be used for converting wavelengths, we consider the problem of minimizing the total number of regenerators/converters required to find a feasible RWA for each request and a given number of wavelengths.

## 5.2 Regenerator Placement Context

Rai *et al.* [84] argued that most impairments can be modeled by (additive) link metrics. In Chapter 4, we have considered impairment-aware path selection for multiple additive impairments. In this chapter, our focus is on the placement of regenerators for a single impairment metric. This metric may be distance, which plays a key role in determining the quality of a signal, or it may represent the worst impairment metric among all the metrics on a link. We note that, similarly to Chapter 4, most of our algorithmic solutions can be extended to work on multiple additive metrics, by replacing single metric (shortest) path computations with multi-constrained path computations, which admit Fully Polynomial-Time Approximation Schemes (FPTAS).

Two types of regeneration/conversion schemes exist: full and single (i.e., per wavelength) regeneration/conversion. In full regeneration/conversion, a node with regeneration capability can regenerate/convert any set of incoming wavelengths to any set of outgoing wavelengths. In single regeneration/conversion, one regenerator/converter device (e.g., optical transponder) is required for each wavelength that needs to be regenerated/converted [26][29][53][69]. Obviously, full regeneration/conversion is more flexible and hence better suited for requests that arrive on-line, in an unpredictable manner. In such scenarios, minimizing the number of regenerators/converters is equivalent to minimizing the total number of regenerator/converter nodes, which is an NP-hard problem [25]. Unfortunately, full regeneration/conversion technology has not matured yet. Even though a few technologies have been developed that allow for simultaneous regeneration/conversion of several wavelengths, optoelectronic devices, which offer single wavelength regeneration/conversion, remain most practical and reliable [22][92]. Assuming full regeneration/conversion for such optoelectronic devices leads to over-provisioning at regenerator nodes. Moreover, as proved in Chapter 4, finding paths on-demand is strongly NP-hard. From a computational perspective, it is therefore cost effective to precompute paths off-line. In this case, oblivious routing techniques could be used to give proven performance bounds for any traffic matrix. With single regeneration/conversion, the number of regenerators/converters is not necessarily coupled to the number of regenerator/converter nodes. In this chapter, we consider single regeneration/conversion, where we optimize on regenerators. We also consider minimizing the number of regenerator nodes as a secondary objective, thereby further reducing the OPEX. When minimization of regenerators nodes is the sole objective, related work presented in the following section can be used to place regenerators at specific nodes, and subsequently the algorithms in Chapter 4 could be used to find feasible paths. In the other cases, routing and regenerator placement can be solved together, as proposed in this chapter.

## 5.3 Related Work

Most work on regenerator placement assumes full regeneration, e.g., [25][33][90][117] and references therein. Chen *et al.* [25] have shown that the regenerator placement problem (with no restriction on the number of wavelengths) under full regeneration is NP-hard, and have provided heuristic algorithms. Flammini *et al.* [33] have considered different variants of the same problem with the assumption that all links have the same cost. Also these variants are NP-hard.

Similarly, some studies assume full wavelength conversion while solving the NP-hard converter placement problem. Jia *et al.* [57] have proposed an approximation algorithm for the problem of placing wavelength converters at certain nodes such that the number of required wavelengths does not exceed the maximum link load (i.e., number of lightpaths on a link). Wan *et al.* [106] have provided exact polynomial algorithms for the same problem in trees, tree-connected rings and trees of rings. Andrews and Zhang [7] have studied the same problem but under single wavelength conversion. They have shown that the problem is NP-hard both in rings and stars.

Our key contributions in this chapter are outlined as follows. In Section 5.5, we show that regeneration placement for a single impairment is polynomially solvable when there is no restriction on the number of wavelengths (for multiple impairments, it is NP-hard). The problem becomes NP-hard when minimizing the number of regenerator nodes is a secondary objective, even for a single impairment. For this NP-hard problem, we develop an algorithm, $MURP$, that is exact in finding the minimum number of regenerators, while simulations show it is able to allocate the regenerators at a small (near-minimum) number of nodes. In Section 5.6, we impose a restriction on the number of wavelengths to be used. Since the restricted problem is hard to approximate in general networks, we propose a heuristic algorithm that closely matches the exact solution in a simulated backbone network. Moreover, we provide approximation bounds for line and ring topologies. We conclude in Section 5.7.

## 5.4 The Regenerator Placement Problem

In this section, we study the regenerator placement problem **(RPP)** when there is no restriction on the number of wavelengths.

**Problem 5.1** ***The Regenerator Placement Problem (RPP):*** *The physical optical network is modeled as a graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of $N$ nodes and $\mathcal{L}$ is the set of $L$ links. Associated with each fiber link $(u, v) \in \mathcal{L}$ are $m$ physical impairments $r_i(u, v)$, $i = 1, \ldots, m$. Given are a set of requests with request $j$ represented by the tuple $(s_j, d_j, \vec{\Delta})$, where $s_j, d_j \in \mathcal{N}$ are the source and destination nodes of request $j$ and $\vec{\Delta} = \{\Delta_1, \ldots, \Delta_m\}$ represents $m$ threshold values for the $m$ physical impairments. The regenerator placement problem is to minimize the total number of regenerators such that each request is assigned a simple path that does not exceed the respective thresholds on any of its regeneration segments.*

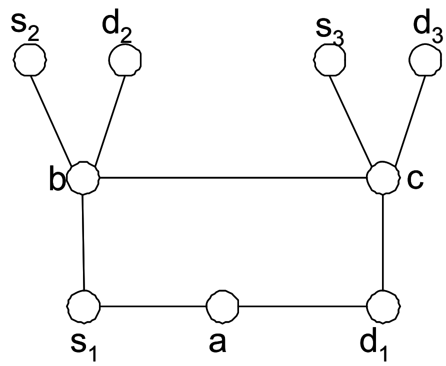



Figure 5.1: An example network where minimizing the number of regenerator nodes does not result in the minimum number of regenerators.

We note that the above problem, namely minimizing the total number of regenerators, is different than that of minimizing the number of regenerator nodes in a network. The latter is shown to be NP-hard [33] for the case $m = 1$. The difference between the two problems is illustrated in the example depicted in Figure 5.1 for $m = 1$. In this example, there are three requests $(s_1, d_1, \Delta)$, $(s_2, d_2, \Delta)$ and $(s_3, d_3, \Delta)$, and each link has a cost that is equal to the impairment threshold $\Delta$. If the objective is to minimize the number of regenerator nodes, the regenerators are placed at nodes $b$ and $c$. The total number of regenerators needed in this case is four, i.e., one for $(s_2, d_2, \Delta)$ and one for $(s_1, d_1, \Delta)$ at node $b$, and one for $(s_3, d_3, \Delta)$ and one for $(s_1, d_1, \Delta)$ at node $c$. However, when the objective is minimizing the total number of regenerators, we need only three regenerators, i.e., one for $(s_1, d_1, \Delta)$ at node $a$, one for $(s_2, d_2, \Delta)$ at node $b$, and one for $(s_3, d_3, \Delta)$ at node $c$. Since the total regeneration cost in a network depends mainly on the number of regenerator nodes needed, our approach leads to a cheaper solution than that of minimizing the number of regenerator nodes in the network.

Since at a given time, a regenerator can only be used by a single lightpath, the allocation of regenerators to each request can be considered independently. Thus, the regenerator placement problem is reduced to solving a regenerator placement problem for individual requests. For each request, the problem is then to assign a feasible simple path by allocating the necessary regenerators, while minimizing the total number of regenerators needed. We call this problem the Single Request Regenerator Placement (SRRP) problem. For $m > 1$, the SRRP problem is obviously NP-complete (it includes the multi-constrained path problem), however we show that the problem is polynomially solvable for $m = 1$. We first provide an exact algorithm called $ESRRP$, and subsequently prove its exactness. In Step 1 of $ESSRP$, a multi-constrained path is computed between each pair of nodes in the network. For $m > 1$, an exact multi-constrained path algorithm, such as SAMCRA [104], can be employed. However, for $m = 1$, the $ESRRP$ algorithm is polynomial, since Step 1 is basically finding the shortest paths between all pairs of nodes.

---

**Algorithm 5.1** $ESRRP(G, s, d, \Delta)$

---

1. For each pair of node, $u, v \in \mathcal{N}$, find a (shortest) path $\{P^*_{u\to v}\}$ such that $r_i(P^*_{u\to v}) \leq \Delta_i$ for $i = 1, \ldots, m$.
2. Make a graph $G^{'}(\mathcal{N}, \mathcal{L}^{'})$, where $\mathcal{L}^{'} = \{(u, v) \mid r_i(P^*_{u\to v}) \leq \Delta_i \quad \forall i = 1, \ldots, m\}$ and assign a cost of 1 to each link.
3. Find the shortest path $P^{'}_{s\to d}$ from $s$ to $d$ in $G^{'}$. Let $\ell(P^{'}_{s\to d})$ represent the cost/hopcount of path $P^{'}_{s\to d}$.
4. Substitute the links of $P^{'}_{s\to d}$ with the corresponding subpaths $P^*_{u\to v}$ in $G$ to obtain $P_{s\to d}$.
5. Remove all loops in $P_{s\to d}$ to obtain the optimal solution.

---

The notations used in the following theorems are given in the $ESRRP$ algorithm.

**Theorem 5.1** *The minimum number of regenerators required by any path from $s$ to $d$ is at least $R = \ell(P^{'}_{s\to d}) - 1$.*

**Proof.** We prove by contradiction. Assume that there is a path $P^{''}_{s\to d}$ in $G$ from $s$ to $d$ that needs only $k < R$ regenerators and let $\{n_1, n_2, \ldots, n_k\}$ be the nodes (in that order) in $P^{''}_{s\to d}$ where the regenerators are placed. Hence, there should be links

$(s, n_1), (n_1, n_2), \ldots, (n_{k-1}, n_k), (n_k, d)$ in graph $G'$. Then, there is a path from $s$ to $d$ through these nodes with a hopcount of $\ell(P''_{s\to d}) = k + 1 < \ell(P'_{s\to d})$ in graph $G'$, which in turn implies $P'_{s\to d}$ is not the shortest path in $G'$. ■

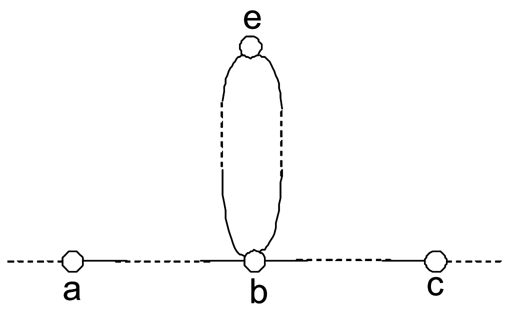



Figure 5.2: A loop along the path at node $b$.

**Corollary 5.1** *A loopless path that uses only $R$ regenerators can be obtained from $P'_{s\to d}$.*

**Proof.** Consider a loop in path $P'_{s\to d}$ starting and ending at a given node $b$ as shown in Figure 5.2. Let $a$ and $c$ represent the nearest regenerator nodes at both sides of node $b$ outside the loop. Thus, $r_i(P_{a-b}) \leq \Delta_i$ and $r_i(P_{b-c}) \leq \Delta_i$ for $i = 1, \ldots, m$. If $r_i(P_{a-b}) + r_i(P_{b-c}) \leq \Delta_i$ for $i = 1, \ldots, m$, then removing the loop will not affect the feasibility of the path and the total number of regenerators required. But if $r_i(P_{a-b}) + r_i(P_{b-c}) > \Delta_i$ for any physical impairment $i$, then there is a regenerator in the loop (the regenerator can be either at $b$ or at any other node in the loop). Then, placing the regenerator at node $b$ and removing the loop will not affect the feasibility of the path and the total number of regenerators required. This process is repeated until all the loops in $P'_{s\to d}$ are removed in order to obtain a loopless path of $R$ regenerators. ■

For $m > 1$, the complexity of $ESRRP$ is entirely dominated by the multi-constrained path computation in Step 1. For example, if SAMCRA is used, the complexity of $ESRRP$ will be $O(k_{\max} N \log(k_{\max} N) + k_{\max}^2 mL)$, where $k_{\max}$ is the maximum number of paths that are computed for any node. On the other hand, for $m = 1$, the complexity of $ESRRP$ is determined by three major operations in the algorithm: constructing graph $G'$, finding the shortest path between $s$ and $d$ in $G'$, and removing the loops of the path in $G$. The construction of graph $G'$ involves finding the shortest paths from each node to all other nodes. This can be implemented with $O(N^2 \log N + NL)$ complexity using Johnson's algorithm [58]. Finding the shortest path in $G'$ can be implemented using a Breadth First Search (BFS) with $O(L')$ complexity, which is $O(N^2)$ for dense graphs. The path obtained in $G'$ has $O(N)$ hopcount in the worst-case and at each node there can be a loop with $O(N)$ hopcount in $G$, thus the total complexity of removing loops is $O(N^2)$. Therefore, the total complexity of $ESRRP$ for $m = 1$ is $O(N^2 \log N + NL)$.

## 5.5 The Minimized Regenerator Placement

Regenerators are active components that require maintenance, a task that is facilitated by grouping them in a smaller number of physical locations. Therefore, besides minimizing the total number of regenerators, it is additionally desirable to minimize the number of regenerator nodes. Henceforth, we focus on the case of $m = 1$ impairment for ease of presentation, but as discussed in the previous chapter our algorithms apply to the general case of $m \geq 1$ impairments by replacing the shortest path computations with multi-constrained path computations (using for instance SAMCRA [104] instead of Dijkstra's algorithm).

**Minimized RPP:** In addition to **RPP**, a secondary objective is to minimize the number of nodes where the regenerators are placed (i.e., regenerator nodes).

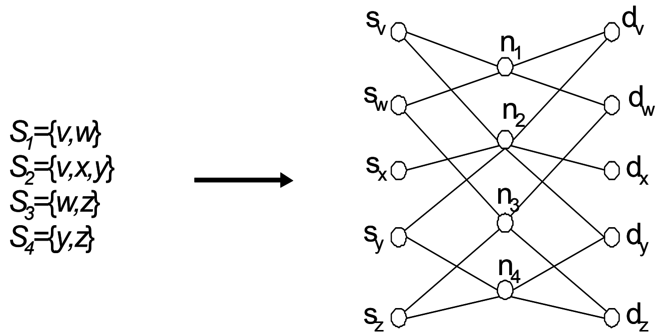



Figure 5.3: Reduction of an instance of the **Minimum Set Cover** problem to an instance of the **Minimized RPP** problem.

**Theorem 5.2** ***Minimized RPP*** *is NP-hard.*

We prove that the problem is NP-hard using a reduction from the NP-hard **Minimum Set Cover Problem** [39]: Given a finite set $\mathcal{U}$ and a family $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$ of subsets of $\mathcal{U}$, a subfamily $\mathcal{C} \subseteq \mathcal{S}$ is a *cover* if $\bigcup_{\mathcal{S}_i \in \mathcal{C}} \mathcal{S}_i = \mathcal{U}$. The problem is to find the cover $\mathcal{C}$ whose cardinality $|\mathcal{C}|$ is minimized.

**Proof.** We show how to construct in polynomial time an instance of Minimized RPP from an instance of the Minimum Set Cover problem. Given an instance of the **Minimum Set Cover** problem, for each $u \in \mathcal{U}$, create two nodes $s_u$ and $d_u$, and for each $\mathcal{S}_i \in \mathcal{S}$, create a node $n_i$. If $u \in \mathcal{S}_i$, connect node $n_i$ to nodes $s_u$ and $d_u$. Set the cost of each link in the new graph equal to the impairment threshold $\Delta$. Figure 5.3 shows an example of the reduction process, where $\mathcal{U} = \{v, w, x, y, z\}$.

The input for the corresponding instance of **Minimized RPP** in the new graph is: for each $u \in \mathcal{U}$, there is a request $(s_u, d_u)$. Observe that, for each request $(s_u, d_u)$, the minimum number of required regenerators is one. Also, for each request $(s_u, d_u)$ and $n_i$ such that $u \in \mathcal{S}_i$, there is exactly one candidate (i.e., requiring one regenerator) path $s_u - n_i - d_u$. Hence, finding the minimum number of regenerator nodes $\{n_i\}$ so that each request has a feasible path is equivalent to finding the minimum number of subsets $\{\mathcal{S}_i\}$ so that $\mathcal{U}$ is covered. ■

### 5.5.1 Line Topology

We provide an exact line regenerator placement algorithm, termed *ELRP*, for a line topology. A line topology corresponds to several cases of interest, e.g., when applying for fixed-path routing techniques and the pre-computed paths fit on a single line, or when considering only a single lightpath. In addition, a line topology is common in long-haul networks.

Let $f = 1, \ldots, F$ represent the lightpaths, and for each lightpath $f$, let $s_f$ and $d_f$ be its source and destination nodes. Without loss of generality (w.l.o.g.), we assume that each lightpath requires at least one regeneration, and for each lightpath $f$, $s_f$ is to the left of $d_f$ along the line topology.

Step 1 takes $O(FN)$ time and returns the optimal number of regenerators $R_f$ for each lightpath. In Step 2, we pick the lightpath with the left-most source node, and then place a regenerator for it at the farthest directly reachable node $u$ to the right in Step 3. In Step 4, we determine all lightpaths that can possibly place regenerators at node $u$, without incrementing their respective optimal number of regenerators. These are lightpaths whose remaining segment, i.e., $(u, d_k)$ for lightpath $k$, requires only $R_k - 1$ regenerators. After placing a regenerator at node $u$ for each of these lightpaths, we remove their respective segments up to node $u$, and decrement their required number of regenerators. Step 4 takes a total of $O(FN)$ time. In Step 5, we return to Step 2 until all lightpaths are handled. Since at each iteration, we move to the right by at least one link, algorithm *ELRP* has a total complexity of $O(FN^2)$.

**Theorem 5.3** *Algorithm ELRP is an exact algorithm for solving* ***Minimized RPP*** *in a line topology.*

**Proof.** The number of regenerators required by each lightpath is optimal (and could also have been found by *ESRRP*). Thus, we only have to show that the number of regenerator nodes is optimal as well.

Let $n_1, n_2, \ldots, n_k$ be the regenerator nodes (arranged from left to right on the line) returned by *ELRP*. Assume by contradiction that $w_1, w_2, \ldots, w_m$, where $m = k - 1$

**Algorithm 5.2** *ELRP*$(G, s, d, \Delta)$

1. For each lightpath $f$: We begin from $s_f$, with $R_f$ set to 0. Then, we move to the farthest directly reachable node $u$ (i.e., within $\Delta$) to the right of $s_f$ along lightpath $f$, and increment $R_f$. Next, we move to the farthest node to the right of $u$ and increment $R_f$, and so on, until we reach $d_f$, where we stop without incrementing $R_f$.

2. Let $j$ be the lightpath with the left-most source node. Let node $u$ be the farthest node to the right that is within the impairment threshold $\Delta$ from $s_j$.

3. Place a regenerator for lightpath $j$ at node $u$.
   a. If $R_j = 1$, remove lightpath $j$.
   b. Else, remove segment $(s_j, u)$, $R_j = R_j - 1$, $s_j = u$.

4. For each lightpath $k$ whose source node $s_k$ belongs to segment $(s_j, u)$,
   a. If its segment $(u, d_k)$ requires only $R_k - 1$ regenerators, then place a regenerator for it at node $u$.
   i. If $R_k = 1$, remove lightpath $k$.
   ii. Else, remove segment $(s_k, u)$, $R_k = R_k - 1$, $s_k = u$.

5. Go to Step 2, if there remain lightpaths. Else, exit.

---

is the optimal solution (also arranged from left to right). From algorithm *ELRP*, for each $2 \leq l \leq k$, $n_l$ is the farthest directly reachable node from $n_{l-1}$ or is not directly reachable from $n_{l-1}$, since lightpaths may not overlap. Thus, for each $l \leq m$, each node $w_l$ must be to the left of node $n_l$ or is the same as node $n_l$. Finally, there is at least one lightpath, say $j$, whose destination node, $d_j$, is to the right of node $n_k$, and whose source node is at or to the left of node $n_{k-1}$. Since $w_m$ is at best the same as node $n_{k-1}$, and segment $(n_{k-1}, d_j)$ is not feasible, this contradicts our assumption. ■

### 5.5.2 General Topologies

For general topologies, we provide a greedy heuristic algorithm, termed *MURP* for solving **Minimized RPP**. *MURP* modifies the extraction process of the shortest path algorithm (e.g., Dijkstra's algorithm) that is used to find shortest paths between the source and destination nodes in graph $G'$ of *ESRRP*. In *MURP*, for each node, we assign a number that counts the regenerators needed at that node for the lightpaths allocated so far. During the extraction process, if two nodes have the same cost, the node with the higher regenerator count is extracted. This is based on the assumption that a node where a large number of requests place their regenerators in graph $G'$ is likely to remain a regenerator node once the (non-simple) paths are transformed to their equivalent simple paths in graph $G$. Since, as $ESRRP$, $MURP$ finds the shortest path in $G^{'}$ for each request, the total number of regenerators needed is optimal, and only the number of regenerator nodes may not be optimal. $MURP$ could be generalized, by considering a weighted combination of the number of regenerators (link costs) and regenerator nodes (node costs). Assigning weight $N$ to regenerators and weight 1 to regenerator nodes, would be equivalent to our tie-breaking formulation in Step 3a.

**Algorithm 5.3** $MURP(G, \Delta)$

1. Construct graph $G^{'}(\mathcal{N}, \mathcal{L}^{'})$ such that for each pair of nodes $u, v \in \mathcal{N}$, link $(u, v) \in \mathcal{L}^{'}$ if node $v$ is directly reachable from node $u$.
2. Assign a cost of 1 to each link and a regenerator count of 0 to each node.
3. For each request $f$,
   (a) Find the shortest path from $s_f$ to $d_f$ in $G^{'}$. The shortest path algorithm should always extract the node with minimum cost and if there is a tie, choose the node with a higher regenerator count.
   (b) For each intermediate node in the shortest path, increment its regenerator count by one.
4. For each request $f$,
   (a) Substitute the links in its shortest path in $G^{'}$ with corresponding subpaths in $G$.
   (b) Remove all loops of the path in $G$ and place the regenerators either at the origin of the removed loop or at nodes common to two consecutive subpaths in Step $4a$.

### 5.5.3 Simulation Results

We first provide simulation results comparing $MURP$ and $ESRRP$ with the exact solution (obtained via an ILP formulation given in Appendix A) on an NSFNET backbone network with 14 nodes and 21 links [90] (Figure 5.4). To simulate a wide range of values, the impairment values of the links are randomly generated in the (scaled) range $(0, 1]$ with impairment threshold $\Delta = 1$, and the source and destination are randomly selected. Figure 5.5 shows the comparison results of an average of 10 iterations for different numbers of requests. $MURP$ performs much better than $ESRRP$ and close to the exact solution in terms of the number of regenerator nodes required in the given network.

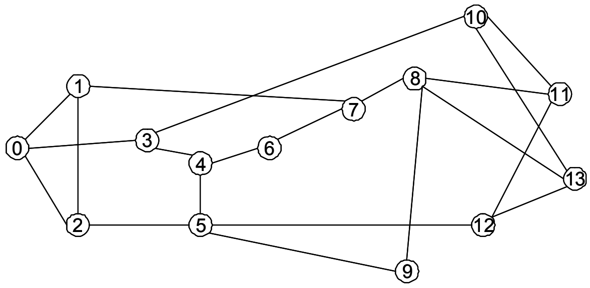



Figure 5.4: NSFNET network.

Since the ILP formulation does not scale well, we provide comparison results of only $MURP$ and $ESRRP$ in larger networks. Figures 5.6 (a) and (b) show simulation results for Erdös-Rényi random (with link density $p = 0.2$) and lattice networks, respectively. As stated earlier, in terms of the number of regenerators needed, $MURP$ obtains optimal results as $ESRRP$. Thus, the comparison is in terms of the number of regenerator nodes. The results represent an average of 10 iterations, with 1000 requests in each iteration. For each request, the source and destination nodes are randomly generated. The impairment threshold is $\Delta = 1$. For the lattice networks, the impairment values are uniformly distributed within the range $(0, 1]$. However, for the random networks, we have observed that no or very few regenerators are needed when the impairments are in this range. Therefore, we set the range for the link costs to $[0.5, 1]$. These results show that $MURP$ requires significantly fewer regenerator nodes than $ESRRP$.

## 5.6 Wavelength-constrained Regenerator Placement

In the previous sections, we have assumed that there is no restriction on the number of wavelengths. However, in general, the number of wavelengths is a constraint. Depending on how the constraint on the number of wavelengths is specified, there can be two major categories of the regenerator/converter placement problem. One is when the number of available wavelengths is given as an input. In this case, the problem is to maximize the number of lightpaths that are accepted and to minimize the number of regenerators/converters required to achieve this, which is a multi-objective problem. The other is to set the number of regenerators/converters to the maximum load among all links in the network. This, in turn, can be of two types depending on how the routing is computed. In the first type, routing is part of the problem and the objective is to find a routing with the minimum maximum link load among all the routings for the

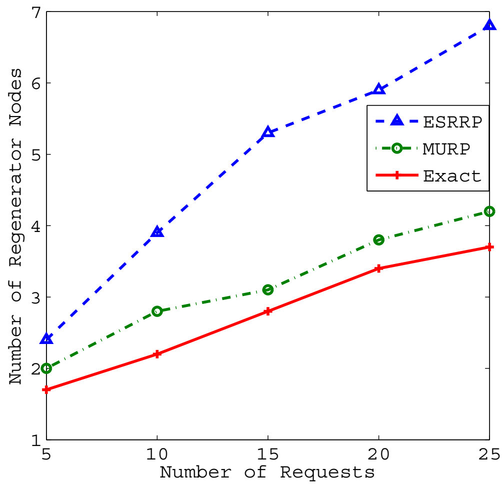


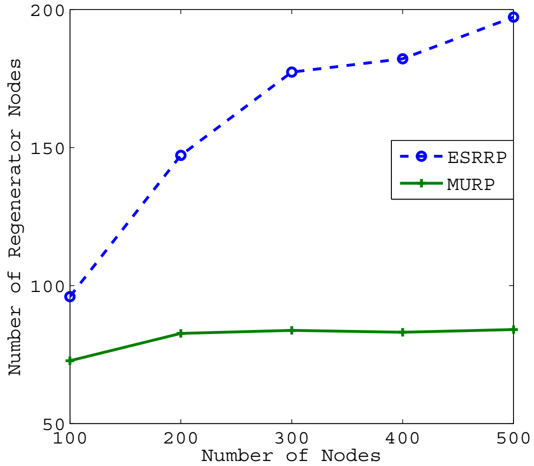

Figure 5.5: Comparison of the number of regenerator nodes of $MURP$ and $ESRRP$ to the exact solution in the NSFNET network.

given set of requests while the required number of regenerators/converters is minimized. In the second type, the routing is predetermined and the problem is to minimize the number of regenerators/converters required so that the number of wavelengths is equal to the maximum link load of the given routing. We consider this last scenario.

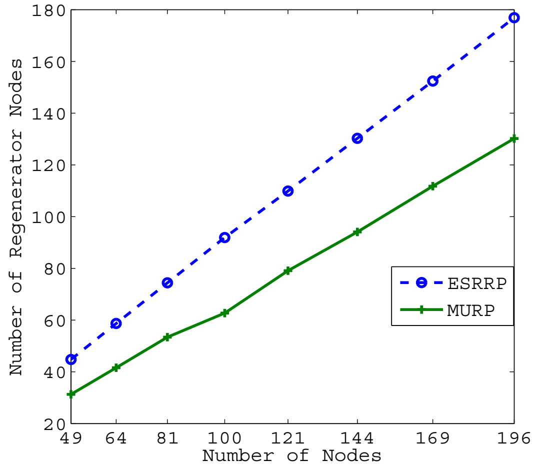
**Wavelength-Constrained Regenerator Placement (WCRP) problem**: Given a network $G(\mathcal{N}, \mathcal{L})$, impairment values for each link $r(u, v)$, a threshold $\Delta$, and a routing with a maximum link load $W$, the problem is to minimize the required number of regenerators/converters such that each lightpath is feasible and the number of wavelengths needed at any link is at most $W$.

**WCRP** consists of two subproblems: the regenerator placement subproblem and the converter placement subproblem. Solving the two subproblems independently generally leads to a heuristic approach since regeneration and conversion capacities are coupled, i.e., when a wavelength is regenerated, it could be converted as well. However, in **WCRP**, since the path taken by each lightpath is predetermined, the regenerator placement subproblem (i.e., **RPP**) can optimally be solved using algorithm $E\_LINE$ for each lightpath. Unfortunately, the converter placement subproblem is NP-hard even for simple topologies such as rings and stars, and is hard to approximate within a constant factor in general topologies [7].

We now give a simple greedy algorithm, termed $GRP$, for solving **WCRP.** We provide an approximation bound for it in ring topologies, and show its performance in the NSFNET network using simulations. The algorithm sequentially assigns regenerators/converters to each lightpath. The order in which the lightpaths are handled



(a)



(b)

Figure 5.6: Comparison of the number of regenerator nodes of $MURP$ and $ESRRP$ in (a) random networks, and (b) lattice networks.

can be random, or in descending order of their hopcounts (since longer lightpaths are more likely to cause conflicts), or in any other order suitable to the network topology. While assigning regenerators/converters to a lightpath, any two nodes of the lightpath are not reachable if the length between them exceeds the threshold (Step 1) or if there is no continuous wavelength between them (Step 2). The former is taken care of using regeneration, while the latter is solved with conversion in Step 4. Since Steps 1 to 5 take a total of $O(NW)$ time, $GRP$ has a complexity of $O(FNW)$.

$GRP$ can be specifically tailored for a ring topology, which is widely used in metro networks, so that an approximation bound is obtained. This is achieved by specifying how the lightpaths are arranged (which is inspired by the arc coloring algorithm given in [103]) and how the wavelengths are assigned. A lightpath is said to *pass through* a node if the node is an internal node of the lightpath. Let $u$ be a node with the minimum number of lightpaths passing through it, and it must be the end point of at least one lightpath. The lightpaths are arranged as follows: Let $i_1$ be a lightpath whose end point is node $u$. Depending on whether node $u$ is the counterclockwise or clockwise end of lightpath $i_1$, we move round and round the ring clockwise or counterclockwise, respectively, indexing lightpaths. W.l.o.g., let us assume that we are moving clockwise. Let $i_2$ be the first unindexed lightpath whose counterclockwise end comes after the clockwise end of $i_1$, $i_3$ be the first unindexed lightpath whose counterclockwise end comes after the clockwise end of $i_2$, and so on until all lightpaths are indexed.

While moving around the ring, let $k$ be the number of times we return to node $u$. Let $F_k$ be the set of lightpaths indexed during the $k^{th}$ round. In Step 2 of algorithm $GRP$, we put the restriction that all lightpaths in $F_k$, except the one that crosses node

**Algorithm 5.4** *GRP*$(G, \Delta, W)$

For each lightpath $f$:

1. Begin at the source node $s_f$. Find node $u$ that is the farthest node within $\Delta$ from $s_f$ along lightpath $f$.
2. Find the farthest node $v$ that can be reached using the same wavelength from node $s_f$ along lightpath $f$. Let $\lambda$ be the wavelength, if there are such multiple wavelengths, choose the one with the lowest index.
3. If node $u$ is closer to $s_f$ than node $v$ is, then let $n = u$. Otherwise, let $n = v$.
4. Place a regenerator/converter at node $n$ for lightpath $f$, unless node $n$ is $d_f$. Remove wavelength $\lambda$ on all links of $G$ between $s_f$ and $n$ along lightpath $f$.
5. Unless node $n$ is $d_f$, replace $s_f$ by $n$ and go to Step 1.

---

$u$, if any, use only wavelengths from 1 to $k$. In round $k$, it may not be possible for the lightpath that passes through node $u$ to use only wavelengths $1, \ldots, k$, since its segment after node $u$ may overlap with previous lightpaths using these wavelengths. Hence, a converter is placed at node $u$ and this lightpath is partitioned into two so that the segment before node $u$ is in $F_k$, while the segment after $u$ is in $F_{k+1}$. In each round, the maximum load decreases by 1, thus, at most $W$ wavelengths are required at each link.

Figure 5.7 shows an example with 6 lightpaths and none of the lightpaths require regeneration. In this example, let node 1 be the starting node since a minimum number of lightpaths pass through it and at least one lightpath has an ending at it. Going clockwise, the lightpaths are arranged as shown in Figure 5.7. In the first round, lightpath $i_1$, $i_2$ and the segment of $i_3$ between nodes 7 and 1 are assigned wavelength 1. Then, a converter is placed at node 1. In round 2, the segment of lightpath $i_3$ between nodes 1 and 2, lightpath $i_4$ and lightpath $i_5$ are assigned wavelength 2. Finally, lightpath $i_6$ is assigned wavelength 3.

Denoting by $\mu$, the total number of lightpaths passing through node $u$, then the following theorem establishes our bound.

**Theorem 5.4** *In a ring topology, GRP requires at most* $OPT{+}\mu$ *regenerators/converters, where* $OPT$ *is the optimal number of regenerators/converters required.*

**Proof.** For each lightpath indexed in round $k$, except the one that crosses node $u$, if any, at least wavelength $k$ is available from end to end. Hence, reachability is affected only by the impairment threshold. An optimal number of regenerators are, thus, placed for these lightpaths. Any possible non-optimality is incurred due to the converter placed

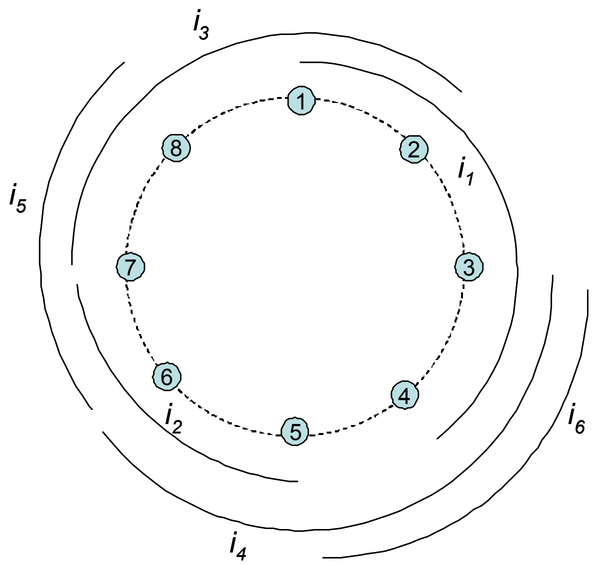



Figure 5.7: An example of a ring topology with 6 lightpaths. Assume that none of the lightpaths require regeneration.

at node $u$ for the lightpath that passes through node $u$. Since there are a total of $\mu$ lightpaths passing through node $u$, in the worst-case, $\mu$ converters are unnecessarily placed at node $u$. ■

The same bound can be achieved by first placing a converter at node $u$ for each lightpath passing through it, and then cutting the ring at node $u$. In the resulting line topology, the **WCRP** problem can be optimally solved using $E\_LINE$, since no conversion is needed for a line topology.

### Simulation Results

Figure 5.8 shows simulation results on the NSFNET network comparing $GRP$ with exact solutions (via an ILP in Appendix B) and a worst-case bound, which occurs when each intermediate node of each lightpath requires regeneration/conversion. Since the ILP does not scale well for a large number of requests, the number of requests is from 5 up to 25. The source and destination of the requests are randomly selected, and the shortest path is assigned for each request. The impairment values are randomly distributed within $(0, 1]$ and $\Delta = 1$. The results represent an average of 10 iterations. Although it is for a small number of requests, these results show that $GRP$ performs significantly better than the worst-case bound, and very close to the exact solution in the given network.

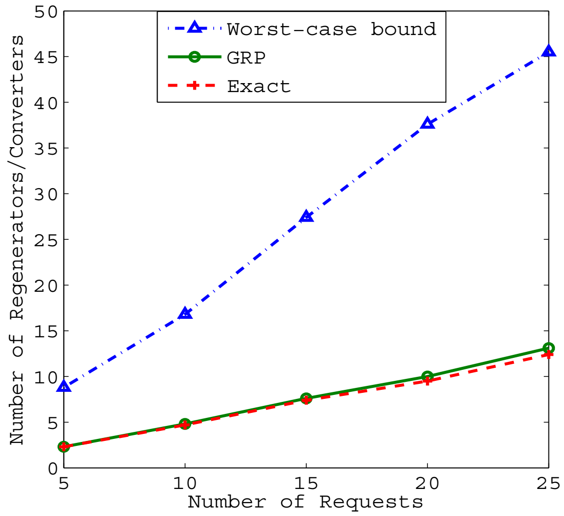



Figure 5.8: Comparison results of *GRP* with the exact solution and worst-case bound in the NSFNET network.

## 5.7 Conclusions

In this chapter, we have studied the regenerator placement problem, where the main objective is minimizing the total number of regenerators required for a given set of requests. We considered this problem with and without a restriction on the number of wavelengths.

When there is no restriction on the number of wavelengths, the problem of minimizing the number of regenerators is polynomially solvable for $m = 1$ physical impairments, and we have established an exact algorithm ($ESRRP$) for it. However, for $m > 1$, we have shown that it is NP-hard. In addition, when minimization of the number of regenerator nodes is a secondary objective, the problem is computationally intractable even for $m = 1$. Accordingly, we proposed a greedy heuristic algorithm that is optimal in terms of minimizing the number of regenerators, and performs close to optimal in terms of minimizing the number of regenerator nodes.

When the number of wavelengths is restricted, the problem becomes NP-hard for $m \geq 1$. Thus, we provided an efficient greedy algorithm for general topologies, which has a proven performance bound in line and ring topologies.

# Chapter 6

# Survivable Regenerator Placement

## 6.1 Introduction

In Chapter 5, we studied the regenerator placement problem for unprotected lightpaths. In this chapter, we turn to *survivable regenerator placement* problem, where given a set of requests, each request is assigned a primary lightpath and a link-disjoint backup lightpath such that the total number of regenerators is minimized. We study the survivable regenerator placement under two survivability schemes: dedicated and shared protection. In dedicated protection, the backup lightpaths do not share resources. In shared protection, backup lightpaths can share resources as long as their respective primary lightpaths do not share links. For both schemes, which we prove to be NP-hard, we provide heuristic, approximation, and exact algorithms, which are compared through simulations. For typical networks, our heuristic algorithms are shown to find near-optimal solutions.

In Section 6.2, we study survivable regenerator placement under dedicated and shared survivability schemes. For both schemes, which we prove to be NP-hard, we provide heuristic, approximation, and exact algorithms, which are compared through simulations. For typical networks, our heuristic algorithms are shown to find near-optimal solutions. We conclude in Section 6.3.

## 6.2 Problem Definition

In this section, we study dedicated and shared survivable regenerator placement. In dedicated survivable regenerator placement, backup lightpaths of different requests are not allowed to share regenerators. In shared survivable regenerator placement, backup lightpaths may share regenerators as long as their respective primary lightpaths do not share links.

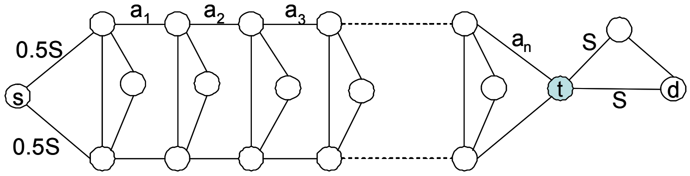



Figure 6.1: An example that shows how to construct an instance of the regeneration problem from an instance of the partition problem.

### 6.2.1 Dedicated Survivable Regenerator Placement

**Dedicated Survivable Regenerator Placement Problem:** Given the input of **RPP** of Chapter 5, the problem is to place regenerators and to find for each request link-disjoint paths that satisfy the impairment threshold, such that the total number of regenerators needed by all the requests is minimized.

Since we are considering dedicated protection, the different requests do not share regenerators. Thus, each request can be considered individually as follows.

**Single Request Survivable Regenerator Placement (SRSRP) Problem:** Given an undirected graph $G(\mathcal{N}, \mathcal{L})$, impairment values $r(u,v)$, a threshold $\Delta$, and a request represented by $(s,d)$, the problem is to find a pair of link-disjoint paths for the request, and to place regenerators along these paths, while minimizing the total number of regenerators needed for both paths.

There can be two variants of the problem: (i) *Dedicated-Dedicated*: there is no sharing of regenerators between the two link-disjoint paths and (ii) *Dedicated-Shared*: if the backup path is used only after the failure of the primary path, regenerators on nodes that belong to both the primary and backup paths can be shared. Unlike the **SRRP** problem in Chapter 5, we will show that the **SRSRP** problem is NP-hard.

**Theorem 6.1** *Both* dedicated-dedicated *and* dedicated-shared *variants of the SRSRP problem are NP-hard.*

Our proof makes use of the NP-hard **partition problem** [39]: Given a set of weights $a_i \in A$, $a_i \geq 0$ for $i = 1, \ldots, n$, where $S = \sum_{i=1}^{n} a_i$. Is there a subset $I \subseteq A$ such that $\sum_{a_i \in I} a_i = \sum_{a_i \in A \setminus I} a_i = \frac{S}{2}$?

**Proof.** Consider graph $G$ in Figure 6.1. For the weights associated with the labeled links $a_i \in A$, $0 < a_i < S$, for $i = 1, \ldots, n$, holds that $S = \sum_{i=1}^{n} a_i$. Links without labels have a cost of zero and $\Delta = S$. The objective is to find a pair of link-disjoint paths such that the total number of regenerators needed (shared or non-shared) for the two paths is minimized. There should definitively be regenerator(s) at node $t$: one in case of regenerator sharing, and two if there is no sharing. The next step is to decide whether more regenerators are required at other nodes. Let the two selected paths be

$P^1_{s-t-d}$ and $P^2_{s-t-d}$. The only scenario where no more regenerators are required is when their two corresponding segments have a cost $r\left(P^1_{s-t}\right) = r\left(P^2_{s-t}\right) = S$. However, this involves equally partitioning the labeled links $a_i \in A$, $i = 1, \ldots, n$ between the two paths. ■

We show that a min-sum link-disjoint paths algorithm that minimizes on the total weight of the two paths, such as Suurballe's algorithm, is an approximation algorithm. We begin with two lemmas that relate to a single unprotected path.

Given a path length $r(P)$ and a threshold $\Delta$, the number of regenerators required is determined by the length of each regeneration segment, as well as the total length covered by any two consecutive regeneration segments. For example, in Figure 6.2, where $r(P) = 12$ and $\Delta = 4$, only two regenerators are required in Figure 6.2(a), while four regenerators are needed in Figure 6.2(b).

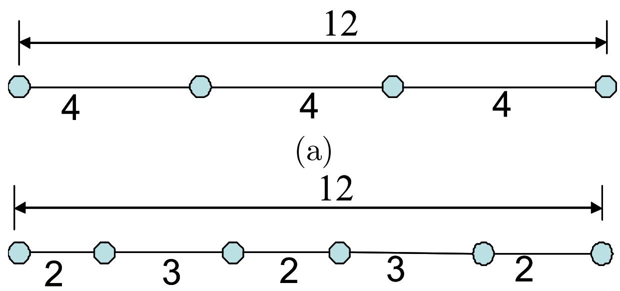



Figure 6.2: In both cases, $r(P) = 12$ and $\Delta = 4$. Only 2 regenerators are required in case (a); whereas 4 regenerators are required in case (b).

**Lemma 6.1** *The number of regenerators $R$ required by any simple path $P$ of length $r(P) > 0$ is bounded by* $\left\lceil \frac{r(P)}{\Delta} \right\rceil - 1 \leq R \leq 2 \left\lceil \frac{r(P)}{\Delta} \right\rceil$.

**Proof.** The number of regenerators required by the given path is minimized if each regeneration segment covers as much length as possible. Hence, a best-case scenario for path $P$ occurs when each regeneration segment, except possibly one, has a length of exactly equal to $\Delta$ as shown in Figure 6.3 (a). In this scenario, there will be $\left\lceil \frac{r(P)}{\Delta} \right\rceil$ regeneration segments, requiring $\left\lceil \frac{r(P)}{\Delta} \right\rceil - 1$ regenerators.

Without loss of generality, any placement of regenerators over a simple path $P$ can be described as in Figure 6.3. Furthermore, it is clear that for all $i$, $0 \leq \varepsilon_i < \mu_i$; otherwise, the regenerator between the segments of length $\Delta - \varepsilon_i$ and $\mu_i$ could be omitted. Similarly, for all $i < k$, $\mu_i > \varepsilon_{i+1}$.

Let $\beta = \min_i(\Delta - \varepsilon_i + \mu_i)$. Due to the above observation, $\beta > \Delta$. Thus, we have:

$$k\beta + \delta \leq \sum_{i=1}^{k} (\Delta - \varepsilon_i + \mu_i) + \delta = r(P). \tag{6.1}$$

We distinguish between two cases:

1. $\delta = 0$: From Equation (6.1),

$$k \leq \frac{r(P)}{\beta} \leq \frac{r(P)}{\Delta}.$$

For $\delta = 0$, the total number of regenerators, $R$, is equal to $2k - 1$. Thus,

$$R = 2k - 1 \leq \frac{2r(P)}{\Delta} - 1 \leq 2\left\lceil \frac{r(P)}{\Delta} \right\rceil - 1.$$

2. $\delta > 0$: From Equation (6.1),

$$k \leq \frac{r(P) - \delta}{\beta} \leq \frac{r(P)}{\Delta}.$$

For $\delta > 0$, the total number of regenerators, $R$, is at most $2k$. Thus,

$$R = 2k \leq \frac{2r(P)}{\Delta} \leq 2\left\lceil \frac{r(P)}{\Delta} \right\rceil.$$

Hence, $R \leq 2\left\lceil \frac{r(P)}{\Delta} \right\rceil$.

■

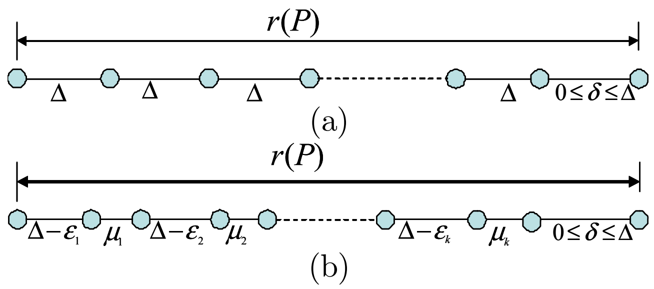



Figure 6.3: For a given path $P$ of length $r(P)$, (a) a best-case scenario, and (b) a representation of any regenerator placement.

**Lemma 6.2** *If the optimal path between nodes $s$ and $d$ requires $R^*$ regenerators, then the shortest (in terms of impairment) path from $s$ to $d$ requires at most $2(R^* + 1)$ regenerators.*

**Proof.** Let $P$ be the shortest path from $s$ to $d$, $r(P)$ be its length, and $R$ be its required number of regenerators. Let $P^*$ be the path that requires the optimal number of regenerators $R^*$. Hence, its length $r(P^*) \geq r(P)$.

Combining with Lemma 6.1,

$$R^* \geq \left\lceil \frac{r(P^*)}{\Delta} \right\rceil - 1 \geq \left\lceil \frac{r(P)}{\Delta} \right\rceil - 1.$$

By multiplying both sides by 2 and adding 2,

$$2(R^* + 1) = 2R^* + 2 \geq 2 \left\lceil \frac{r(P)}{\Delta} \right\rceil.$$

According to Lemma 6.1, the number of regenerators required by the shortest path $P$ is at most $2\left\lceil \frac{r(P)}{\Delta} \right\rceil$. ■

We are now ready to state our main result for the *dedicated-dedicated* case.

**Theorem 6.2** *Given an instance of the* dedicated-dedicated *SRSRP problem, the min-sum (in terms of impairment) link-disjoint pair of paths between s and d require at most* $2(R^* + 3)$ *regenerators, where* $R^*$ *is the optimal solution for the given* dedicated-dedicated *SRSRP instance.*

**Proof.** Let $P_1^*$ and $P_2^*$ be the pair of link-disjoint paths that give the optimal solution, and require $R_1^*$ and $R_2^*$ regenerators, respectively. Thus, $R_1^* + R_2^* = R^*$. Similarly, let $P_1$ and $P_2$ be the shortest pair of link-disjoint paths, and $R_1$ and $R_2$ be their respective required number of regenerators.

Since $r(P_1^*) + r(P_2^*) \geq r(P_1) + r(P_2)$,

$$\left\lceil \frac{r(P_1^*) + r(P_2^*)}{\Delta} \right\rceil \geq \left\lceil \frac{r(P_1) + r(P_2)}{\Delta} \right\rceil.$$

From the property of the ceiling function,

$$\left\lceil \frac{r(P_1^*)}{\Delta} \right\rceil + \left\lceil \frac{r(P_2^*)}{\Delta} \right\rceil \geq \left\lceil \frac{r(P_1^*) + r(P_2^*)}{\Delta} \right\rceil,$$
$$\left\lceil \frac{r(P_1) + r(P_2)}{\Delta} \right\rceil \geq \left\lceil \frac{r(P_1)}{\Delta} \right\rceil + \left\lceil \frac{r(P_2)}{\Delta} \right\rceil - 1.$$

Hence,

$$\left\lceil \frac{r(P_1^*)}{\Delta} \right\rceil + \left\lceil \frac{r(P_2^*)}{\Delta} \right\rceil \geq \left\lceil \frac{r(P_1)}{\Delta} \right\rceil + \left\lceil \frac{r(P_2)}{\Delta} \right\rceil - 1.$$

Multiplying both sides by 2 and adding 2, we get:

$$2\left(\left\lceil\frac{r(P_1^*)}{\Delta}\right\rceil - 1\right) + 2\left(\left\lceil\frac{r(P_2^*)}{\Delta}\right\rceil - 1\right) + 6 \geq$$
$$2\left\lceil\frac{r(P_1)}{\Delta}\right\rceil + 2\left\lceil\frac{r(P_2)}{\Delta}\right\rceil.$$

Hence, combining with Lemma 6.1 yields

$$2(R_1^* + R_2^* + 3) = 2(R^* + 3) \geq R_1 + R_2.$$

■

Similarly, we obtain the following result for the *dedicated-shared* case.

**Theorem 6.3** *Given an instance of the* dedicated-shared *SRSRP problem, the min-sum link-disjoint paths between* $s$ *and* $d$ *require at most* $4R^* + 6$ *regenerators, where* $R^*$ *is the optimal solution for the given* dedicated-shared *SRSRP instance.*

**Proof.** The best-case for the *dedicated-shared* SRSRP problem occurs when all regenerators of one of the paths are shared by the other path. Using the same notation as in the proof of Theorem 6.2, we have that $P_1^*$ requires at least $\left\lceil\frac{r(P_1^*)}{\Delta}\right\rceil - 1$ regenerators and $P_2^*$ requires at least $\left\lceil\frac{r(P_2^*)}{\Delta}\right\rceil - 1$ regenerators. W.l.o.g., assume that $r(P_2^*) \geq r(P_1^*)$. Hence, the two link-disjoint paths require at least $\left\lceil\frac{r(P_2^*)}{\Delta}\right\rceil - 1$ regenerators; otherwise $P_2^*$ is not feasible.

From the proof of Theorem 6.2, we have

$$2\left\lceil\frac{r(P_2^*)}{\Delta}\right\rceil \geq \left\lceil\frac{r(P_1)}{\Delta}\right\rceil + \left\lceil\frac{r(P_2)}{\Delta}\right\rceil - 1,$$
$$4\left(\left\lceil\frac{r(P_2^*)}{\Delta}\right\rceil - 1\right) + 6 \geq 2\left\lceil\frac{r(P_1)}{\Delta}\right\rceil + 2\left\lceil\frac{r(P_2)}{\Delta}\right\rceil.$$

Hence,

$$4R^* + 6 \geq R_1 + R_2.$$

■

The above results can be strengthened in the case where all links have equal cost. We begin with the following lemma.

**Lemma 6.3** *If all links in the network have equal cost, then the number of regenerators required by any path* $P$ *with length* $r(P)$ *exactly matches the lower bound.*

**Proof.** Let the cost of each link be $r$. We assume that $\Delta$ is a multiple of $r$. Otherwise, since all links have the same cost $r$, if a given segment satisfies the threshold $\Delta$, it also satisfies $r\left\lfloor\frac{\Delta}{r}\right\rfloor$. Therefore, $\Delta$ can be replaced by $r\left\lfloor\frac{\Delta}{r}\right\rfloor$. Thus, the regeneration segments require exactly $\left\lceil\frac{r(P)}{\Delta}\right\rceil - 1$. ■

We then obtain the following improved approximation bound for the *dedicated-dedicated* case.

**Theorem 6.4** *For a given instance of the* dedicated-dedicated *SRSRP problem, if all links in the network have equal cost, the min-sum link-disjoint paths between s and d require at most $R^* + 1$ regenerators, where $R^*$ is the optimal solution for the given* dedicated-dedicated *SRSRP instance.*

**Proof.** From the proof of Theorem 6.2, we have

$$\left\lceil\frac{r(P_1^*)}{\Delta}\right\rceil + \left\lceil\frac{r(P_2^*)}{\Delta}\right\rceil \geq \left\lceil\frac{r(P_1)}{\Delta}\right\rceil + \left\lceil\frac{r(P_2)}{\Delta}\right\rceil - 1.$$

Subtracting 1 on both sides,

$$\left(\left\lceil\frac{r(P_1^*)}{\Delta}\right\rceil - 1\right) + \left(\left\lceil\frac{r(P_2^*)}{\Delta}\right\rceil - 1\right) + 1 \geq$$
$$\left(\left\lceil\frac{r(P_1)}{\Delta}\right\rceil - 1\right) + \left(\left\lceil\frac{r(P_2)}{\Delta}\right\rceil - 1\right).$$

Combining this with Lemma 6.3 yields

$$R_1^* + R_2^* + 1 = R^* + 1 \geq R_1 + R_2.$$

■

Similarly, we obtain the following improved result for the *dedicated-shared* case.

**Theorem 6.5** *For a given instance of the* dedicated-shared *SRSRP problem, if all links in the network have equal cost, the min-sum link-disjoint pair of paths between s and d require at most $2R^* + 1$ regenerators, where $R^*$ is the optimal solution for the given* dedicated-shared *SRSRP instance.*

**Proof.** Using the same notations as the proof of Theorem 6.3, we have

$$2\left\lceil\frac{r(P_2^*)}{\Delta}\right\rceil \geq \left\lceil\frac{r(P_1)}{\Delta}\right\rceil + \left\lceil\frac{r(P_2)}{\Delta}\right\rceil - 1,$$
$$2\left(\left\lceil\frac{r(P_2^*)}{\Delta}\right\rceil - 1\right) + 1 \geq \left(\left\lceil\frac{r(P_1)}{\Delta}\right\rceil - 1\right) + \left(\left\lceil\frac{r(P_2)}{\Delta}\right\rceil - 1\right)$$

Combining this with Lemma 6.3, we have

$$2R^* + 1 \geq R_1 + R_2.$$

■

### Heuristic Algorithm

While the above algorithmic scheme, based on Suurballe's algorithm, provides proven (worst-case) performance guarantees, performance in typical scenarios could be improved. To that end, we present the following heuristic, termed Single Request Heuristic ($SRH$), for solving the **SRSRP** problem. Later in Section 6.2.3, we will show through simulations that $SRH$ performs better than Suurballe's algorithm, albeit without proven worst-case guarantees.

**Algorithm 6.1** $SRH(G, s, d, \Delta)$

1. Make a graph $G'(\mathcal{N}, \mathcal{L}')$, where $\mathcal{L}' = \{(u, v) \mid r(P^*_{u-v}) \leq \Delta\}$ and $P^*_{u-v}$ is the shortest path between $u$ and $v$. Assign a cost of 1 to each link in $G'$.
2. Find the shortest path $P'_{s-d}$ from $s$ to $d$ in $G'$.
3. Substitute all the links $(u, v)$ of $P'_{s-d}$ with the corresponding subpaths $P^*_{u-v}$ in $G$ to obtain $P_{s-d}$.
4. Remove all loops of $P_{s-d}$ in $G$ to obtain path $P_{s-d;1}$.
5. Redirect all links in $P_{s-d;1}$ from $d$ to $s$ to obtain $G''(\mathcal{N}, \mathcal{L}'')$ and assign a cost of 0 to these links.
6. On graph $G''$ repeat steps $1 - 4$ to obtain path $P_{s-d;2}$.
7. Remove links that are both in $P_{s-d;1}$ and $P_{s-d;2}$ to obtain two link-disjoint paths.
8. Place regenerators (shared or not shared depending on what is needed) for each path.

In Step 1 of algorithm $SRH$, graph $G'$ is constructed by connecting all directly reachable nodes (i.e., within $\Delta$). The links in graph $G'$ represent subpaths in graph $G$. Once the shortest path is obtained in Step 2, the path is transformed to its equivalent path $P_{s-d}$ in graph $G$. Since this path is made of a concatenation of path segments, it may not be a *simple* path in $G$. Hence, its loops are removed in Step 4 and the links along the loopless path $P_{s-d;1}$ are redirected from $d$ to $s$ to obtain graph $G''$ in Step 5. In Step 6, the same procedures are repeated in graph $G''$ to find the second loopless path $P_{s-d;2}$. For undirected graphs, the directed links in $G''$ may result in cases where $P^*_{u\to v} \neq P^*_{v\to u}$, in which case the graph obtained from $G''$ may contain two directed links between nodes $u$ and $v$, one in either direction. Once the second path $P_{s-d;2}$ is computed, the interlacing links between $P_{s-d;1}$ and $P_{s-d;2}$ are removed to obtain the solution. Finally, the regenerators are placed on these paths. For the shared variant, the regenerators for the primary lightpath are placed first, followed by those of the backup lightpath, while reusing the regenerators of the primary path wherever necessary.

The complexity of algorithm $SRH$ is determined by the following major operations: constructing graphs $G'$ and $G''$ in $O(N^2 \log N + NL)$ complexity (e.g., using Dijkstra's algorithm $N$ times); and finding the shortest paths between $s$ and $d$ in these graphs with Breadth First Search (BFS), which requires $O(L')$ and $O(L'')$. The paths obtained in $G'$ and $G''$ have $O(N)$ hopcount in the worst-case and at each node there can be a loop with $O(N)$ hopcount in graph $G$, thus the total complexity of removing loops is $O(N^2)$. Therefore, the total complexity of algorithm $SRH$ is $O(N^2 \log N + NL)$.

### 6.2.2 Shared Survivable Regenerator Placement

We turn to regenerator placement for shared protection.

**Shared Survivable Regenerator Placement (SSRP) Problem:** Given the input of **RPP** of Chapter 5, the problem is to find a pair of link-disjoint paths for each request that satisfy the impairment threshold on their regenerator segments. The objective is to place a minimum number of regenerators needed by all requests such that two backup lightpaths can share regenerators as long as their primary lightpaths do not share links.

The **SSRP** problem is NP-hard since it contains the **SRSRP** problem. We provide a corresponding heuristic algorithm.

#### Heuristic Algorithm

We employ an Active Path First ($APF$) approach where the primary path is computed first and then its links are dropped before the backup path is computed. This approach is chosen because it is easier to determine the sharing of resources among backup paths when the primary paths are already in place. In algorithm $APF$, once the primary path is computed, its links are dropped and the costs of links incident to "shareable" regenerator nodes are set to zero. This is to encourage the re-use of regenerators in those nodes.

### 6.2.3 Simulation Results

In this section, we evaluate the performance of the above heuristics, namely $SRH$ and $APF$, by way of simulations. Specifically, we provide simulation results that show the average number of regenerators needed (*per request*) on a USANET network (Figure 3.5) for impairment threshold values in the range $\Delta \in [1, 2]$. In our simulations, the impairment values of the links are randomly generated in the range $(0, 1]$, and the source and destination are randomly selected. The simulation results represent an average of

**Algorithm 6.2** $APF(G, \Delta)$

For each request $f$,

1. In $G$, find the shortest paths $\{P^*_{u-v}\}$ between all nodes $u, v \in \mathcal{N}$, for which $r(P^*_{u-v}) \leq \Delta$.
2. Create a graph $G^{'}(\mathcal{N}, \mathcal{L}^{'})$, where $\mathcal{L}^{'} = \{(u, v) \mid r(P^*_{u-v}) \leq \Delta\}$ and assign a cost of 1 to each link. Find the shortest path $P^{'}_{s_f-d_f}$ from $s_f$ to $d_f$ in $G^{'}$.
3. Substitute the links of $P^{'}_{s_f-d_f}$ with the corresponding subpaths $P^*_{u-v}$ in $G$ to obtain $P_{s_f-d_f}$.
4. Remove all loops of $P_{s_f-d_f}$ in $G$ to obtain path $P_{s_f-d_f;1}$. Place the necessary regenerators for $P_{s_f-d_f;1}$.
5. Remove all links in $P_{s_f-d_f;1}$ to obtain $G^{''}(\mathcal{N}, \mathcal{L}^{''})$.
6. For each primary path that does not share a link with $P_{s_f-d_f;1}$, set the cost of each link incident to the regenerator nodes of its backup path to zero.
7. Repeat Steps $a$–$d$ to obtain $P_{s_f-d_f;2}$. Place the necessary regenerators for $P_{s_f-d_f;2}$.

10 iterations for 100 requests. Figure 6.4 (a) shows a comparison of Suurballe's algorithm, $SRH$ and an exact solution (obtained via an ILP formulation given in Appendix C) for the *dedicated-dedicated* problem variant, while Figure 6.4(b) shows the same for the *dedicated-shared* problem variant. These results show that $SRH$ performs close to the exact solution in both variants of the dedicated survivable regenerator placement problem, even though it is slightly worse for *dedicated-shared*. In addition, it outperforms Suurballe's algorithm. Figure 6.5 is a comparison of dedicated ($SRH$) and shared ($APF$) survivable regenerator placement schemes. As one might expect, sharing of regenerators among backup lightpaths decreases the number of regenerators needed.

## 6.3 Conclusions

In this chapter, we have studied regenerator placement for dedicated and shared protection schemes. We have shown that the regenerator placement problem that minimizes the total number of regenerators is NP-hard for both schemes. For the case of dedicated protection, we established a (constant factor) approximation scheme based on Suurballe's algorithm; furthermore, we provided a heuristic scheme that, based on simulations, was shown to outperform the approximation scheme in typical scenarios. For

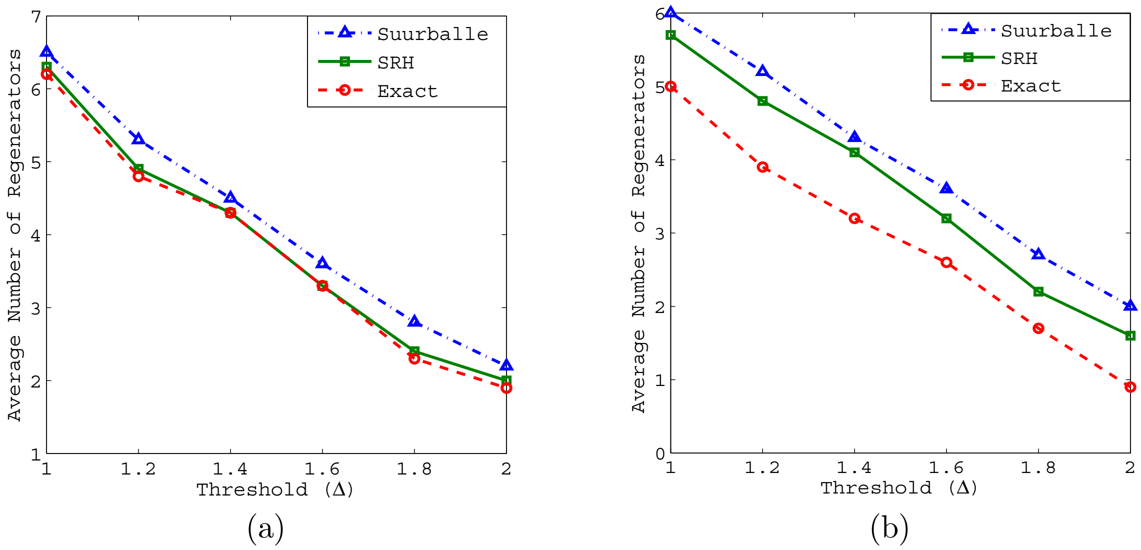



Figure 6.4: Comparison of the average number of regenerators needed for the (a) *dedicated-dedicated* and (b) *dedicated-shared* variants of the survivable regenerator placement problem.

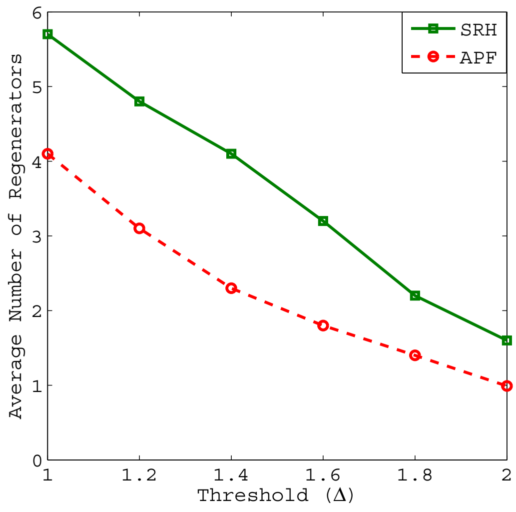



Figure 6.5: Comparison of the average number of regenerators needed in the case of dedicated and shared regenerators among backup lightpaths.

the case of shared protection, we have provided a heuristic algorithm, and demonstrated its appealing performance through simulations.

# Chapter 7

# Survivable Impairment-aware Traffic Grooming

## 7.1 Introduction

With current commercial technology, a lightpath can be independently operated at a data rate ranging up to 100 Gb/s [15]. However, traffic between a pair of nodes may not fill up the available bandwidth of a lightpath. Under such scenario, it will be wasteful to allocate a full wavelength capacity to each traffic stream. Thus, in order to efficiently utilize the available bandwidth, several independent traffic streams can be aggregated to share the capacity of a lightpath. This is known as *traffic grooming*, and it allows the aggregation of low-rate traffic onto high-rate traffic lightpaths.

Since lightpaths may carry a large amount of data, survivability, which is the ability to reconfigure and retransmit data after failure, is vital. This is usually achieved by computing a link/node-disjoint backup lightpath that will take over after failure of the primary lightpath. In addition, due to the signal degradation caused by physical impairments, a lightpath may require regeneration after a certain distance. A routing of lightpaths which takes into account physical impairments is known as *impairment-aware routing* [63].

In this chapter, we study *survivable impairment-aware traffic grooming* in WDM ring networks. Currently, ring topologies (such as SONET/SDH rings) are widely deployed in metro/regional networks [14] [24]. Nodes are assumed to be equipped with an optical add/drop multiplexer (OADM) to selectively add/drop wavelengths. We will follow the approach described in [42] and [55], where transceivers are used to terminate lightpaths. As in [42], the lightpaths are assumed to be full-duplex, and the forward and reverse direction signals use the same wavelength and path. Unless a wavelength carries traffic destined for a given node or needs regeneration, it passes through optically. Otherwise, all the following take place: (1) the lightpath is terminated, (2) the traffic is processed electronically (and regenerated simultaneously), (3) traffic destined to the node is dropped, and (4) the rest of the traffic, including locally added traffic, if any, is forwarded on other lightpaths through the transceivers. In this model, the cost of transceivers is the dominant component [42].

The outline of this chapter is as follows. In Section 7.2, we overview related work. In Section 7.3, we provide a formal definition of the survivable impairment-aware traffic grooming problem and show that it is NP-hard. In Section 7.4, we focus on the (basic) survivable traffic grooming problem by considering uniform and non-uniform traffic. For the former, we provide a constant-factor approximation algorithm, while we give a variable-factor approximation algorithm for the latter. In Section 7.5, we provide similar results for the survivable *impairment-aware* traffic grooming problem. Finally, we conclude in Section 7.7.

## 7.2 Related Work

The issue of traffic grooming has been widely studied in the literature, especially in relation to SONET/SDH rings over WDM networks. Most of the previous studies did not consider survivability or impairment-aware routing. Chiu and Modiano [27] studied the traffic grooming problem where the objective is to minimize the total number of SONET add-drop multiplexers (ADMs) in unidirectional SONET/WDM ring networks. They showed that the problem is NP-complete. The same problem was also shown to be NP-complete in bidirectional ring networks, where a request between two nodes can be routed on the clockwise or counter-clockwise direction [28]. Amini *et al.* [4] further showed that the grooming problem is APX-hard in WDM rings for a fixed value of grooming factor $g$, i.e., each request uses $1/g$ of the capacity of a wavelength. Huang *et al.* [50] studied traffic grooming in different topologies: line, star, and tree, and showed that traffic grooming is NP-complete in these topologies. Chen *et al.* [24] considered a different variant of the traffic grooming problem with a min-max objective where the cost at the node with the maximum grooming cost is minimized. They showed that this problem is NP-complete in both unidirectional and bidirectional rings.

Sankaranarayanan *et al.* [87] considered survivable traffic grooming in unidirectional WDM rings under uniform traffic with a mix of protected and unprotected requests. Ou *et al.* [77] gave heuristic algorithms for the survivable grooming in mesh networks, while Yao and Ramamurthy [115] considered the same problem under shared risk link group (SRLG) constraints, and provided heuristic algorithms. Unlike [77], [87] and [115], we consider both survivability *and* impairment-aware routing in WDM ring networks, and give constant-factor approximation (for uniform traffic) and a variable-factor approximation or a heuristic algorithm with provable upper-bounds (for non-uniform traffic). Patel *et al.* [80] considered impairment-aware traffic grooming, where regeneration is performed through regenerator cards. In this approach, there is a distinction between add/drop nodes and regeneration nodes, since regenerator cards are not capable of adding/dropping traffic. However, regeneration can also be achieved using back-to-back transceivers [94], in which case, regeneration nodes can also be used as add/drop nodes, and vice versa. We follow the second approach since it allows the use of the same type of devices for both add/drop and regeneration, and the regenerators may also be directly used as wavelength converters [94]. Flammini *et al.* [34] studied the case when regeneration and grooming are used interchangeably, but they (1) did not consider survivability, and (2) assumed that a regenerator has to be placed at every internal node of a lightpath, i.e., the signal must be regenerated at each internal node or the impairment threshold is equal to a single hop. In practice, the impairment threshold can be any given value, and it may not be necessary to regenerate a lightpath at every internal node.

## 7.3 Problem Definition

In this section, we give a formal definition of the survivable impairment-aware grooming problem. We define a *wavelength link* as a single wavelength channel of a given link, and a *wavelength ring* as a ring made up of wavelength links of the same wavelength. In effect, a ring comprises of stacks of wavelength rings. In a given wavelength ring, a *regenerator node* is a node where the wavelength is regenerated and an *add/drop node* is where traffic is added/dropped from the wavelength. A *wavelength segment* is a segment of a wavelength ring between two consecutive add/drop or regenerator nodes, i.e., there is no other add/drop or regenerator node in this segment. Associated with each wavelength link is an additive impairment value. The impairment value of a given wavelength segment is the sum of the impairment values of its links, and it is said to be a *feasible* segment if its impairment value does not exceed a given *impairment threshold*.

Splitting traffic of a single request might cause re-ordering problems at the receiving end as some higher layer protocols may not be able to deal with it. In addition, since the regenerator nodes as well as links that the signals go through may be different, it may lead to different signal quality. Therefore, we assume that the traffic of a given request is not split unless its demand exceeds the full capacity of a wavelength ring. In addition, in order to facilitate control, the primary and backup lightpaths of a given request are assumed to be on the same wavelength ring. Thus, for any given wavelength ring, the amount of traffic on each of its links is the same, and a pair of transceivers is required when a wavelength is added/dropped or regenerated at a given node. Since a wavelength is regenerated when traffic is added/dropped to it, an add/drop node is also a regenerator node. The network cost mainly comprises of the electronic and opto-electronic cost associated with grooming and regeneration (i.e., cost of transceivers), and the number of wavelengths. In practice, the cost of transceivers dominates the cost of the number of wavelengths [19] [42]. Hence, we minimize the total number of transceivers under the assumption that there are enough wavelengths to accommodate all the requests, which is equivalent to minimizing the total number of add/drop and regenerator nodes in the network.

**Problem 7.1** ***Survivable Impairment-aware Traffic Grooming:*** *Given is an undirected ring topology $\mathcal{G}(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is a set of $N$ nodes, $\mathcal{L}$ is a set of $N$ links. Associated with each link $l \in \mathcal{L}$ is an impairment value $r(l)$. A wavelength has a capacity $C$. In addition, given are an impairment threshold $\Delta$ and a set $\mathcal{F}$ of $F$ requests. Each request $f$ is represented by a pair of nodes $(u_f, v_f) = (v_f, u_f)$ and $\delta_f$, where $u_f$ and $v_f$ are the endpoints of request $f$, and $\delta_f$ is the amount of demand of request $f$. The survivable impairment-aware traffic grooming problem is to minimize the total number of transceivers (or add/drop and regenerator nodes) in the network such that (1) each request is assigned a primary and backup path, (2) the capacity of any wavelength link is not exceeded, and (3) each wavelength segment in any wavelength ring is feasible.*

**Theorem 7.1** *The survivable impairment-aware traffic grooming problem is NP-hard.*

We employ the NP-hard Bin Packing Problem [39], which is defined as follows.

**Problem 7.2** ***The Bin Packing Problem:*** *Given a finite set $\mathcal{U}$ of $n$ items, a size $a_i$ for each $u_i \in \mathcal{U}$, and a bin capacity $B$, the* ***bin packing problem*** *is to find a partition of $\mathcal{U}$ with the minimum number of disjoint sets (bins) $S_1, \ldots, S_K$ such that the sum of the sizes of the items in each $S_i$ is less than $B$.*

**Proof.** We show that the survivable traffic grooming problem, which is a subset of the survivable impairment-aware traffic grooming problem (by taking $\Delta$ sufficiently large) is NP-hard. For a given instance of the bin packing problem of $n$ items, create a corresponding survivable traffic grooming problem as follows. For each item $i$, create a corresponding node $i$ in the ring. Create a hub node $h$. Therefore, the number of nodes $N = n + 1$. Let the capacity $C$ of a wavelength be equal to the bin capacity $B$, and there is a request of demand $\delta_i = a_i$ between each node $i$ and the hub node $h$.

Since there is exactly one request originating at each node (except the hub node $h$), any feasible solution requires an add/drop node at each of the $N-1$ nodes. Hence, only the total number of add/drop nodes at node $h$ can be minimized. Since there is one add/drop node per used wavelength at node $h$, the total number of add/drop nodes is the same as the total number of wavelength rings. Therefore, the objective is to minimize the total number of wavelength rings. On the other hand, each wavelength ring is equivalent to a bin, and the requests in the wavelength ring are equivalent to the items in the bin of the corresponding bin packing problem instance. Therefore, minimizing the total number of wavelength rings required for all the requests is equivalent to minimizing the number bins of the corresponding bin packing problem instance. ■

## 7.4 Survivable Traffic Grooming

We begin with the case where no regeneration is required in the ring network. We will consider uniform and non-uniform traffic independently.

### 7.4.1 Uniform Traffic

In a uniform traffic scenario, there is a request of equal demand $\delta$ between each pair of nodes. Thus, there are a total of $\frac{N(N-1)}{2}$ requests, each with a demand of $\delta$. Even though, this type of traffic is less practical, it can help us gain insight into the complexity of survivable traffic grooming. In addition, as shown in [28], it may be possible to extend the results obtained for uniform traffic to that of the more practical *quasi-uniform* traffic. An algorithm is said to be an *$\alpha$-approximation algorithm*, for some $\alpha > 1$, if it returns at most $\alpha$ times the optimal number of transceivers (add/drop nodes). Before we provide an approximation algorithm for solving the survivable traffic grooming problem under uniform traffic, we give a lower-bound for the total number of add/drop nodes.

**Theorem 7.2** *For uniform traffic, the total number of add/drop nodes $m$ is lower bounded by:*

$$m \geq \left\lceil \sqrt{\frac{\delta}{2C}} N(N-1) \right\rceil . \tag{7.1}$$

**Proof.** We provide a proof along the lines of the proof given in [28] for unprotected traffic grooming. Given a feasible solution $S$, let $G(A)$ be the wavelength ring on which add/drop node $A$ is. For each add/drop node A, define $B(A)$ as:

$$B(A) = \frac{\text{Total bandwidth of the traffic carried on wavelength ring } G(A)}{\text{Total number of add/drop nodes on wavelength ring } G(A)}.$$

For a given wavelength ring of $S$, let $k$ be the number of add/drop nodes. Hence, there can be at most $k(k-1)/2$ requests in this ring and the total bandwidth requirement (i.e., the sum of bandwidth needed on all wavelength links) of all the requests in this wavelength ring is at most $\delta N k(k-1)/2$.
Since the total bandwidth capacity of a wavelength ring is $CN$,

$$\begin{aligned} B(A) &\leq \frac{\min(\delta N k(k-1)/2, CN)}{k} = CN \min\left(\frac{\delta}{2C}(k-1), 1/k\right) \\ &\leq CN \min\left(\frac{\delta}{2C}k, 1/k\right) \leq CN\sqrt{\frac{\delta}{2C}} = N\sqrt{\frac{\delta C}{2}}. \end{aligned} \tag{7.2}$$

The last inequality is due to the property that $min(ak, 1/k) \leq \sqrt{a}$ for any $k > 0$. Let $B$ be the total amount of bandwidth consumed by all the requests. Summing the last inequality of Equation 7.2 (which is independent of $k$) over all the add/drop nodes,

$$B \leq mN\sqrt{\frac{\delta C}{2}}.$$

For uniform traffic, the total bandwidth $B$ is

$$B = \delta N\left(\frac{N(N-1)}{2}\right).$$

From which Equation 7.1 follows. ■

In order to show that the given lower-bound is tight, let $N = 3$, $\delta = 1$ and $C = 3$. The requests are $(1, 2), (1, 3)$ and $(2, 3)$, and they can all fit in a single wavelength ring, in which case the number of add/drop nodes is $3 \geq \left\lceil\sqrt{\frac{\delta}{2C}}N(N-1)\right\rceil = \left\lceil\frac{6}{\sqrt{6}}\right\rceil = 3$.

**Corollary 7.1** *Any survivable traffic grooming algorithm is a* $\sqrt{\frac{2C}{\delta}}$*-approximation algorithm for uniform traffic.*

**Proof.** This follows from the fact that in the worst-case there is no grooming at all, i.e., each request is added/dropped independently. Since there are $\frac{N(N-1)}{2}$ requests, a total of $N(N-1)$ add/drop nodes will be needed in this case. However, by Theorem 7.2, we have that the optimal number of add/drop nodes is at least $\sqrt{\frac{\delta}{2C}}N(N-1)$. Thus, the approximation ratio is $\sqrt{\frac{2C}{\delta}}$. ■

We now provide an algorithm for the survivable traffic grooming problem, termed *USGA* (Uniform traffic Survivable Grooming Algorithm), and show that it is a $\min\left(\sqrt{\frac{2C}{\delta}}, 4\right)$-approximation algorithm for uniform traffic.

In Step 1 of *USGA*, if the demand per request is greater than the capacity of a wavelength ring, a separate wavelength ring(s) is assigned for each request and the remaining traffic of the request is assigned a wavelength ring in the next steps. In Step 2, if $N \leq \sqrt{\frac{2C}{\delta}}$, all the requests can optimally fit in a single wavelength ring. Similarly in Step 3, if $C < 2\delta$, only a single request can be assigned in a wavelength ring. Step 4 partitions the nodes into a group of sets, and Step 5 combines a pair of these sets in such a way that any pair of nodes belongs to at least one of the newly-formed sets. Once the sets are created, the requests are assigned sequentially in Step 6. Since there are $O\left(N/\sqrt{\frac{C}{2\delta}}\right)$ sets in Step 4, there will be $O\left(\frac{N^2\delta}{C}\right)$ sets in Step 5. The most time-consuming operation in *USGA* is Step 6, where for each wavelength ring, requests between each pair of its add/drop nodes are considered to decide whether they

**Algorithm 7.1** *USGA*$(G, \mathcal{F}, C)$

1. If $\delta \geq C$, then assign $\left\lfloor \frac{\delta}{C} \right\rfloor$ wavelength rings for each request and let $\delta = \delta - C \left\lfloor \frac{\delta}{C} \right\rfloor$.
2. If $N \leq \sqrt{\frac{2C}{\delta}}$, assign all the requests in one wavelength ring and exit.
3. If $C < 2\delta$, then assign a single request per wavelength ring and exit.
4. Let $k = \left\lfloor \sqrt{\frac{C}{2\delta}} \right\rfloor$. Partition the $N$ nodes into $\left\lceil \frac{N}{k} \right\rceil$ sets such that each set, except possibly one, contains $k$ distinct nodes and each node belongs to *exactly* one set.
5. For each pair of sets among those created in Step 4, create a set which is the union of this pair of sets.
6. Sequentially, for each set in Step 5, assign a separate wavelength ring as follows:
   (a) Each node in the set is an add/drop node (i.e., transceivers are placed).
   (b) For each pair of nodes, allocate the primary and backup path of the corresponding request in this wavelength ring, unless the request has already been allocated in a previous wavelength ring.

---

belong to the wavelength ring. Since the size of a set is at most $2k = 2\left\lfloor \sqrt{\frac{C}{2\delta}} \right\rfloor$ and each pair of nodes in the set is considered, Step 6 has a total running time of $O\left(N^2\right)$.

Through the following example, we illustrate how the algorithm works. Let $N = 7$, $C = 9$ and $\delta = 1$. Thus, $k = \left\lfloor \sqrt{\frac{C}{2\delta}} \right\rfloor = 2$. The nodes are then grouped into sets of at most 2 elements: $\{1, 2\}$, $\{3, 4\}$, $\{5, 6\}$, $\{7\}$. By combining each pair of sets, we get $\{1, 2, 3, 4\}$, $\{1, 2, 5, 6\}$, $\{1, 2, 7\}$, $\{3, 4, 5, 6\}$, $\{3, 4, 7\}$, $\{5, 6, 7\}$. For each set, a separate wavelength ring is used, a pair of transceivers is assigned at all its nodes, and the requests between each pair of nodes are allocated in this wavelength ring unless they have been allocated before. For example, the request between nodes 1 and 2 is assigned only to the first wavelength ring.

We proceed to establish the correctness of *USGA*.

**Theorem 7.3** *The following holds for the outcome of USGA: (1) The capacity of any of the wavelength links is not exceeded, and (2) each request is assigned a primary and a backup path.*

**Proof.** (1) In each wavelength ring, there are at most $2k = 2\left\lfloor \sqrt{\frac{C}{2\delta}} \right\rfloor$ add/drop nodes. Thus, there can be at most $2k(2k-1)/2$ requests in any given wavelength ring. Hence, the total capacity required at any wavelength link is at most,

$$\frac{\delta 2k(2k-1)}{2} = \frac{\delta\left(2\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor\right)\left(2\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor-1\right)}{2} \leq \frac{\delta\left(2\sqrt{\frac{C}{2\delta}}\right)^2}{2} = C.$$

(2) Each node belongs to at least one set in Step 4. Since the sets in Step 5 are a combination of each pair of sets in Step 4, any given pair of nodes belongs to at least one set in Step 5. Thus, the corresponding request is allocated primary and backup paths in Step 6. ■

**Theorem 7.4** *USGA is a* $\min\left(\sqrt{\frac{2C}{\delta}}, 4\right)$ *−approximation algorithm.*

**Proof.** Let

$$a = \begin{cases} 0, \text{ if } (N \bmod k) = 0; \\ k - (N \bmod k), \text{ otherwise.} \end{cases}$$

In Step 4, there are a total of $\left\lceil\frac{N}{k}\right\rceil = \frac{N+a}{k}$ sets, and each set contains $k$ elements, except possibly the last set that has only $k-a$ elements if $a > 0$. Hence, the total number of sets in Step 5 is $\frac{\lceil\frac{N}{k}\rceil(\lceil\frac{N}{k}\rceil-1)}{2} = \frac{\frac{N+a}{k}(\frac{N+a}{k}-1)}{2}$, and each set requires at most $2k$ add/drop nodes. However, among these sets, there are $\left(\frac{N+a}{k}-1\right)$ sets that require only $2k-a$ add/drop nodes. Hence, the total number of add/drop nodes is

$$\left(\frac{\frac{N+a}{k}(\frac{N+a}{k}-1)}{2}\right)2k - \left(\frac{N+a}{k}-1\right)a = N\left(\frac{N+a}{k}-1\right) = N\left(\frac{N+a-k}{k}\right).$$

By definition, $a \leq k-1$. Thus, the total number of add/drop nodes is at most $N\left(\frac{N-1}{k}\right)$. Combining this with Theorem 7.2, the approximation ratio $\alpha$ is,

$$\alpha \leq \frac{N\left(\frac{N-1}{k}\right)}{\sqrt{\frac{\delta}{2C}}N(N-1)} = \frac{\sqrt{\frac{2C}{\delta}}}{\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor} = 2\frac{\sqrt{\frac{C}{2\delta}}}{\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor} \leq 4.$$

The last inequality is due to the fact that since $C \geq 2\delta$, $\frac{\sqrt{\frac{C}{2\delta}}}{\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor} < 2$. Combined with Corollary 7.1, this proves our theorem. ■

## 7.4.2 Non-uniform Traffic

Non-uniform traffic is a general scenario where the amount of demand between nodes is arbitrary. For any node $u$, let $\mathcal{F}_u$ be the set of requests for which node $u$ is an endpoint, $F_u$ be the number of such requests (i.e., $F = \frac{1}{2}\sum_{u=1}^{N} F_u$), and $C_u = \sum_{(u,v)\in\mathcal{F}_u} \delta_{(u,v)}$. For any node $u$, let $OPT_u$ be the optimal solution for the corresponding bin packing problem of set $\mathcal{F}_u$. We first provide a simple lower-bound for non-uniform traffic.

**Theorem 7.5** *For non-uniform traffic, the total number of add/drop nodes $m$ is bounded by:*

$$m \geq \sum_{u=1}^{N} OPT_u.$$

**Proof.** The number of add/drop nodes at any node $u$ is the same as the number wavelength rings terminated at this node. As shown in the proof of Theorem 7.1, at any given node $u$, the minimum possible number of such wavelength rings is the same as the solution of the corresponding instance of the bin packing problem (i.e., for each $(u, v) \in \mathcal{F}_u$, there is an item of size $\delta_{(u,v)}$). ■

We now provide an approximation algorithm, termed $NSGA$ (Non-uniform traffic Survivable Grooming Algorithm), for the non-uniform traffic case (see Algorithm 7.2). The algorithm considers each node sequentially and allocates wavelength rings for requests originating at this node by first solving a corresponding bin packing problem instance.

---

**Algorithm 7.2** $NSGA(G, \mathcal{F}, C)$

---

1. Sort the nodes in non-increasing order according to $F_u$.
2. For each node $u$,
   (a) Let $\mathcal{F}'_u = \{(u, v) = (v, u) | u < v\}$ and $F'_u = |\mathcal{F}'_u|$.
   (b) Create an instance of the bin-packing problem such that for each request $(u, v) \in \mathcal{F}'_u$, there is an item whose size is the demand of the request. Use the *first fit decreasing (FFD)* algorithm [58] as follows:
      i. Sort the items in non-increasing order.
      ii. Go through all the items by placing the current item in the lowest indexed bin that has enough space left, otherwise create a new bin for it.
   (c) For each bin in the solution of the bin packing problem instance, create a new wavelength ring and place the requests corresponding to the items of the bin in this wavelength ring.

---

For any node $u$, let $OPT'_u$ be the optimal solution for the corresponding bin packing problem of set $\mathcal{F}'_u$.

**Theorem 7.6** *$NSGA$ is a $\left(\frac{3}{2} + \frac{F}{\sum_{u=1}^{N} OPT_u}\right)$-approximation algorithm.*

**Proof.** For any node $u$, the FFD algorithm in Step $2b$ returns at most $\frac{3}{2}OPT'_u$ bins [93]. In Step $2c$, a separate wavelength ring is assigned for each bin, and on each wavelength ring, node $u$ is an add/drop node. Additionally, at the other end of each request, an add/drop is required. Thus, the total number of add/drop nodes required by $NSGA$ when node $u$ is considered in Step 2 is at most $\frac{3}{2}OPT'_u + F'_u$. Since $OPT'_u \leq OPT_u$ for each node $u$ and $\sum_{u=1}^{N} F'_u = F$, combined with Theorem 7.5, this proves our theorem. ■

The scenario given in the proof of Theorem 7.1, where there is a hub node and there is a single request destined to each of the other nodes, can be used to show the tightness of this upper-bound. In this scenario, each request needs an add/drop node at its other end. Thus, a total of $F$ add/drop nodes. At the hub node, the number of add/drop nodes is equal to the number of bins of the corresponding bin-packing problem, for which the FFD algorithm is a $\frac{3}{2}$-approximation [93].

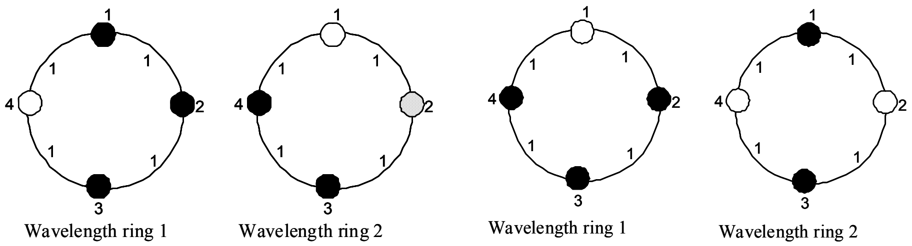
## 7.5 Survivable Impairment-aware Traffic Grooming

In this section, we consider the general problem of survivable impairment-aware traffic grooming, where transceivers are used not only for adding/dropping traffic but also for regeneration. Consider the following example to illustrate the difference from the previous impairment-agnostic problem. Let $N = 4$, $C = 2$, $\Delta = 2$, and each link has an impairment value of 1. Let the requests be $(1, 3)$, $(2, 3)$ and $(3, 4)$, and each request has a demand of $\delta = 1$. For the survivable traffic grooming problem, both of the following solutions (see Figure 7.1) are optimal, and each solution requires a total of 5 add/drop nodes. Solution 1: $\{(1, 3), (2, 3)\}$ on the first wavelength ring and $\{(3, 4)\}$ on the second wavelength ring; and Solution 2: $\{(2, 3), (3, 4)\}$ on the first wavelength ring and $\{(1, 3)\}$ on the second wavelength ring. However, for the survivable impairment-aware traffic grooming problem, only solution 2, which needs no extra regenerator node, is optimal. Solution 1 requires an extra regenerator node at node 1 or node 2 of the second wavelength ring to accommodate the backup path of $(3, 4)$.

The number of regenerator nodes needed to feasibly and survivably route all the assigned requests in a wavelength ring depends on the endpoints of the requests (or the wavelength segments). However, it is possible to determine the minimum number $R$ of regenerator nodes required at any wavelength ring for a given survivable impairment-aware grooming problem using the procedure $FindR$ (see Algorithm 7.3).

**Theorem 7.7** *FindR returns the minimum number of regenerator nodes required on any wavelength ring.*

**Proof.** We give a proof by contradiction. Assume that the minimum number of regenerator nodes is $R' < R$. Let these regenerator nodes be $n_1, ..., n_{R'}$ in the clockwise direction. W.l.o.g., for each node $n_j$, node $n_{j+1}$ is the farthest reachable node from $n_j$ in




(a) Solution 1 (b) Solution 2

Figure 7.1: An example showing the difference between the survivable traffic grooming problem and the survivable impairment-aware traffic grooming problem. Solid nodes are add/drop nodes and shaded nodes are regenerator nodes.

---

**Algorithm 7.3** FindR

1. For each node $u$, find a value $R_u$ as follows:

   (a) Initialize $R_u$ to 0. Beginning at node $u$, move clockwise (can also be anti-clockwise, but should be consistent) to the farthest reachable node $v$ (i.e., within a distance of $\Delta$) from node $u$ and increment $R_u$. Let $r_1$ be the distance between $u$ and $v$ in the clockwise direction.

   (b) Then, at node $v$ move to the farthest clockwise reachable node and increment $R_u$, and so on, until just before we cross node $u$, at which point let the last such node be $t$. Let $r_2$ be the distance between $t$ and $u$ in the clockwise direction. If $r_1 + r_2 > \Delta$, increment $R_u$.

2. $R = \min_u\{R_u\}$.

---

the clockwise direction, otherwise the regeneration at $n_{j+1}$ can be moved to the farthest reachable node. In addition, the distance between $n_{R'}$ and $n_2$ exceeds $\Delta$, otherwise the regeneration at $n_1$ is not necessary. Thus, employing $FindR$ at node $n_1$ returns $R'$, which is a contradiction. ■

### 7.5.1 Uniform Traffic

We first provide a lower-bound for the survivable impairment-aware traffic grooming problem under uniform traffic.

**Theorem 7.8** *For uniform traffic, the total number of add/drop and regenerator nodes*

*$m$ is bounded by:*

$$m \geq \max\left(\left\lceil\sqrt{\frac{\delta}{2C}}N(N-1)\right\rceil, R\left\lceil\frac{N(N-1)}{2\left\lfloor\frac{C}{\delta}\right\rfloor}\right\rceil\right).$$

**Proof.** Since transceivers are required for adding/dropping traffic as well as regeneration, the total number of transceivers depends on which scenario is dominant. If adding/dropping is the dominant factor, Theorem 7.2 gives that $m \geq \left\lceil\sqrt{\frac{\delta}{2C}}N(N-1)\right\rceil$. Therefore, we need to show only the case when the number of regenerator nodes dominates. The maximum number of requests that can be assigned in any wavelength ring is $\left\lfloor\frac{C}{\delta}\right\rfloor$. Since, we have a total $\frac{N(N-1)}{2}$ requests, we need at least $\left\lceil\frac{N(N-1)}{2\left\lfloor\frac{C}{\delta}\right\rfloor}\right\rceil$ wavelength rings to accommodate all the requests, and each wavelength ring requires at least $R$ regenerator nodes. ■

Algorithm $USGA$ can be reused for solving the survivable impairment-aware traffic grooming problem as follows: (1) Solve the corresponding survivable traffic grooming problem, (2) From this solution, for each wavelength ring identify non-feasible segments and place regenerator nodes to make these wavelength segments feasible. For each wavelength segment, this can be done using the regenerator placement algorithm in [63]. We first give approximation ratios for this approach. However, the approximation ratios may be too high for practical use. Therefore, we suggest a scheme to improve the average performance of $USGA$, while maintaining the worst-case ratio.

**Theorem 7.9** *$USGA$ has an approximation ratio of 16 if $R \leq \sqrt{\frac{2C}{\delta}}$, and 20 otherwise.*

**Proof.** We use the same notation as in Theorem 7.4. W.l.o.g, the lower-bound on $m$ can be replaced with $\max\left(\sqrt{\frac{\delta}{2C}}N(N-1), R\frac{N(N-1)}{2\frac{C}{\delta}}\right)$. From the proof of Theorem 7.4, the total number of add/drop nodes needed by $USGA$ is at most $N\left(\frac{N-1}{k}\right)$. In the worst-case, we additionally need a total of $R$ regenerator nodes on each wavelength ring. Thus, the total number of add/drop and regenerator nodes is at most

$$\begin{aligned} N\left(\frac{N-1}{k}\right) + \left(\frac{\frac{N+a}{k}(\frac{N+a}{k}-1)}{2}\right)R &= N\left(\frac{N-1}{k}\right) + R\left(\frac{\frac{N+a}{k}(\frac{N+a-k}{k})}{2}\right) \\ &= \left(\frac{N-1}{k}\right)\left(N + R\left(\frac{N+a}{2k}\right)\right) \\ &\leq \left(\frac{N-1}{k}\right)\left(N + R\left(\frac{1.5N}{2k}\right)\right) \end{aligned}$$

The second equality is because $a \leq k-1$. The last inequality is because $2a \leq 2k \leq N$ (See Step 2 of $USGA$). We consider two cases.

**Case 1:** $\sqrt{\frac{\delta}{2C}}N(N-1) \geq R\frac{N(N-1)}{2\frac{C}{\delta}}$ or $R \leq \sqrt{\frac{2C}{\delta}}$.
The total number of add/drop and regenerator nodes is:

$$\begin{aligned}\left(\frac{N-1}{k}\right)\left(N+R\left(\frac{1.5N}{2k}\right)\right) &\leq \left(\frac{N-1}{k}\right)\left(N+1.5N\frac{\sqrt{\frac{2C}{\delta}}}{2\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor}\right)\\ &\leq 4N\left(\frac{N-1}{k}\right).\end{aligned}$$

The first inequality is because $R \leq \sqrt{\frac{2C}{\delta}}$, and the last inequality is due to the fact that since $C \geq 2\delta$, $\frac{\sqrt{\frac{C}{2\delta}}}{\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor} < 2$.
Since $k = \left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor$, the approximation ratio $\alpha$ is:

$$\alpha \leq \frac{4N\left(\frac{N-1}{k}\right)}{\sqrt{\frac{\delta}{2C}}N(N-1)} \leq 4\frac{\sqrt{\frac{2C}{\delta}}}{\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor} = 8\frac{\sqrt{\frac{C}{2\delta}}}{\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor} \leq 16.$$

**Case 2:** $\sqrt{\frac{\delta}{2C}}N(N-1) \leq R\frac{N(N-1)}{2\frac{C}{\delta}}$ or $R \geq \sqrt{\frac{2C}{\delta}}$.
The total number of add/drop and regenerator nodes is:

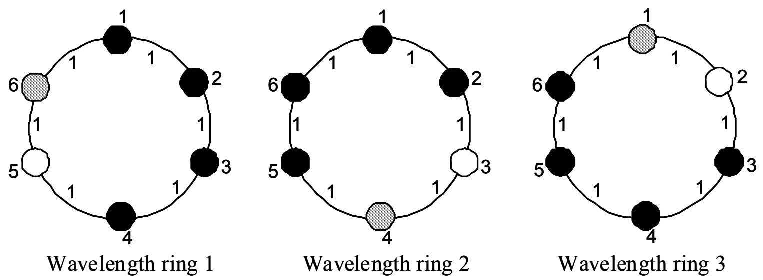
$$\begin{aligned}\left(\frac{N-1}{k}\right)\left(N+R\left(\frac{1.5N}{2k}\right)\right) &= N\left(\frac{N-1}{k}\right)\left(1+\frac{1.5}{2k}R\right)\\ &\leq N\left(\frac{N-1}{k}\right)\left(\frac{2.5}{2k}R\right).\end{aligned}$$

The last inequality follows from $R \geq \sqrt{\frac{2C}{\delta}} \geq 2\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor = 2k$.
The approximation ratio $\alpha$ is:

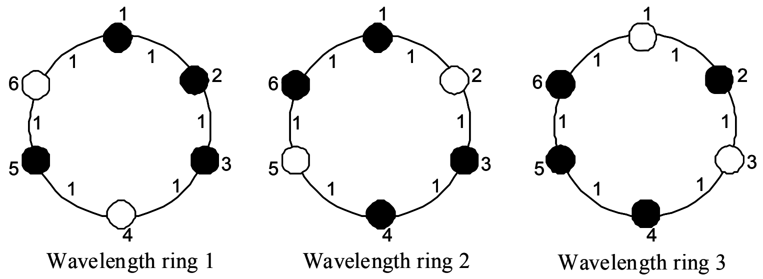
$$\alpha \leq \frac{N\left(\frac{N-1}{k}\right)\left(\frac{2.5}{2k}R\right)}{R\left(\frac{N(N-1)}{2\frac{C}{\delta}}\right)} = 2.5\,\frac{\frac{C}{\delta}}{\left(\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor\right)^2} \leq 5\frac{\frac{C}{2\delta}}{\left(\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor\right)^2} = 5\left(\frac{\sqrt{\frac{C}{2\delta}}}{\left\lfloor\sqrt{\frac{C}{2\delta}}\right\rfloor}\right)^2 \leq 20.$$

■

The average performance of $USGA$ can be improved by rearranging the nodes before creating the sets in Step 4. The basic idea is to group together pairs of nodes that are within a distance close to the impairment threshold so that the number of extra regenerator nodes is reduced. We show this using the following example. Let $N = 6$,




(a) Solution of $USGA$.




(b) Solution of $USGA$ after rearranging the nodes.

Figure 7.2: An example showing how rearranging the order of nodes affects the solution of $USGA$. Solid nodes are add/drop nodes, while shaded nodes are extra regenerator nodes.

$C = 8$, $\delta = 1$, and $\Delta = 2$. Thus, $k = 2$. By simply applying $USGA$, the sets in Step 4 will be $\{1, 2\}$, $\{3, 4\}$, $\{5, 6\}$, and the sets in Step 5 are $\{1, 2, 3, 4\}$, $\{1, 2, 5, 6\}$, $\{3, 4, 5, 6\}$. This solution will require a total of 12 add/drop nodes and 3 extra regenerator nodes (one in each wavelength ring as shown in Figure 7.2(a)). However, if the nodes are rearranged in such a way that pairs of nodes with a distance of $\Delta$ or more are grouped together, the sets in Step 4 will be $\{1, 3\}$, $\{5, 2\}$, $\{4, 6\}$, and the sets in Step 5 are $\{1, 3, 5, 2\}$, $\{1, 3, 4, 6\}$, $\{5, 2, 4, 6\}$. This solution will require 12 add/drop nodes and no extra regenerator nodes (see Figure 7.2(b)). In general, the nodes can be rearranged before applying $USGA$ as follows:

- Mark node 1, then mark the clockwise unmarked node $i$ that is at a distance of $\Delta$ from node 1 or is the first unmarked node that is unreachable (i.e., whose distance is larger than $\Delta$) from node 1.

- Repeat this process from node $i$, until all nodes are marked.

### 7.5.2 Non-uniform Traffic

We first give a lower-bound for non-uniform traffic. Let $BIN$ be the optimal number of bins required for the following instance of the bin packing problem: For each request

$f$, create an item $a_f$ of size $\delta_f$, and let the bin capacity $B = C$.

**Theorem 7.10** *For non-uniform traffic, the total number of add/drop and regenerator nodes $m$ is lower-bounded by:*

$$m \geq \max\left(\sum_{u=1}^{N} OPT_u, R \cdot BIN\right).$$

**Proof.** To accommodate all the requests, at least $BIN$ number of wavelength rings are required. On each wavelength ring, at least $R$ number of regenerator nodes are needed. Combined with Theorem 7.5, this proves our theorem. ■

In order to solve the survivable impairment-aware traffic grooming problem, we modify $NSGA$ in such a way that after the requests are assigned to wavelength rings and add/drop nodes are identified, for each wavelength ring, we place the extra regenerator nodes required to make all its wavelength segments feasible. Using the same notation as in Theorem 7.6, the following theorem can be established.

**Theorem 7.11** *The total number of add/drop nodes $m$ returned by the modified $NSGA$ is upper-bounded by:*

$$m \leq \frac{3}{2}(R+1)\left(\sum_{u=1}^{N} OPT'_u\right) + F.$$

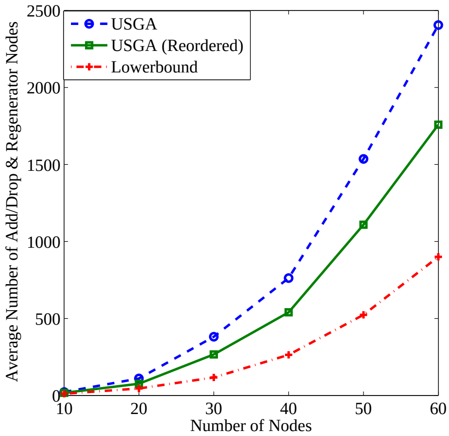
**Proof.** As is shown in the proof of Theorem 7.6, the number of wavelength rings returned by $NSGA$ is at most $\frac{3}{2}\left(\sum_{u=1}^{N} OPT'_u\right)$. In the worst-case, we need $R$ extra regenerator nodes in each wavelength ring. ■

This upper-bound is tight since it reduces to the upper-bound of $NSGA$ given in Theorem 7.6 when $R = 0$ (i.e., when the impairment threshold is sufficiently large).

## 7.6 Simulation Results

We first provide simulation results showing the performance gain achieved by rearranging the order of nodes before applying $USGA$ as described in Section 7.5.1. Figures 7.3(a) and 7.3(b) show the performance of $USGA$ (with and without reordering) against the lower-bound for different number of nodes (fixed capacity) and different capacity (fixed number of nodes), respectively. From these results, we observe that (1) even though $USGA$ has an approximation ratio of 16 or 20, the performance ratio against the lower-bound is at most 4 in these results, and (2) reordering the nodes provides a performance gain as high as 30%.

Figures 7.4(a) and 7.4(b) show the results obtained for $NSGA$ when solving the survivable traffic grooming problem, while Figures 7.5(a) and 7.5(b) are for the modified



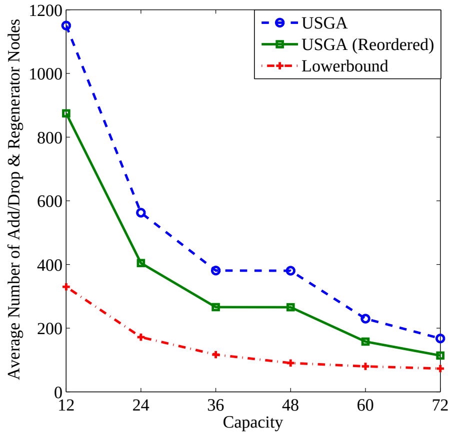
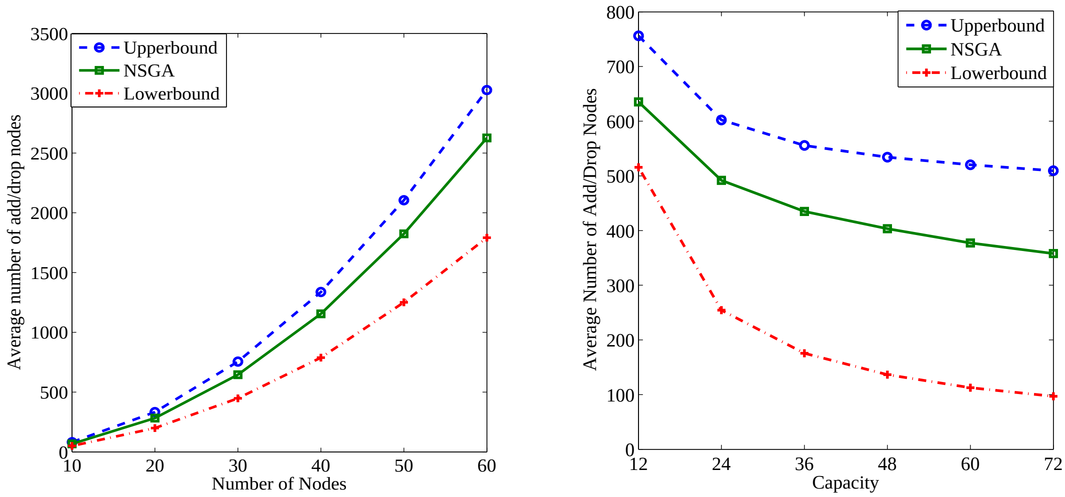




Figure 7.3: Comparison of the number of add/drop and regenerator nodes required by $USGA$ (with and without ordering) in the survivable impairment-aware traffic grooming problem for (a) different number of nodes ($C = 36$, demand values are within the range $[0, C]$), and (b) different capacity ($N = 30$ and demand values are within the range $[0, 12]$). The impairment values are uniformly distributed within the range $(0, 1]$ and $\Delta = 1$.

$NSGA$ when solving the survivable impairment-aware traffic grooming problem under non-uniform traffic. These figures show that the results of $NSGA$ are not generally far-off from the lower-bounds of the optimal solutions. In addition, the lower-bound is based on the assumption that all wavelength rings are fully utilized, but in reality, this is not the case as some wavelength rings will only be partially utilized since requests are not allowed to be split. Therefore, the optimal solution will in practice be much higher than the lower-bound. Since finding the optimal solution (e.g., using exact Integer Linear Programming (ILP) formulations) is not tractable even for small sized networks and small number of requests, $NSGA$'s performance and scalability make it suitable for practical purposes.

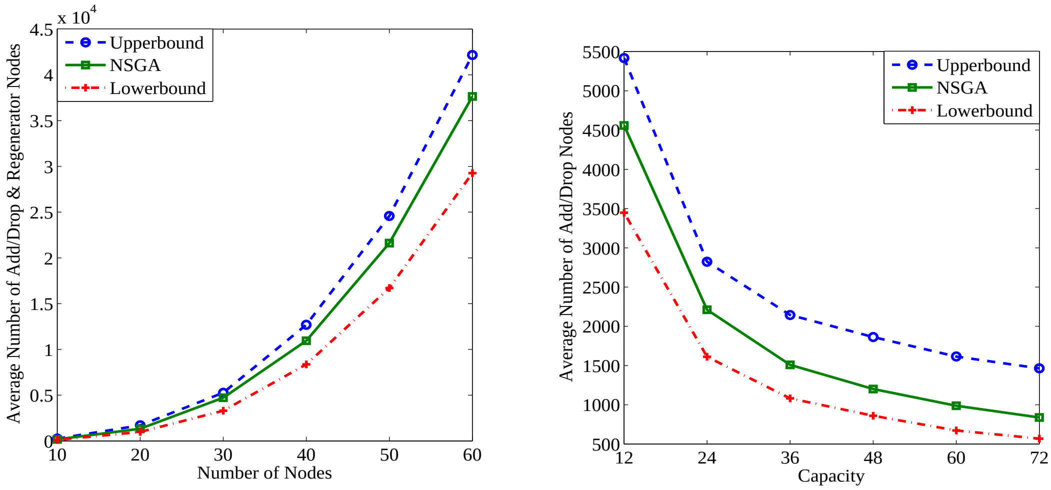
## 7.7 Conclusions

In this chapter, we have studied the survivable impairment-aware traffic grooming problem in WDM wavelength rings, where the objective is to minimize the total cost of grooming and regeneration. Unlike previous studies in traffic grooming, we consider both survivability and impairment-aware routing, which are gaining a lot of interest from both network operators and researchers. We have shown that the problem is NP-hard. We have considered two cases of the problem, (1) when the impairment threshold can be ignored, and (2) when the impairment threshold should be taken into account




(a) (b)

Figure 7.4: Comparison of the number of add/drop nodes required by $NSGA$ in the survivable traffic grooming problem for (a) different number of nodes ($C = 36$ and demand values are within the range $[0, C]$), and (b) different capacity ($N = 30$ and demand values are within the range $[0, 12]$).




(a) (b)

Figure 7.5: Comparison of the number of add/drop nodes required by the modified $NSGA$ in the survivable impairment-aware traffic grooming problem for (a) different number of nodes ($C = 36$ and demand values are within the range $[0, C]$), and (b) different capacity ($N = 30$ and demand values are within the range $[0, 12]$).

under uniform and non-uniform traffic scenarios.

For the survivable traffic grooming problem, we have given a 4-approximation algorithm for uniform traffic, and a variable-factor approximation algorithm for non-uniform traffic. For the survivable impairment-aware traffic grooming problem, the approximation ratio (i.e., worst-case performance) is 16 or 20 depending on the problem instance under uniform traffic. The approximation ratio may be too high for practical purposes. Therefore, we proposed a scheme to improve the average performance of the approximation algorithm, while the worst-case ratio is maintained. Similarly, we provided a heuristic algorithm with a provably tight upper-bound for non-uniform traffic.

# Chapter 8

# Inter-domain Routing in Optical Networks

## 8.1 Introduction

In the previous chapters, we have focused only on networks within the same domain, i.e., intra-domain routing and wavelength assignment (RWA). However, optical networks using wavelength-division multiplexing (WDM) technology are being widely deployed across domains. Future optical networks will require new protocols in order to route and support on-demand provisioning of lightpaths between different domains. Unlike traditional IP multi-domain networks, the study of optical multi-domain issues is at a very early stage. One important issue is what type of information should be exchanged among neighboring domains in order to increase efficiency. Previous works [113][114], have proposed approaches where neighboring domains are able to exchange both *Network Reachability Information* (NRI), and highly aggregated *Path State Information* (PSI). However, the presence of wavelength converters is not analyzed in these works. Our main contribution in this chapter is to seamlessly incorporate modifications to the protocols proposed in [113] and [114], so that wavelength converters are utilized.

In WDM optical networks without wavelength converters, a lightpath has to use the same wavelength all along its path. This implies that lightpath requests may be blocked, even though there are unused wavelengths. In order to decrease the blocking ratio, wavelength converters are employed. Moreover, the optical signal can be regenerated at converter nodes to extend its reach. There are different methods for sharing a pool of wavelength converters at a given node among the wavelengths of its different fiber links [64]. Due to its sharing efficiency, we assume a share-per-node approach, where there is a single bank of converters at a given switching node shared by all its links, and only wavelengths that need to be converted are directed to this bank.

Since wavelength converters are costly (yet usually more affordable than adding fibers in already existing networks), we assume that for inter-domain traffic in a given domain, the wavelength converters are placed at border optical cross-connects (OXCs). This assumption is a realistic representation of emergent multi-domain optical networks [68]. Due to the large amount of traffic that goes through border OXCs, putting wavelength converters at the border OXCs is expected to have a significant performance improvement.

## 8.2 Related Work

In the literature, there are only few works dealing with optical multi-domain networks; there are even fewer works that study the effect of wavelength converters. The three relevant standardization bodies, namely, the International Telecommunications Union (ITU), the Internet Engineering Task Force (IETF), and the Optical Internetworking Forum (OIF) have analyzed some of the topics related to multi-domain optical networks. In 2002, the OIF proposed the Domain-to-Domain Routing Protocol (DDRP). The drawbacks of DDRP are that it represents a major change in the routing system and it is not suitable for path protection. The IETF has proposed the generalized multi-protocol label switching (GMPLS) framework, which extends the features of multi-protocol label switching (MPLS) for provisioning circuit-switched connections via label abstractions for wavelengths, timeslots, etc. The ITU-T has specified a broad-based automatic switched optical network (ASON) framework. However, most of the research surrounding GMPLS and ASON is limited to intra-domain routing.

OBGP (Optical BGP) is an extension of BGP that has been proposed to "glue" multi-domain optical networks [17][37][107]. The strength of this approach is that future optical networks will benefit from the advantages of the BGP-based routing model, such as scalability, clear administrative limits of routing domains, etc. However, besides inheriting the well-known disadvantages of BGP, a multi-domain routing model mainly based on the exchange of network reachability information, which is currently the case in BGP, may not be sufficient. This has initiated the proposal of different path state aggregation schemes and updating policies at the inter-domain level for WDM optical networks [67][113][114].

In [113], the authors showed that by integrating only plain and highly aggregated PSI in OBGP (in the form of an extended protocol called OBGP+), it is possible to drastically improve its performance, without increasing the number or the frequency of routing updates exchanged between domains. In [114], a novel distributed route control model is proposed, which is based on the deployment of inter-domain routing agents (IDRAs). We refer to the routing protocol running among the IDRAs as an IDRAs-based routing protocol (IDRP). IDRP is able to significantly reduce the blocking ratio compared to that of OBGP. However, mechanisms to take advantage of the presence of wavelength converters in these protocols were not developed.

In this chapter, we make simple but important modifications that will allow OBGP+ and IDRP to benefit from the use of wavelength converters. The modifications are simple in that the algorithmic details of these protocols are not affected, and they are important because a significant reduction in the blocking can be achieved due to these modifications. We also show the performance gain obtained by having wavelength converters at border OXCs, and compare the performances of OBGP, OBGP+, and IDRP in the presence of wavelength converters.

In Section 8.3, we give a brief description of OBGP+ and IDRP. In Section 8.4, we show how these protocols can be modified to take into account the presence of wavelength converters. In Section 8.5, we present simulation results comparing the performance of the three protocols and also the improvement associated with having wavelength converters at the border OXCs. Finally, we give conclusions in Section 8.6.

## 8.3 OBGP+ and IDRP

The major advantage of our approach is that our modifications can be seamlessly integrated in OBGP+ and IDRP. In other words, the algorithmic details of these protocols can be reused since our modifications concern only the wavelength aggregation process. For completeness and in order to introduce the notation used in Section 8.4, we give a brief introduction to OBGP+ and IDRP. For a detailed description of these protocols, the reader is referred to [113] and [114].

OBGP+ is an improved version of OBGP in that PSI is advertised besides the usual NRI exchanged in OBGP; whereas IDRP is a novel optical routing protocol that allows the exchange of useful traffic engineering (TE) information.

### 8.3.1 Network Reachability Information (NRI)

NRI messages are triggered when a new destination becomes available, or an already known one becomes unreachable. The reachability information contained in the NRI messages conveyed by OBGP+ consists of:

1. The set of destination networks $\{d\}$ and their associated autonomous system (AS)-path.
2. The Next-Hop (NH) to reach those destinations, i.e., the address of the ingress OXC in the neighboring domain from which the advertisement was sent.
3. A set of pairs $(\lambda_i, W(\lambda_i))$ available for each destination $d$, where $\lambda_i$ denotes a particular wavelength, and $W(\lambda_i)$ denotes the maximum multiplicity of $\lambda_i$.

Unlike BGP/OBGP, the NRI exchanged among the IDRAs does not include the AS-path to reach a destination. In IDRP, rather than comparing candidate routes according to the length of the AS-path, the IDRAs use the TE information contained in the routing advertisements.

### 8.3.2 Aggregated Path State Information (PSI)

At a given OXC, PSI messages aggregate (i) intra-domain PSI; (ii) PSI related to the inter-domain links towards its downstream domains; and (iii) the already aggregated PSI contained in the inter-domain advertisements received from downstream domains. In OBGP+, the PSI is basically composed of aggregated wavelength availability information. In IDRP, the PSI is not only composed of aggregated wavelength availability information, but it also contains aggregated load information, which is represented by associating a cost with each candidate (path, wavelength) pair [114]. For notation purposes, we describe how the aggregated wavelength availability is computed.

The aggregated wavelength availability information is obtained by computing the *Effective Number of Available Wavelengths* (ENAW) for each type of wavelength, both inside an AS and across ASs. Inside an AS, the aggregation process is as follows. Let $u$ and $v$ be a pair of OXCs inside an AS, $P(u, v)$ be a candidate path between $u$ and $v$, and $l$ be a link within the path $P(u, v)$. The ENAW of wavelength type $\lambda_i$ between the OXCs $u$ and $v$ within a domain is computed as follows:

$$W_{u,v}(\lambda_i) = \max_{P(u,v)} \left\{ \min_{l \in P(u,v)} [W_l(\lambda_i)] \right\} \tag{8.1}$$

The rationale behind Equation (8.1) is that the ENAW of a wavelength $\lambda_i$ along a path $P$, which is basically the number of lightpaths that can possibly be setup on $P$ using $\lambda_i$, is determined by the value of $\lambda_i$ at the bottleneck link, i.e., the link with the minimum number of $\lambda_i$ along $P$. Among all the paths between $u$ and $v$, the path with the largest ENAW is chosen.

The inter-domain part is composed of the unused wavelengths on the directly-connected inter-domain links of the OXC, and wavelengths that are available downstream, which are known through the PSI advertisements from neighboring OXCs. Let $W_{l_b,l'_b}(\lambda_i)$ be the ENAW of type $\lambda_i$ between OXC $l_b$ and a local border OXC $l'_b$, $W_{l'_b,r_b}(\lambda_i)$ be the number of free wavelengths of type $\lambda_i$ in the inter-domain link between the local border OXC $l'_b$ and a remote border OXC $r_b$; and $W^{adv}_{r_b,d}(\lambda_i)$ be the ENAW of type $\lambda_i$ between the remote border OXC $r_b$ and the destination OXC $d$, which is advertised by $r_b$ or the IDRA of $r_b$. By combining these inter-domain components and Equation (8.1), the OXC advertises to upstream neighbors the ENAW between the local border OXC $l_b$ and the destination OXC $d$ as:

$$W^{adv}_{l_b,d}(\lambda_i) = \min \left\{ W_{l_b,l'_b}(\lambda_i),\, W_{l'_b,r_b}(\lambda_i),\, W^{adv}_{r_b,d}(\lambda_i) \right\} \tag{8.2}$$

## 8.4 Wavelength Aggregation with Wavelength Converters

In this section, we present one of the main contributions of this chapter, which is the extension of OBGP+ and IDRP to deal with the presence of wavelength converters. Having wavelength converters relaxes the wavelength continuity constraint, thereby increasing the "availability" of wavelengths. We show that with simple but necessary modifications, this information can be incorporated in the wavelength aggregation process. Our approach does not entail too much overhead since the only additional information is the number of wavelength converters at the remote border router.

We identify two types of unoccupied wavelength channels at any given border OXC: *converter* and *non-converter channels*. A converter channel consists of different types of wavelengths on either side of the OXC, thus requiring wavelength conversion if it is to be used for lightpath establishment. A non-converter channel, on the other hand, is made up of the same wavelength on both sides of the OXC and does not require wavelength conversion. In this section, unless explicitly specified, wavelengths/channels refer to *unoccupied* wavelengths/channels.

Since wavelength converters are scarce, it is assumed that they are used only when absolutely necessary. Therefore, we first compute the number and type of non-converter channels the same way as in the case where there are no converters. Then, the remaining wavelengths on either side of the OXC are candidates of converter channels. However, since a single wavelength converter can translate only one input wavelength to another output wavelength, the number of unused wavelength converters also affects the possible number of converter channels. Usually, there are more candidate wavelengths than the possible number of converter channels. Hence, there should be a mechanism to pick a specific wavelength for each converter channel (e.g., first-fit, random-fit, etc.) before being advertised upstream. This approach provides a highly aggregated state information, while capturing the availability of wavelength channels.

We now explain how the ENAW is computed using Figure 8.1, which shows an example network with two ASs, their border OXCs and the unoccupied wavelengths at each OXC. For AS1, $l_b$ and $l'_b$ represent its border nodes, whereas $r_b$ is the node that is directly connected to AS1. The downstream AS (in this case AS2) advertises a set of available wavelengths to the upstream AS (in this case AS1). Let $W^{adv}_{r_b,d}(\lambda_i)$ be the advertised number of wavelengths of type $\lambda_i$ from the downstream AS. Also, let $R^{adv} = R_{r_b}$ be the advertised number of available converters at $r_b$. $W_{l'_b,r_b}(\lambda_i)$ is the number of wavelengths of type $\lambda_i$ on the link between $l'_b$ and $r_b$. This value is known to $l'_b$ since the link is physically attached to it.

Thus, the number of non-converter channels of type $\lambda_i$ at $l'_b$ is:

$$W_{l'_b,d}(\lambda_i) = \min\left\{W_{l'_b,r_b}(\lambda_i), W^{adv}_{r_b,d}(\lambda_i)\right\} \tag{8.3}$$

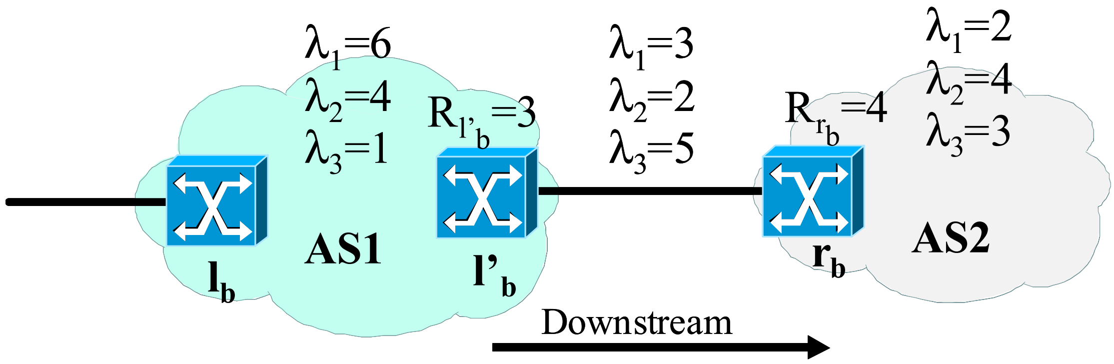



Figure 8.1: An example depicting border OXCs of two domains connected by a single inter-domain link. This example shows the number of wavelengths and wavelength converters available at border OXCs of the two domains.

In Figure 8.1, $W_{l'_b,d}(\lambda_1) = \min\{3,2\} = 2$, $W_{l'_b,d}(\lambda_2) = \min\{2,4\} = 2$, and $W_{l'_b,d}(\lambda_3) = \min\{5,3\} = 3$.

The remaining wavelengths can be part of converter channels at $l'_b$. The maximum number of possible converter channels is determined not only by the number of wavelengths that are *not* in the non-converter channels, but also by the number of available converters. Hence, it can be shown that the maximum number of converter channels is,

$$\min \left\{ \left[ \sum_i \left( W_{l'_b,r_b}(\lambda_i) - W_{l'_b,d}(\lambda_i) \right) \right], \left[ \sum_i \left( W^{adv}_{r_b,d}(\lambda_i) - W_{l'_b,d}(\lambda_i) \right) \right], R^{adv} \right\} \tag{8.4}$$

In Figure 8.1, the number of converter channels is: $\min\{\{(3-2)+(2-2)+(5-3)\}, \{(2-2)+(4-2)+(3-3)\}, 4\} = \min\{3,2,4\} = 2$.

For these converter channels, wavelengths are selected from the set $\{W_{l'_b,r_b}(\lambda_i)\} \backslash \{W_{l'_b,d}(\lambda_i)\}$, i.e., the set of wavelengths in $W_{l'_b,r_b}(\lambda_i)$ that are not in the non-converter channels. Then, $W_{l'_b,d}(\lambda_i)$ is updated so that it includes both the converter and non-converter channels before being advertised upstream. Let us assume that a random selection is used and the updated $W_{l'_b,d}(\lambda_1) = 3$, $W_{l'_b,d}(\lambda_2) = 2$, and $W_{l'_b,d}(\lambda_3) = 4$.

Similarly, the number of non-converter channels of type $\lambda_i$ at $l_b$ is:

$$W_{l_b,d}(\lambda_i) = \min \left\{ W_{l_b,l'_b}(\lambda_i), W_{l'_b,d}(\lambda_i) \right\} \tag{8.5}$$

In Figure 8.1, $W_{l_b,d}(\lambda_1) = \min\{6,3\} = 3$, $W_{l_b,d}(\lambda_2) = \min\{4,2\} = 2$, and $W_{l_b,d}(\lambda_3) = \min\{1,4\} = 1$.

The total number of converter channels at $l_b$ is,

$$\min\left\{\left[\sum_i \left(W_{l_b,l_b'}(\lambda_i) - W_{l_b,d}(\lambda_i)\right)\right], \left[\sum_i \left(W_{l_b',d}(\lambda_i) - W_{l_b,d}(\lambda_i)\right)\right], R_{l_b'}\right\} \tag{8.6}$$

where $R_{l_b'}$ is the number of converters at $l_b'$. In Figure 8.1, this is equal to $\min\{\{(6-3)+(4-2)+(1-1)\}, \{(3-3)+(2-2)+(4-1)\}, 3\} = \min\{5,3,3\} = 3$. Let us assume that after randomly selecting from wavelengths that are not in the non-converter channels for the three converter channels, the updated $W_{l_b,d}^{adv}(\lambda_1) = 5$, $W_{l_b,d}^{adv}(\lambda_2) = 3$, and $W_{l_b,d}^{adv}(\lambda_3) = 1$.

Finally, AS1 advertises $W_{l_b,d}^{adv}(\lambda_i)$ and $R^{adv} = R_{l_b}$ to upstream domains. However, without the modified wavelength aggregation process (see Equation (8.2)), AS1 would have instead advertised $W_{l_b,d}^{adv}(\lambda_1) = 2$, $W_{l_b,d}^{adv}(\lambda_2) = 2$, and $W_{l_b,d}^{adv}(\lambda_3) = 1$.

In [113][114], it is proposed to piggyback *Keepalive* messages that are exchanged between neighboring OXCs with PSI messages. In this approach, keepalive messages are, just like in BGP, exchanged to notify if the neighboring node is still operative. However, unlike in BGP, the keepalive messages are extended to convey PSI messages. A major advantage of this strategy is that it does not increase the number of routing messages exchanged between domains. In this chapter, we employ the same approach.

## 8.5 Results and Discussion

In this section, we present simulation results that compare the performance of OBGP, OBGP+ and IDRP. Our performance metrics are the *Blocking Ratio* (BR) of inter-domain lightpath requests, and the number of routing messages exchanged to achieve this blocking ratio. To this end, we have conducted extensive simulations using OPNET. In our simulations, we have used a PAN-European topology, which was introduced in [31] (and shown in Figure 8.2) as a reference topology suitable for a PAN-European fiber-optic network. The network consists of 28 domains and 41 inter-domain links, and the nodes were chosen in such a way that some of the main European Internet Exchange Points are included.

Inside each domain of the PAN European network, we placed a random number of OXCs, which is equal to or higher than the number of inter-domain links of that domain. There are 18 source and 10 destination OXCs randomly located covering the entire PAN European network in such a way that each domain has one source or destination OXC. In other words, we simulate inter-domain traffic which is transferred between domains. Each link in the network consists of 5 fibers and each fiber has 14 wavelengths.

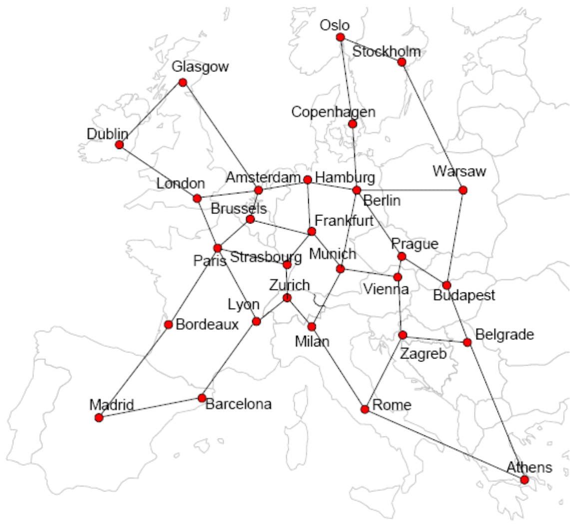



Figure 8.2: A PAN-European network.

In our simulation, traffic was modeled according to a Poisson distribution with exponentially distributed inter-arrivals. The blocking ratio and routing messages are collected under different traffic loads, varying from 100 up to 300 Erlangs. In order to evaluate the impact of the frequency of updates in the PSI messages, we have tested three scaled and normalized *Keepalive Update Interval* ($K_T$) of the Keepalive messages: $K_T = 1$, $K_T = 3$, and $K_T = 5$ units. In terms of the availability of converters, we have considered three scenarios: no converters, 5 converters and 10 converters at each border OXC of the domains in the network. For each case, the results are the averages of over 30 randomly generated PAN European network configurations. These network configurations are different from each other in the network topology inside each domain, and the location of source and destination OXCs over the entire network.

Due to space constraints, we are able to show only some of the results. Figures 8.4 and 8.5 show the efficiency of using wavelength converters in OBGP+ and IDRP for $K_T = 1$. Similar results have been obtained for $K_T = 3$ and $K_T = 5$. Figure 8.3 shows the improvement factor (**IF**) in the blocking ratios of OBGP+ and IDRP over OBGP and the number of messages generated under traffic values 200, 250 and 300 Erlangs for 5 converters. Similar results have been obtained for 10 converters. The following observations can be made from our results.

- Increasing the update interval $K_T$ causes more blocking because a higher value of $K_T$ means that the PSI is not accurate enough since messages are exchanged less frequently. In fact, a major advantage of embedding PSI messages in Keepalive messages is that when $K_T$ is decreased so as to improve the responsiveness of OXC neighbors, PSI messages will be updated more frequently.

- IDRP always significantly outperforms both OBGP+ and OBGP (whereas OBGP+ outperforms OBGP). This is due to the fact that IDRP additionally utilizes aggregated load information. In fact, for 10 converters and $K_T = 1$, IDRP achieves a blocking ratio of less than 0.1% for all simulated traffic values. The 0.1% blocking ratio is a threshold recommended by the IST FP6 NOBEL project [76] for optical networks in order to support real-time and streaming applications.

- The total number of messages generated decreases as more wavelength converters are used in the network. The reason for this is that in the presence of wavelength converters, the wavelength continuity constraint is relaxed and there will be more wavelengths available along a path. Therefore, it is less likely for the wavelengths of a path to be exhausted fast, thereby triggering reachability messages and path exploration.

- The blocking ratios for IDRP and OBGP+ decrease as more wavelength converters are placed in the network. But this is not the case in OBGP (results not shown here) if it always chooses the wavelength with the lowest identifier (First-Fit) along the shortest path. Such a first-fit approach increases conflicts as different OXCs tend to simultaneously choose lower identifier wavelengths, while higher identifier wavelengths are available. The situation is worsened as the number of converters in the network is increased, since the "availability" of these lowest indexed wavelengths is also increased, thereby exacerbating the possibility of conflicts. This situation can be avoided by choosing wavelengths randomly (Random-Fit) instead of always choosing lower indexed wavelengths.

## 8.6 Conclusions

In this chapter, we have made simple but important modifications to two inter-domain optical protocols, namely, OBGP+ and IDRP, to handle the presence of wavelength converters. We have also performed extensive simulations comparing the performance of OBGP (Optical BGP) and these protocols. The results obtained in a PAN European network show that IDRP significantly outperforms OBGP+ and OBGP, and OBGP+ outperforms OBGP. The performance metrics in the simulation were blocking ratio and the number of messages generated (for a duration of one week).

From these results, it can be inferred that the exchange of aggregated path state information (PSI), and the presence of wavelength converters at border OXCs improve the blocking ratio and the number of messages generated significantly. In fact, using

| | Keepalive Update Interval ($K_T$=1) | | | Keepalive Update Interval ($K_T$=1) | | | Keepalive Update Interval ($K_T$=5) | | |
|---|---|---|---|---|---|---|---|---|---|
| | 200 Erlangs | 250 Erlangs | 300 Erlangs | 200 Erlangs | 250 Erlangs | 300 Erlangs | 200 Erlangs | 250 Erlangs | 300 Erlangs |
| **IF (OBGP+)** | 3673.38 | 68.50 | 10.97 | 572.01 | 50.79 | 10.30 | 315.5 | 38.55 | 9.69 |
| **IF (IDRP)** | 13395.80 | 172.95 | 25.14 | 827.13 | 103.67 | 18.42 | 518.14 | 69.97 | 15.69 |
| **Traffic (Erlangs)** | Routing Messages OBGP | Routing Messages OBGP+ | Routing Messages IDRP | Routing Messages OBGP | Routing Messages OBGP+ | Routing Messages IDRP | Routing Messages OBGP | Routing Messages OBGP+ | Routing Messages IDRP |
| 100 | 7,393,754 | 4,106,295 | 3,316,670 | 6,538,899 | 4,045,308 | 3,262,196 | 5,810,185 | 3,996,540 | 3,204,790 |
| 150 | 8,433,843 | 3,999,754 | 3,362,538 | 7,066,636 | 3,920,216 | 3,278,214 | 6,113,651 | 3,830,622 | 3,195,828 |
| 200 | 9,139,267 | 4,002,240 | 3,377,721 | 7,593,317 | 3,881,279 | 3,260,521 | 6,381,400 | 3,775,545 | 3,149,897 |
| 250 | 9,149,884 | 4,025,679 | 3,378,519 | 7,410,155 | 3,916,994 | 3,239,177 | 6,323,175 | 3,823,118 | 3,110,352 |
| 300 | 9,420,468 | 4,771,478 | 3,455,367 | 7,433,076 | 4,469,614 | 3,279,149 | 6,254,482 | 4,252,166 | 3,121,277 |

Figure 8.3: Improvement Factors (**IF**) in the blocking ratios of OBGP+ and IDRP over OBGP for 200, 250, and 300 Erlangs, and overall number of routing messages exchanged for 5 converters.

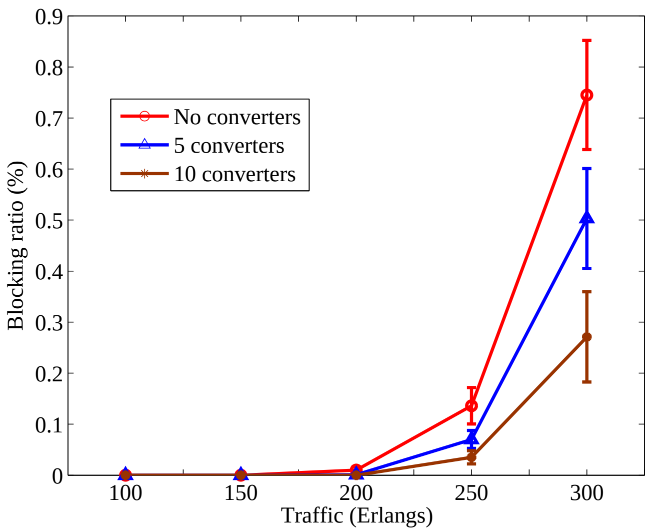



Figure 8.4: Average blocking ratio and standard deviation. Comparison of different number of wavelength converters for OBGP+ ($K_T = 1$).

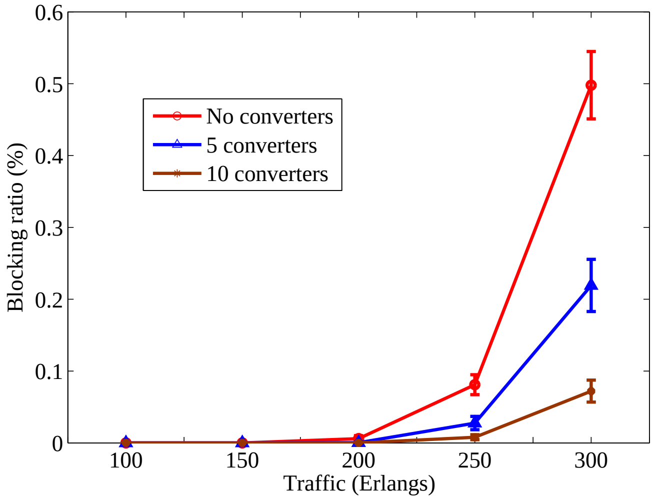



Figure 8.5: Average blocking ratio and standard deviation. Comparison of different number of wavelength converters for IDRP ($K_T = 1$).

IDRP with enough wavelength converters, it is possible to achieve the 0.1% blocking ratio threshold that is recommended by the IST FP6 NOBEL project [76] for optical networks to support real-time and streaming applications. The decrease in the blocking ratio is obtained without an increase in the total number of messages exchanged, because we have employed a strategy of piggybacking PSI updates in the Keepalive messages exchanged between neighboring IDRAs/OBGP+ nodes.

# Chapter 9

# Case Study

## 9.1 Introduction

As was discussed in Chapter 7, traffic grooming allows efficient utilization of the large capacity of lightpaths by aggregating several independent traffic streams onto a single wavelength. In addition, in the course of this thesis, we have emphasized the importance of survivability (Chapters 2-7) and impairment-aware routing (Chapters 4-7) in WDM networks. In Chapter 7, we considered survivable impairment-aware traffic grooming in WDM rings. In this chapter, we consider the same problem but with a special focus on the SURFnet6[1] network shown in Figure 9.1 (the approaches followed in this chapter can be employed for other WDM networks as well). We also make use of realistic data (types of nodes, impairment values, traffic matrix, etc.) obtained from the SURFnet6 network.

Nodes are equipped with an optical add/drop multiplexer (OADM) to add/drop wavelengths. There can be two types of OADMs in the network: fixed OADMs and Reconfigurable OADMs (ROADMs) [3]. In fixed OADMs, wavelengths that are added/dropped at a given node are fixed and reconfiguration requires human intervention. On the other hand, in ROADMs, one or more wavelengths can be added/dropped automatically with minimal user intervention. The key enabling component in ROADM configuration is the Wavelength Selective Switch (WSS), which allows for individual wavelengths on a common input fiber to be selectively switched to any of multiple output fibers [48]. Even though ROADMs are flexible and efficient, the initial cost of ROADM components is higher than that of fixed OADM components. Thus, not all nodes may be equipped with ROADMs. Amplifiers at nodes, which are required to compensate for transmission fiber loss and the loss of passive optical components, add noise and contribute to signal distortions. The impairment value associated with a node

[1]The SURFnet6 network connects research and educational institutes in the Netherlands using lightpaths (http://www.surfnet.nl/en).

depends on the type of OADM used at the node. Unlike most previous studies, we not only consider impairments associated with links, but also nodal impairments in this chapter. Similarly to the model in Chapter 7, the cost of transceivers is the dominant component in the cost of traffic grooming and regeneration.

In Section 9.2, we give a formal definition of the survivable impairment-aware traffic grooming problem, which minimizes the total number of transceivers placed in the network, and show that the problem is NP-hard. Subsequently, we provide a heuristic approach for solving the problem in Section 9.3. We study the performance of this approach in Section 9.4 using data obtained from the SURFnet6 network, and we finally conclude in Section 9.5.

## 9.2 Problem Definition

We use the same terms used in Chapter 7, and the objective remains to minimize the total number of transceivers required to feasibly and survivably route all requests (with the assumption that there are enough wavelengths to accommodate all the requests). Here also, as in Chapter 7, traffic between a given pair of nodes is not split unless it exceeds the full capacity of a wavelength. Since a wavelength is regenerated when traffic is added/dropped from it, an add/drop node is also a regenerator node. Thus, a wavelength can be regenerated under two scenarios: (1) when regeneration is required so that the impairment threshold is not exceeded, and (2) when traffic carried by the wavelength is added/dropped. We refer to the former as simply *regeneration*, while to the latter as *add/drop regeneration*.

**Problem 9.1** ***Survivable impairment-aware traffic grooming problem*** *Given is an optical network $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes, $\mathcal{L}$ is the set of links. Associated with each link $l \in \mathcal{L}$ is an impairment value $r(l)$, and with each node $u \in \mathcal{N}$ is an impairment value $r(u)$. The impairment threshold is $\Delta$ and a wavelength has a capacity $C$. Also given are a set $\mathcal{F}$ of $F$ requests. Each request $f \in \mathcal{F}$ is represented by the tuple $(s_f, d_f, \delta_f)$, where $s_f$ and $d_f$ are the source and destination nodes, respectively, and $\delta_f$ is the amount of demand of request $f$. The problem is to minimize the total number of transceivers needed in the network so that (1) each request is assigned a pair of disjoint lightpaths; (2) the capacity of each wavelength in a link is not exceeded; and (3) for any lightpath, the impairment values of its regeneration segments should not exceed the impairment threshold.*

**Theorem 9.1** *The survivable impairment-aware traffic grooming problem is NP-hard.*

**Proof.** We provide a proof based on the (survivable) *impairment-aware path selection* problem, where given a sparse regeneration network (i.e., a network wherein only a few nodes have regeneration capacity), an impairment threshold, and a request between two pairs of nodes, the problem is to find a feasible simple path for the given request, i.e., its regeneration segments do not violate the impairment threshold. The (survivable) impairment-aware path selection problem is proved to be NP-hard in Chapter 4. As is explained earlier, we have two scenarios that lead to the regeneration of a given wavelength: add/drop regeneration (i.e., when traffic is added/dropped from the wavelength), and regeneration (i.e., when the impairment value since last regeneration exceeds the threshold).

*Instance*: A single wavelength and a set of requests that can all fit in this wavelength. In this instance, the number of add/drop nodes is fixed, i.e., the total number of distinct source and destination nodes. Hence, the objective reduces to minimizing the number of extra regenerations. A decision problem related to the given instance of the survivable impairment-aware traffic grooming problem is described as follows:

*Question*: Is it possible to feasibly route all requests with at most $K$ extra regenerations?

For $K = 0$, the question reduces to: is it possible to feasibly route each request using only add/drop regenerations? In this scenario, the source and destination nodes of the given requests are the only (add/drop) regeneration nodes. In other words, the network has sparse regeneration capacity. By solving the decision problem, each request will be assigned feasible primary and backup lightpaths using only the existing regeneration capacity in the network. However, this is equivalent to solving the NP-hard survivable impairment-aware path selection problem for each request. ■

## 9.3 Heuristic Approach

Since the survivable impairment-aware traffic grooming problem is NP-hard, we propose a heuristic approach for solving it. The complexity of the problem can be reduced by limiting the number of paths that are considered for each request. However, since it is a design problem, time is of less importance since the algorithm needs to run once. We provide a two-phase heuristic approach that makes use of a precomputed set of paths.

### 9.3.1 Phase 1: Precomputed Paths

In the first phase, $K$ pairs of (shortest) disjoint paths are pre-computed for each request (using an algorithm given in [82]), and the solution will be selected from these pairs of paths using an Integer Linear Programming (ILP) formulation. In this phase, the objective is to minimize the number of transceivers required for adding/dropping the wavelengths (i.e., regeneration is not considered). It is based on the assumption that putting requests that originate or end at a given node in the same wavelength minimizes the total number of transceivers needed. This approach has also an additional advantage in that the total number of used wavelengths is minimized, since it tends to aggregate (groom) traffic in a smaller number of wavelengths. Let $W$ be the number of wavelengths, and all the requests can be accommodated. We assume that each request has a demand that is less than the capacity of a wavelength, otherwise a full wavelength is independently assigned for the request and the remaining amount is considered as its demand.

**Indices, constants, variables:**

| | |
|---|---|
| $P_{f,k} = \{P_{f,k,1}, P_{f,k,2}\}$ for $k = 1, \ldots, K$ | A set of precomputed pairs of disjoint paths for request $f$. |
| $\alpha_{f,k,\lambda}$ | is 1 if the $k^{th}$ disjoint path pair is selected and uses wavelength $\lambda$; 0 otherwise. |
| $a_{f,k,l}$ | is 1 if either the primary or the backup path of the $k^{th}$ disjoint path pair uses link $l$; 0 otherwise. |

**Objective:**

Minimize the total number of add/drop nodes:

$$\text{Minimize:} \qquad \sum_u \sum_\lambda x_{u,\lambda}.$$

**Constraints:**

For each request, only one pair of disjoint paths is selected:

$$\sum_k \sum_\lambda \alpha_{f,k,\lambda} = 1 \quad \text{for } f = 1, \ldots, F.$$

The capacity of each wavelength on each link should not be exceeded:

$$\sum_f \sum_k \delta_f \cdot a_{f,k,l} \cdot \alpha_{f,k,\lambda} \leq C \qquad \text{for } \forall l \in \mathcal{L}; \lambda = 1, \ldots, W.$$

A given node is an add/drop node for a given wavelength if traffic is added/dropped (groomed) on this wavelength at the node:

$$\sum_{f \in \{f | s_f = u \text{ or } d_f = u\}} \sum_k \alpha_{f,k,\lambda} \leq F \cdot x_{u,\lambda} \; \forall u \in \mathcal{N}; \lambda \; = 1, ..., W.$$

### 9.3.2 Phase 2: Rerouting Lightpaths

In phase 1, the objective is to reduce the number of transceivers needed for adding/dropping (grooming) the given set of requests at the source and destination nodes. However, some of the lightpaths obtained in phase 1 may not be feasible, thus require the placement of extra transceivers. Algorithm *Reroute* (see Algorithm 9.1) minimizes the additional number of regenerations by rerouting requests whose lightpaths are infeasible. Unlike the wavelength rerouting discussed in Chapter 3, the rerouting in this section changes both the path and the wavelength of a lightpath. A request needs extra regeneration if its primary or backup lightpath has an infeasible segment (i.e., its impairment value exceeds $\Delta$) in the current setup. Let $\mathcal{P}$ be the set of requests that need extra regeneration, and $\mathcal{N}_\lambda$ be the set of regenerator nodes for wavelength $\lambda$ in the given network.

Algorithm *Reroute* works as follows. In Step 1, it (randomly) chooses a request $f$ from the requests in $\mathcal{P}$. In the next steps, it tries to find a feasible pair of disjoint lightpaths using only the existing regenerator nodes. This is done by constructing a new graph on each wavelength that $s_f$ and $d_f$ are add/drop nodes. In Step 2, graph $G_\lambda$ represents a graph on wavelength $\lambda$, and is made up of links with enough residual capacity on wavelength $\lambda$ to support request $f$ or belong to the primary or backup lightpaths of request $f$. In Step 2$b$, a new graph $G'_\lambda$ is obtained from graph $G_\lambda$ as follows. Its nodes are the add/drop or regenerator nodes of wavelength $\lambda$ (including the source and destination nodes of request $f$), and a link exists between two nodes if they are directly reachable (i.e., without regeneration). Then in Step 2$c$, two disjoint paths are computed using Suurballe's algorithm [99] in graph $G'_\lambda$. These paths are then translated to their equivalent paths in $G_\lambda$ by replacing the links in $G'_\lambda$ with the corresponding subpaths in $G_\lambda$. If the paths are simple and feasible, they are accepted as a solution. Otherwise, we add extra regenerator nodes to make the original lightpaths of $f$ feasible. Adding extra regenerator nodes, however, may render some of the requests in $\mathcal{P}$ feasible. These requests are removed from $\mathcal{P}$ before the next iteration.

## 9.4 Simulation Results

In this section, we compare our heuristic approach with a sequential approach in the SURFnet6 network shown in Figure 9.1. In the sequential approach, each request is assigned the shortest link-disjoint pair of paths between its source and destination nodes. Then, the lightpaths are sequentially allocated wavelengths in such a way that a lightpath is assigned to the lowest-indexed wavelength that has sufficient capacity for its traffic. We first provide a description of the physical impairment considered in these simulations.

### 9.4.1 Figure of Merit (FoM)

Amplifiers are placed at several points along a fiber-link to overcome fiber losses. The segment of a link between two consecutive amplifiers is known as a *fiber span*. However, each amplifier adds noise, which is referred to as *Amplifier Spontaneous Emission (ASE)*, along the fiber. ASE degrades the optical signal to noise ratio (OSNR) and is an important physical impairment, especially when the power levels are low enough to ignore non-linearities [94]. The noise figure of a link, which is the ratio of the OSNR

**Algorithm 9.1** *Reroute*

1. While $\mathcal{P}$ is not empty, pick a request $f \in \mathcal{P}$. Let its assigned disjoint pair of lightpaths be $\{P_{f,1}, P_{f,2}\}$.
2. For each wavelength $\lambda$ for which $s_f$ and $d_f$ are add/drop nodes, let $B'_{l,\lambda}$ be the residual capacity of wavelength $\lambda$ on link $l$. Let $G_\lambda = (\mathcal{N}, \mathcal{L}_\lambda)$, where $\mathcal{L}_\lambda = \{l \in \mathcal{L} | B'_{l,\lambda} \geq \delta_f$ or $l \in P_{f,1}$ or $l \in P_{f,2}\}$. Let $\mathcal{N}'_\lambda$ be the set of nodes on which $\lambda$ is add/dropped or regenerated.
   (a) For any $u, v \in \mathcal{N}'_\lambda$, let $r_\lambda(P_{u-v})$ be the length of the shortest path (in terms of impairment values) between nodes $u$ and $v$ in $G_\lambda$.
   (b) Create graph $G'_\lambda = (\mathcal{N}'_\lambda, \mathcal{L}'_\lambda)$, where $\mathcal{L}'_\lambda = \{(u, v) | u, v \in \mathcal{N}'_\lambda$ and $r_\lambda(P_{u-v}) \leq \Delta\}$. Assign a cost of 1 to each link in $G'_\lambda$.
   (c) Find two disjoint paths $P'_1$ and $P'_2$ in graph $G'_\lambda$.
   (d) For $P'_1$ and $P'_2$, find their corresponding paths $P_1$ and $P_2$ in $G_\lambda$.
   (e) If $P_1$ and $P_2$ are simple and disjoint lightpaths:
      i. Assign them to request $f$.
      ii. Remove $f$ from $\mathcal{P}$ and update the residual capacities of all links that belong to the old and new lightpaths of $f$.
      iii. Go to Step 1.
   (f) Else, go to Step 2 for the next wavelength.
3. If all wavelengths are exhausted and no feasible lightpaths are found,
   (a) Place the minimum number of regenerators needed to make $P_{f,1}, P_{f,2}$ feasible.
   (b) Remove $f$ from $\mathcal{P}$.
   (c) Remove all requests in $\mathcal{P}$ whose lightpaths are now feasible.
   (d) Go to Step 1.

---

at the start of a link to that at the end of a link, is the sum of the noise figures of its spans. The noise figure of a system is usually given in *dB*. In order to express the noise figure in linear units, we introduce the following formula to quantify the quality of an optical fiber link [94], which we refer to as the *Figure of Merit (FoM)*.

$$FoM = \sum_{j=1}^{H} 10^{\left(\frac{L_j}{10}\right)},$$

where $L_j$ is the fiber loss of span $j$ in $dB$ (it is the same as the gain of amplifier $j$ when the net gain of the amplified link is unity), and $H$ is the number of spans.

### 9.4.2 Results and Discussion

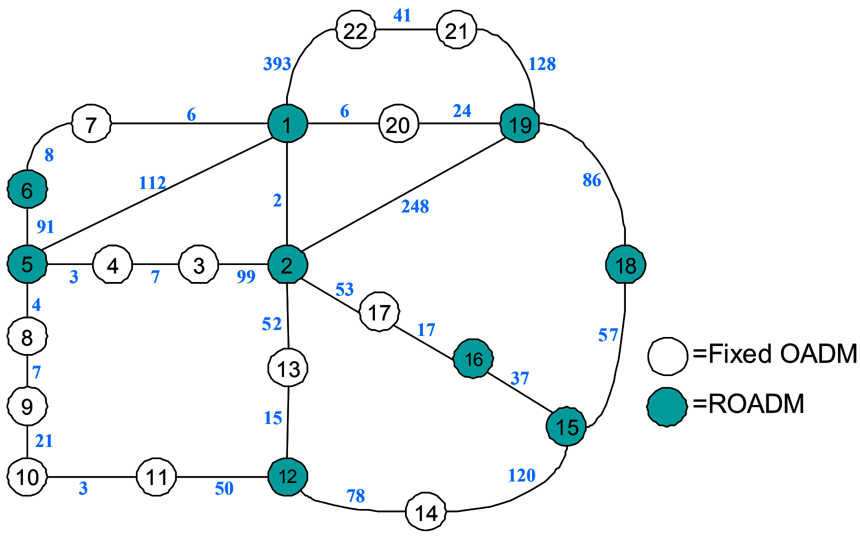



Figure 9.1: The SUFnet6 network. Shown in this figure are the different types of nodes and the FoM values of the links in the network (for fiber loss of 25 $dB/km$).

As shown in Figure 9.1, there are two types of nodes in the network: Fixed OADM and ROADM nodes. The FoM value of a fixed OADM node is 65, while that of a ROADM node is 37. The FoM values of the links are shown in the figure. We have considered five traffic matrices (TMs) that represent synchronous digital hierarchy (SDH) data over the WDM network. One of the traffic matrices (TM1) represents a realistic traffic matrix of the SURFnet6 network, while the others represent predicted traffic scenarios of this traffic matrix. The impairment (FoM) threshold for the 10Gb/s WDM interfaces that we considered is equal to 600.

Table 9.1 compares our approach (for $K = 3$ pairs of disjoint paths per request) with the sequential approach in terms of the total number of transceivers and wavelengths required in the network for five different traffic matrices. The results show that both the number of transceivers and wavelengths required by our heuristic approach are significantly less than those of the sequential approach. Additionally, in our approach, all the regenerations for the given traffic matrices are handled using add/drop regenerations, i.e., no extra transceivers are needed for only regeneration. This is achieved through rerouting of the lightpaths in the second phase of our approach. We have developed an exact ILP as well, but it was too complex to return solutions in a reasonable time frame (within a few weeks). We have, therefore, compared our approach only to the sequential approach, which is often used by practitioners.

Table 9.1: Results comparing our approach and a sequential approach.

| Total Amount of Traffic (Gb/s) | | | | | |
|---|---|---|---|---|---|
| | TM1 | TM2 | TM3 | TM4 | TM5 |
| | 106 | 130 | 144 | 162 | 180 |
| Number of Transceivers | | | | | |
| Our Approach | 158 | 170 | 178 | 196 | 206 |
| A Sequential Approach | 232 | 284 | 300 | 294 | 318 |
| Number of Wavelengths | | | | | |
| Our Approach | 5 | 7 | 8 | 9 | 12 |
| A Sequential Approach | 8 | 11 | 12 | 13 | 15 |

## 9.5 Conclusions

In this chapter, we have studied the survivable impairment-aware traffic grooming problem with a special focus on the SURFnet6 network. We have shown that this problem is NP-hard, and provided an efficient heuristic approach for solving it. We have performed simulations on this realistic network (using actual data obtained from the network) comparing our approach with a greedy sequential approach, which is usually used by practitioners. The simulation results have shown that the number of regenerators and wavelengths required by our heuristic approach are significantly less than those of the sequential approach.

Minimizing the number of transceivers will not only lead to a significant reduction in the capital expenditure (CAPEX), but also results in a reduced operational expenditure (OPEX) because of the significant decrease in power consumption and heat dissipation. In addition, the reduced number of wavelengths decreases the operating cost (OPEX) associated with each wavelength.

# Chapter 10

# Conclusions

With the development and growth of bandwidth-intensive applications, there is an ever-increasing demand for bandwidth. Wavelength Division Multiplexing (WDM) optical networks that provide a large bandwidth capacity by partitioning a fiber into multiple, but independent wavelength channels are promising candidates to satisfy the growing bandwidth demand. Each wavelength channel can currently operate at rates as high as 100 Gb/s. In WDM networks, requests between nodes are handled using *lightpaths* that are made up of wavelength channels. In WDM networks, unlike traditional networks, routing involves not only path selection, but also wavelength assignment, and is known as *Routing and Wavelength Assignment (RWA)*. The main focus of this thesis is to study two important issues pertinent to RWA in WDM networks:

(1) Survivability: Lightpaths in WDM networks usually transport a large amount of data. If a lightpath fails due to various natural or man-made disasters, the data loss can be costly. Hence, survivability, which is the ability to reconfigure and resume communication, is vital.

(2) Impairment-aware routing: As an optical signal traverses its path, it encounters noise and signal distortions along its way. The effect of these physical impairments may lead to bit errors, which make the signal unrecognizable at the receiving end. The solution for this is to regenerate the signal at some intermediate points. Impairment-aware routing is a routing where the effect of physical impairments is taken into account.

In this thesis, we began with the study of impairment-agnostic survivability (Chapters 2 and 3), and then dealt with both survivability and impairment-aware routing (Chapters 4-7) in intra-domain networks. We have also considered RWA in inter-domain networks (Chapter 8). In addition, we made a case study (Chapter 9) of survivable and impairment-aware routing on a realistic network using data obtained from the network.

In short, this thesis makes theoretical as well as practical contributions in the study of survivable and impairment-aware RWA in WDM optical networks. The proposed algorithms and approaches may also be extended to other types of networks, since the routing issues discussed in this thesis are related to such issues as QoS routing, faulttolerance, etc. in other types of networks. For example, impairment-aware routing relates to the gas station problem, where given a set of nodes (towns) with gas stations, the objective is to find a route from one town to another in such a way that a driver is not stranded between gas stations. We now give general conclusions of the thesis.

## 10.1 Survivability (Impairment-agnostic)

Survivability usually involves finding link-disjoint primary (which is used under normal operations) and backup (which is used after the failure of the primary lightpath) lightpaths. Minimizing the total cost of the primary and backup lightpaths (min-sum) is the most common objective. The min-sum link-disjoint paths problem is polynomially solvable. However, there can be some secondary objectives in order to satisfy additional requirements such as minimizing the cost of the longer or the shorter path, and bounding the costs of the two paths. We have studied variants of the min-sum problem with these secondary objectives, and our main conclusions are:

- These secondary objectives turn the polynomially-solvable min-sum problem to NP-complete problems.
- Since only min-sum pairs of paths are considered, the search space is reduced. Thus, exact algorithms (which we proposed) can solve the respective problems in a reasonable running time (in the order of seconds) for fairly large network sizes (in the order of hundred of nodes), which makes them possible candidates for practical purposes.

When lightpaths are dynamically setup and torn-down, lightpath requests arrive and depart the network over time. Since there is generally a limited amount of resources, assigning a lightpath may block future lightpath requests. However, network operators should decide whether to accept or reject the decision based on the input seen so far, i.e., without any knowledge of future requests. This is known as on-line routing, as opposed to off-line routing, where the whole input sequence is known *a priori*. The performance of an on-line routing algorithm is measured against that of a corresponding off-line routing algorithm using what is known as the competitive ratio. In this thesis, we have considered (on-line) survivable routing and wavelength assignment (SRWA), and our main conclusions and contributions are:

- For specific topologies such as rings, star-of-rings, tree-of-rings and lattices, it is possible to have good (i.e., constant or logarithmic) competitive ratios. Even though these topologies are relatively simple, not only they provide us insight into the complexity of the on-line SRWA problem, but they are also used in some realistic networks.

- For general topologies, it is not possible to obtain a good competitive ratio. However, wavelength rerouting, which involves changing the wavelength of existing lightpaths to accommodate a newly arriving request, improves the average performance of on-line algorithms. In practice, since it may not be desirable to disrupt too many lightpaths, the number of rerouted lightpaths should be minimized.

- The complexity of the wavelength rerouting problem, which minimizes the total number of rerouted lightpaths, depends on the survivability scheme. It is polynomially solvable for dedicated protection scheme, whereas it is NP-complete for shared protection scheme.

- Combined with sharing of resources among backup lightpaths, wavelength rerouting can significantly reduce the rejection ratio in on-line SRWA.

- Our main contributions are (1) approximation algorithms that perform close to the best-known, but costly (in terms of time and memory requirements) exact algorithm for dedicated protection scheme, and (2) a heuristic algorithm for shared protection scheme.

## 10.2 Impairment-aware Routing

The two main questions with regards to impairment-aware routing are:

(1) Impairment-aware path selection: how to find a feasible path from the source to the destination node?

(2) Regenerators placement: where and how many regenerators need to be placed?

The effect of physical impairments can be well approximated by considering some of the main linear (additive) impairments. Therefore, we considered impairment-aware path selection, where there are a set of $m \geq 1$ such physical impairments associated with each link. The following are our main conclusions in this regard:

- The path selection problem is strongly NP-complete for $m \geq 1$ in general topologies.

- The problem is polynomially solvable for $m = 1$ in line topology and directed acyclic topologies or for such topologies as tree-of-rings, which can be transformed to directed acyclic graphs (DAGs) before solving the problem.

- Our main contributions are: (1) An efficient heuristic algorithm for general topologies, which is derived from an exact algorithm (which we proposed) and whose appealing performance is demonstrated through simulations; (2) An exact algorithm for $m = 1$ in line topology and directed acyclic topologies.

- In practice, there is usually sparse regeneration capacity in the network. Therefore, the objective may not only be finding a feasible path but also minimizing the required number of regenerations. Our algorithms are capable of returning feasible paths with minimal number of regenerations.

Since regenerators are costly, and are active elements that consume power and require maintenance, the main objectives when placing regenerators in a network are minimizing the total number of regenerators and minimizing the total number of regenerator nodes (i.e., nodes with regeneration capability). The regenerator placement problem with the objective of minimizing the total number of regenerator nodes is proved to be NP-hard in the literature. This approach is based on the assumption that regenerator nodes have full regeneration capability, i.e., there is no limitation on the number of wavelengths that are simultaneously regenerated at a node. However, under existing practical and reliable technologies, regeneration is performed per wavelength. Therefore, assuming full regeneration is costly since a separate regenerator is placed beforehand for each possible wavelength regeneration. Thus, we instead focused on minimizing the total number of regenerators as a main objective. Our main conclusions are:

- For $m > 1$, the regeneration placement problem is NP-hard.
- For unprotected lightpaths and $m = 1$, the problem is polynomially solvable when there is no restriction on the number of wavelengths, and is NP-hard if the number of wavelengths is restricted.
- For unprotected lightpaths and $m = 1$, the problem becomes NP-hard when minimizing the total number of regenerator nodes is a secondary objective, even when there is no restriction on the number of wavelengths.
- For protected lightpaths (dedicated or shared), the regenerator placement problem is NP-hard, even for $m = 1$.
- Our main contributions are (1) an exact algorithm for unprotected lightpaths when there is no restriction on the number of wavelengths, (2) efficient heuristic algorithms for unprotected lightpaths when minimizing the number of regenerator nodes is a secondary objective or when there is restriction on the number of wavelengths, and (3) approximation algorithm and efficient heuristic algorithms for dedicated and shared protection schemes.

Since the capacity offered by wavelength channels, which can be as high as 100 Gb/s, may not be fully utilized by individual requests, it may be necessary to aggregate several independent traffic streams onto a single lightpath. This is known as traffic grooming and leads to efficient use of available capacity. Our main conclusions with regards to traffic grooming are:

- The survivable (impairment-aware) traffic grooming problem, which minimizes the cost of traffic grooming and regeneration, is NP-hard.

- Our main contributions are (1) constant-factor approximation algorithms (when the traffic is uniform) and (2) variable-factor approximation algorithms or heuristic algorithms with upper-bounds (when the traffic is non-uniform) for ring topologies, which are among the most widely deployed topologies in optical networks.

In addition, a case study of survivable and impairment-aware RWA is made on the SURFnet6 network, which connects research and educational institutes in the Netherlands. In this case study, simulations were conducted on the network (using realistic data such as types of nodes, impairment values, traffic matrix, etc.) to compare our proposed approach with a greedy sequential approach, which is often used by practitioners. The results showed that the number of regenerators and wavelengths required by our approach are significantly less than those of the sequential approach. This leads to a significant reduction of not only the capital expenditure (CAPEX), but also the operational expenditure (OPEX) because of the reduced power consumption and heat dissipation, and operational cost associated with each wavelength. From a practical implementation point of view, the results obtained from these simulations have been employed as an input during the redesigning process of the SURFnet6 network.

## 10.3 Inter-domain RWA

With the increasing deployment of optical networks, future optical networks will require new protocols in order to route and support on-demand provisioning of lightpaths between different domains. In this thesis, we have modified and studied the performance of three different inter-domain routing protocols in the presence of wavelength converters. The following are our main conclusions:

- The exchange of (even a highly aggregated) path state information (PSI) between domains can significantly reduce the blocking ratio, and the number of messages sent between border optical cross-connects (OXCs).

- With enough wavelength converters, an inter-domain routing algorithm suggested in [114], which allows the exchange of a highly aggregated PSI, can achieve a blocking ratio that is recommended for optical networks to support real-time and streaming applications.

## 10.4 Future Work

In this thesis, we have thoroughly studied various problems related to survivability and impairment-aware routing in WDM optical networks. Therefore, this thesis work will provide a strong foundation for related future works. There are several interesting venues for future work, which can be direct extensions of this thesis or independent work that may result in another thesis.

## Extension work

1. Different survivability schemes: As is mentioned in Chapter 1, there are several survivability techniques depending on the objective at hand. In this thesis, we have mainly focused on dedicated and shared protection schemes. Future studies may look into other survivability techniques, and whether our approaches can also be extended to employ these techniques.

2. Non-linear impairments: As is described in [98], the effect of physical impairments can be well approximated by considering some of the main linear impairments, and assuming that the non-linear impairments are bounded and implicitly reflected in a maximum number of spans. However, it may still be worth investigating the effect of non-linear impairments to obtain a complete picture.

## Independent work

1. Survivability and impairment-aware issues in inter-domain networks: In this thesis, we focused mainly on survivability and impairment-awareness in intra-domain networks. Due to administrative policies, the type and amount of messages exchanged between domains may not be enough to give a complete and global information. Therefore, survivability and impairment-aware approaches in intra-domain networks may not be suitable for inter-domain networks. Therefore, a study of survivability and impairment-aware routing in inter-domain networks can be a good venue for further research.

2. Energy-aware routing and energy-efficient ("green") designs in optical networks: This is an area that is gaining a lot of interest in the research world, and the main goal is to propose routing and design techniques that are sustainable, and have minimal environmental impacts. Most of the work done in this area is for wireless networks, and there is only a limited work for wired networks in general, and optical networks in particular. In this thesis, we have studied survivable and impairment-aware traffic grooming, where the objective is to minimize the cost of regeneration and grooming. Even though energy efficiency is not the main objective, our approach usually leads to a reduction in power consumption. However, it might not necessarily result in the most energy-efficient solution. Therefore, a study into the trade-off (if necessary) between regeneration/grooming cost and energy-efficiency can also be another venue.

# Appendix A

# Exact Algorithm for Minimized RPP

We now provide an exact ILP formulation to solve the **Minimized RPP** problem of Chapter 5. As stated earlier, the optimal number of regenerators required by each request can be computed in polynomial time using algorithm $ESRPP$. Thus, we assume that for each request, this value is precomputed and our interest is to find paths that minimize the total number of distinct nodes where these regenerators are placed. We first give the formulation for directed networks, and then show how it is slightly modified in the case of undirected networks.

**Indices/constants:**

| | |
|---|---|
| $f = 1, \dots, F$ | ID of requests ($F$ in total). |
| $\mathcal{L}^+(u)$ | Set of outgoing links of node $u$. |
| $\mathcal{L}^-(u)$ | Set of incoming links of node $u$. |
| $R_f$ | The minimum number of regenerators required for routing request $f$. |

**Variables (integers):**

| | |
|---|---|
| $\gamma_{f,l,u}$ | is 1 if request $f$ uses link $l$ and node $u$ is the last regenerator node (or the source node) before encountering link $l$; 0 otherwise. |
| $\tau_{f,u,v}$ | is 1 if request $f$ places a regenerator at node $u$ directly followed by a regenerator at node $v$, i.e., there is a regeneration segment of lightpath $f$ between $u$ and $v$; 0 otherwise. Node $u$ can be the source node. |
| $x_u$ | is 1 if a regenerator is placed at node $u$; 0 otherwise. |
| $y_u$ | (*auxiliary variable*) represents the number of regenerators placed at node $u$. |

**Objective:**

Minimize the number of regenerator nodes in the network,

$$Minimize : \sum_{u \in \mathcal{N}} x_u \tag{A.1}$$

**Constraints:**

*Conservation constraints*:

At the source node of any request,

$$\sum_{l \in \mathcal{L}^+(s_f)} \gamma_{f,l,s_f} = 1 \qquad 1 \leq f \leq F \tag{A.2}$$

At intermediate nodes:

If a given node $v$ is not the source or destination node of a request $f$, then any flow related to the request that enters $v$ has to leave it after being regenerated ($\tau_{fuv} = 1$) or not ($\tau_{fuv} = 0$).

$$\sum_{l \in \mathcal{L}^-(v)} \gamma_{f,l,u} - \sum_{l \in \mathcal{L}^+(v)} \gamma_{f,l,u} = \tau_{f,u,v} \tag{A.3}$$
$$1 \leq f \leq F; \forall v \in \mathcal{N} \backslash \{s_f, d_f\}; \forall u \in \mathcal{N} \backslash \{v\}$$

If the request $f$ is regenerated at node $v$, the last regenerator node in the new segment should be node $v$, and not any other node.

$$\sum_{l \in \mathcal{L}^+(v)} \gamma_{f,l,v} - \sum_{u \in \mathcal{N} \backslash \{v\}} \tau_{f,u,v} = 0 \quad 1 \leq f \leq F; \forall v \in \mathcal{N} \backslash \{s_f, d_f\} \tag{A.4}$$

*Simple path constraints*:

A path taken by a request should not contain loops.

For the source node of any request $f$, there should not be a flow associated with any of its incoming links.

$$\sum_{l \in \mathcal{L}^-(s_f)} \sum_{u \in \mathcal{N}} \gamma_{f,l,u} = 0 \quad 1 \leq f \leq F \tag{A.5}$$

In addition, for request $f$, any flow that exits its source node, other than the one originating at the source node, should explicitly be set to 0.

$$\sum_{l \in \mathcal{L}^+(s_f)} \sum_{u \in \mathcal{N} \backslash \{s_f\}} \gamma_{f,l,u} = 0 \quad 1 \leq f \leq F \tag{A.6}$$

Similarly, for any intermediate node, there can at most be one flow associated with request $f$ entering the node.

$$\sum_{l \in \mathcal{L}^-(v)} \sum_{u \in \mathcal{N}} \gamma_{f,l,u} \leq 1 \quad 1 \leq f \leq F; \forall v \in \mathcal{N} \backslash \{s_f\} \tag{A.7}$$

*Impairment constraints*:
The impairment value of any transparent segment should be less than the threshold,

$$\sum_{l \in \mathcal{L}} r(l) \cdot \gamma_{f,l,u} \leq \Delta \qquad 1 \leq f \leq F; \forall u \in \mathcal{N} \tag{A.8}$$

*Number of Regenerator constraints:*
The number of regenerators needed for request $f$ should exactly be $R_f$.

$$\sum_{u \in \mathcal{N}} \sum_{v \in \mathcal{N} \setminus \{u\}} \tau_{f,u,v} = R_f \qquad 1 \leq f \leq F \tag{A.9}$$

**Equations:**
The number of regenerators at a given node,

$$y_v = \sum_{f=1}^{F} \sum_{u \in \mathcal{N}} \tau_{f,u,v} \qquad \forall v \in \mathcal{N} \tag{A.10}$$

A node is a regenerator node if there is at least one regenerator placed at that node.

$$y_v \leq F \cdot x_v \qquad \forall v \in \mathcal{N} \tag{A.11}$$

In order to facilitate the performance of the ILP formulation, the following observations can be incorporated.

1. A feasible regeneration segment should have a cost of at most $\Delta$. Thus, for any pair of nodes $u$ and $v$, if $r(P^*_{u \to v}) > \Delta$, then $\tau_{f,u,v} = 0$ and $\gamma_{f,l,v} = 0$, for every request $f$ and $l \in \mathcal{L}^+(v)$.

2. For any request $f$, exactly $R_f$ regenerators are needed. Thus, for any node $u$, if $r(P^*_{s_f \to u}) > (R_f + 1)\Delta$, then $\gamma_{f,l,u} = 0$ for any $l \in \mathcal{L}^+(u)$.

For undirected networks, we first replace each link with two directed links in either direction. Let for each $l = (u, v) \in \mathcal{L}$, its corresponding oppositely directed link be $l' = (v, u) \in \mathcal{L}$. Then, we add the following equation,

$$\sum_{u \in \mathcal{N}} (\gamma_{f,l,u} + \gamma_{f,l',u}) \leq 1 \qquad 1 \leq f \leq F; \forall l \in \mathcal{L} \tag{A.12}$$

# Appendix B

# Exact Algorithm for WCRP

We now provide an exact ILP formulation for solving the **WCRP** problem of Chapter 5. We first give the ILP formulation for directed networks, and show how it is modified for undirected networks.

**Indices:**

| | |
|---|---|
| $f = 1, \ldots, F$ | ID of requests. |
| $\lambda = 1, \ldots, W$ | ID of wavelengths. |
| $\mathcal{P} = \{P_f \mid f = 1, \ldots, F\}$ | Set of paths assigned for the requests. |

For each $f$, $f = 1, \ldots, F$, let $P_f$ represent the set of links in path $P_f$, and $\{P_f\}$ represent the set of nodes in path $P_f$.

**Variables (binary):**

| | |
|---|---|
| $\gamma_{f,l,u,\lambda}$ | is 1 if request $f$ uses wavelength $\lambda$ on link $l$ and node $u$ is the last regenerator node (or the source node) before encountering link $l$; 0 otherwise. |
| $\tau_{f,u,v,\lambda}$ | is 1 if request $f$ uses a regenerator at node $u$ followed by a regenerator at node $v$, and wavelength $\lambda$ is used on the segment between $u$ and $v$; 0 otherwise. Node $u$ can be the source node. |

**Objective:**

Minimize the total number of regenerators/converters placed for all requests,

$$Minimize : \sum_{f=1}^{F} \sum_{\lambda=1}^{W} \sum_{u \in \{P_f\}} \sum_{v \in \{P_f\} \setminus \{u\}} \tau_{f,u,v,\lambda} \tag{B.1}$$

**Constraints:**

*Flow Conservation constraints*:

At the source node of each request,

$$\sum_{l \in \mathcal{L}^+(s_f) \cap P_f} \sum_{\lambda=1}^{W} \gamma_{f,l,s_f,\lambda} = 1 \qquad 1 \leq f \leq F \tag{B.2}$$

At intermediate nodes:

A lightpath at a given wavelength may be regenerated/converted at an intermediate node,

$$\sum_{l \in \mathcal{L}^-(v) \cap P_f} \gamma_{f,l,u,\lambda} - \sum_{l \in \mathcal{L}^+(v) \cap P_f} \gamma_{f,l,u,\lambda} = \tau_{f,u,v,\lambda} \tag{B.3}$$
$$1 \leq f \leq F; \forall v \in \{P_f\} \backslash \{s_f, d_f\}; \forall u \in \{P_f\} \backslash \{v\}; \lambda \leq f \leq W$$

A lightpath regenerated/converted at an intermediate node $v$ may change its wavelength, and the last regenerator node in the new segment should be node $v$.

$$\sum_{\lambda} \sum_{l \in \mathcal{L}^+(v) \cap P_f} \gamma_{f,l,v,\lambda} - \sum_{\lambda} \sum_{u \in \{P_f\} \backslash \{v\}} \tau_{f,u,v,\lambda} = 0 \tag{B.4}$$
$$1 \leq f \leq F; \forall v \in \{P_f\} \backslash \{s_f, d_f\}$$

*Wavelength constraints*:

A wavelength at a given link can only be used once,

$$\sum_{f=1}^{F} \sum_{u \in \{P_f\}} \gamma_{f,l,u,\lambda} \leq 1 \qquad 1 \leq \lambda \leq W; \forall l \in \mathcal{L} \tag{B.5}$$

*Impairment constraints*:

The physical impairment of any transparent segment should be less than the threshold,

$$\sum_{\lambda=1}^{W} \sum_{l \in P_f} r(l) \cdot \gamma_{f,l,u,\lambda} \leq \Delta \qquad 1 \leq f \leq F; \forall u \in \{P_f\} \tag{B.6}$$

For undirected networks, we first replace each link with two directed links in either direction. Let for each $l = (u, v) \in \mathcal{L}$, its corresponding oppositely directed link be $l' = (v, u) \in \mathcal{L}$. Then, we add the following equation,

$$\sum_{\lambda=1}^{W} \sum_{u \in \{P_f\}} (\gamma_{f,l,u,\lambda} + \gamma_{f,l',u,\lambda}) \leq 1 \qquad 1 \leq f \leq F; \forall l \in P_f \tag{B.7}$$

and replace Equation B.5 with the following equation,

$$\sum_{f=1}^{F} \sum_{u \in \{P_f\}} \gamma_{f,l,u,\lambda} + \gamma_{f,l',u,\lambda} \leq 1 \qquad 1 \leq \lambda \leq W; \forall l \in \mathcal{L} \tag{B.8}$$

# Appendix C

# Exact Algorithm for SRSRP

We now provide an exact ILP formulation using network flow equations to solve the **SRSRP** problem of Chapter 6. We first give the ILP formulation for directed networks, and then show how it is slightly modified in the case of undirected networks.

**Variables (binary):**

$x_{l,u}$ is 1 if the primary lightpath uses link $l$ and node $u$ is its last regenerator node (or the source node) before encountering link $l$; 0 otherwise.

$y_{l,u}$ is 1 if the backup lightpath uses link $l$ and node $u$ is its last regenerator node (or the source node) before encountering link $l$; 0 otherwise.

$\tau_{u,v}$ is 1 if the primary lightpath uses a regenerator at node $u$ directly followed by a regenerator at node $v$. Node $u$ can also be the source node.

$\psi_{u,v}$ is 1 if the backup lightpath uses a regenerator at node $u$ directly followed by a regenerator at node $v$. Node $u$ can also be the source node.

$\alpha_u$ (Only for the shared variant) is 1 if a regenerator (shared or not) is needed at node $u$; 0 otherwise.

**Objective:**

Minimize the total number of regenerators needed by the primary and backup lightpaths,

For the *dedicated-dedicated* variant:

$$Minimize : \sum_{u \in \mathcal{N}} \sum_{v \in \mathcal{N}} \left( \tau_{u,v} + \psi_{u,v} \right) \tag{C.1}$$

For the *dedicated-shared* variant:

$$Minimize : \sum_{u \in \mathcal{N}} \alpha_u \tag{C.2}$$

**Constraints:**

*Flow Conservation constraints*:

At the source node:

There are exactly two flows leaving the source node, one for the primary and another for the backup lightpaths.

$$\sum_{l \in \mathcal{L}^+(s)} (x_{l,s} + y_{l,s}) = 2 \tag{C.3}$$

At intermediate nodes:

If a given node $v$ is not the source or the destination node, then the flow related to the primary/backup lightpath that enters $v$ has to leave it after being regenerated ($\tau_{uv} = 1$ for the primary and $\psi_{uv} = 1$ for the backup lightpath) or not ($\tau_{uv} = 0$ for the primary and $\psi_{uv} = 0$ for the backup lightpath).

$$\begin{aligned} \sum_{l \in \mathcal{L}^-(v)} x_{l,u} - \sum_{l \in \mathcal{L}^+(v)} x_{l,u} = \tau_{u,v} \qquad \text{and} \\ \sum_{l \in \mathcal{L}^-(v)} y_{l,u} - \sum_{l \in \mathcal{L}^+(v)} y_{l,u} = \psi_{u,v} \\ \forall v \in \mathcal{N} \backslash \{s,d\}; \forall u \in \mathcal{N} \backslash \{v\} \end{aligned} \tag{C.4}$$

If a lightpath is regenerated at node $v$, the last regenerator node in the new segment should be node $v$, and not any other node.

$$\begin{aligned} \sum_{l \in \mathcal{L}^+(v)} x_{l,v} - \sum_{u \in \mathcal{N} \backslash \{v\}} \tau_{u,v} &= 0 \quad \text{and} \\ \sum_{l \in \mathcal{L}^+(v)} y_{l,v} - \sum_{u \in \mathcal{N} \backslash \{v\}} \psi_{u,v} &= 0 \qquad \forall v \in \mathcal{N} \backslash \{s,d\} \end{aligned} \tag{C.5}$$

*Disjointedness constraints*

The primary and backup lightpaths should be link-disjoint.

$$\sum_{u \in \mathcal{N}} (x_{l,u} + y_{l,u}) \leq 1 \qquad \forall l \in \mathcal{L} \tag{C.6}$$

*Simple path constraints*:

The lightpaths should not contain loops.

At the source node, there should not be a flow associated with any of its incoming links.

$$\sum_{l \in \mathcal{L}^-(s)} \sum_{u \in \mathcal{N}} (x_{l,u} + y_{l,u}) = 0 \tag{C.7}$$

In addition, any flow that exits the source node, other than the one originating at the source node, should explicitly be set to 0.

$$\sum_{l \in \mathcal{L}^+(s)} \sum_{u \in \mathcal{N} \backslash \{s\}} (x_{l,u} + y_{l,u}) = 0 \tag{C.8}$$

Similarly, for any intermediate node, there can at most be one flow of the primary or backup lightpath entering the node.

$$\sum_{l \in \mathcal{L}^-(v)} \sum_{u \in \mathcal{N}} x_{l,u} \leq 1 \text{ and } \sum_{l \in \mathcal{L}^-(v)} \sum_{u \in \mathcal{N}} y_{l,u} \leq 1 \ \forall v \in \mathcal{N} \backslash \{s\} \tag{C.9}$$

*Impairment constraints*:

The physical impairment of any transparent segment should be less than the threshold,

$$\sum_{l \in \mathcal{L}} r(l) \cdot x_{l,u} \leq \Delta \text{ and } \sum_{l \in \mathcal{L}} r(l) \cdot y_{l,u} \leq \Delta \qquad \forall u \in \mathcal{N} \tag{C.10}$$

Only for the *dedicated-shared* variant:

$$\sum_{u \in \mathcal{N}} \left(\tau_{u,v} + \psi_{u,v}\right) \leq 2 \cdot \alpha_u \qquad \forall v \in \mathcal{N} \tag{C.11}$$

For undirected networks, we first replace each link with two directed links in either direction. Let for each $l = (u, v) \in \mathcal{L}$, its corresponding oppositely directed link be $l' = (v, u) \in \mathcal{L}$. Then replace Equation C.6 with the following equation.

$$\sum_{u \in \mathcal{N}} (x_{l,u} + y_{l,u} + x_{l',u} + y_{l',u}) \leq 1 \qquad \forall l \in \mathcal{L} \tag{C.12}$$

# Appendix D

# Abbreviations

| | |
|---|---|
| ADM | Add/Drop Multiplexer |
| AS | Autonomous System |
| ASE | Amplifier Spontaneous Emission |
| ASON | Automatic Switched Optical Network |
| BER | Bit Error Rate |
| BFS | Breadth First Search |
| BGP | Border Gateway Protocol |
| CRMP | Convertor/Regenerator Minimization Problem |
| CAPEX | Capital Expenditure |
| DDRP | Domain-to-Domain Routing Protocol |
| ENAW | Effective Number of Available Wavelengths |
| FFD | First Fit Decreasing |
| FoM | Figure of Merit |
| FPTAS | Fully Polynomial-Time Approximation Scheme |
| FWM | Four-Wave Mixing |
| Gb/s | Giga bits per second |
| GMPLS | Generalized Multi-Protocol Label Switching |
| IDRA | Inter-Domain Routing Agents |
| IDRP | Inter-Domain Routing Agents based Protocol |
| IETF | Internet Engineering Task Force |
| ILP | Integer Linear Programming |
| ITU | International Telecommunications Union |
| MDLDP | Minimum Disruption Link-Disjoint Paths problem |
| MDP | Maximum Disjoint Paths problem |
| MPLS | Multi-Protocol Label Switching |
| NRI | Network Reachability Information |
| OADM | Optical Add/Drop Multiplexer |
| OBGP | Optical Border Gateway Protocol |
| O-E-O | Optical-to-Electrical-to-Optical conversion |

| | |
|---|---|
| OIF | Optical Internetworking Forum |
| OPEX | Operational Expenditure |
| OSNR | Optical Signal to Noise Ratio |
| OTN | Optical Transport Network |
| OXC | Optical Cross-connect |
| PDL | Polarization Dependent Loss |
| PMD | Polarization Mode Dispersion |
| PSI | Path State Information |
| ROADM | Reconfigurable Optical Add/Drop Multiplexer |
| RPP | Regenerator Placement Problem |
| RWA | Routing and Wavelength Assignment |
| SAMCRA | Self-Adaptive Multiple Constraint Routing Algorithm |
| SDH | Synchronous Digital Hierarchy |
| SONET | Synchronous Optical Networking |
| SRLG | Shared Risk Link Group |
| SRRP | Single Request Regenerator Placement problem |
| SRSRP | Single Request Survivable Regenerator Placement problem |
| SRWA | Survivable Routing and Wavelength Assignment |
| SSRP | Shared Survivable Regenerator Placement problem |
| Tb/s | Tera bits per second |
| TE | Traffic Engineering |
| WCRP | Wavelength-Constrained Regenerator Placement problem |
| WDM | Wavelength Division Multiplexing |
| WSS | Wavelength Selective Switch |

# Appendix E

# Notations

| | |
|---|---|
| $\mathcal{N}$ | Set of nodes |
| $N$ | Number of nodes |
| $\mathcal{L}$ | Set of links |
| $L$ | Number of links |
| $\mathcal{W}$ | Set of wavelengths |
| $W$ | Number of wavelengths |
| $\mathcal{F}$ | Set of requests |
| $F$ | Number of requests |
| $\mathcal{N}_R$ | Set of regenerator or converter nodes |
| $N_R$ | Number of regenerator or converter nodes |
| $R$ | Number of regenerators or converters |
| $r$ | Impairment value |
| $\Delta$ | Impairment threshold |
| $m$ | The number of physical impairments |
| $k_{\max}$ | The maximum number of paths stored for a given node |
| $s$ | Source node |
| $d$ | Destination node |

# Appendix F

# List of Algorithms

| Name | Complexity | Problem | Type | Ch. |
|---|---|---|---|---|
| MIN_SUM+ | $O(K \cdot N(L+N\log N))$ | Variants of the min-sum problem with secondary objectives. | Heuristic | 2 |
| MSA | $O(WN\log N + WL)$ | Dedicated wavelength rerouting. | Approx. | 3 |
| ESA | $O(WN^2\log N + WNL)$ | Dedicated Wavelength rerouting. | Approx. | 3 |
| APFR | $O(KWN^2L)$ | Shared Wavelength Rerouting. | Heuristic | 3 |
| PIARA | $O(N_RN\log N + N_RL)$ | Impairment-aware path selection (loops allowed). | Exact | 4 |
| EIARA | $O(mN_RN\log N + k_{\max}N\log(k_{\max}N) + k_{\max}LN)$ | Impairment-aware path selection. | Exact | 4 |
| TIARA | $O(mN_RN\log N + mN_RL + NL)$ | Impairment-aware path selection. | Heuristic | 4 |
| LAH | $O(mN_RN\log N + mN_RL)$ | Impairment-aware path selection. | Heuristic | 4 |
| DCRM | $O(WN_R^2(L+N\log N))$ | Impairment-aware path selection in DAGs. | Exact | 4 |
| ESRRP | $O(N^2\log N + NL)$ | Regenerator placement. | Exact | 5 |
| ELRP | $O(FN^2)$ | Regenerator placement with a secondary objective (Line topology). | Exact | 5 |
| MURP | $O(FN^2\log N + FNL)$ | Regenerator placement with a secondary objective. | Heuristic | 5 |
| GRP | $O(FNW)$ | Regenerator/converter placement. | Heuristic | 5 |
| SRH | $O(N^2\log N + NL)$ | Dedicated surv. regen. placement. | Heuristic | 6 |
| APF | $O(FN^2\log N + FNL)$ | Shared surv. regen. placement. | Heuristic | 6 |
| USGA | $O\left(N^2\right)$ | Survivable impairment-aware routing (uniform traffic) in rings. | Approx. | 7 |
| NSGA | $O\left(N^2\right)$ | Surv. impairment-aware routing (non-uniform traffic) in rings. | Heuristic | 7 |

# Bibliography


[1] R. Adler and Y. Azar, "Beating the logarithmic lower bound: Randomized preemptive disjoint paths and call control algorithms," *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms*, pp. 1-10, 1999.

[2] M. Ali Ezzahdi, S. Al Zahr, M. Koubaa, N. Puech, and M. Gagnaire, "LERP: a quality of transmission dependent heuristic for routing and wavelength assignment in hybrid WDM networks," *Proceedings of ICCCN 2006*, pp. 125–136, 2006.

[3] M. Allen, C. Liou, S. Melle, and V. Vusirikala, "Digital optical networks using photonic integrated circuits (PICs) address the challenges of reconfigurable optical networks," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 35-43, Jan. 2008.

[4] O. Amini, S. Pérennes, and I. Sau, "Hardness and approximation of traffic grooming," *Proceedings of 18th International Symposium on Algorithms and Computation*, pp. 561-573, 2007.

[5] R.S. Anand and T. Erlebach, "On-line algorithms for edge-disjoint paths in trees of rings," *Proceedings of the 5th Latin American Symposium on Theoretical Informatics*, pp. 584 - 597, 2002.

[6] R. Andersen, F. Chung, A. Sen, and G. Xue, "On disjoint path pairs with wavelength continuity constraint in WDM networks," *Proceedings of IEEE INFOCOM'04*, pp. 524–535, March 2004.

[7] M. Andrews and L. Zhang, "Complexity of wavelength assignment in optical network optimization," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 646-657, April 2009.

[8] B. Awerbuch, Y. Azar, and S. Plotkin, "Throughput-competitive on-line routing," *Proceedings of the 34th IEEE Foundations of Computer Science*, pp. 32-40, Nov. 1993.

[9] B. Awerbuch, Y. Azar, A. Fiat, and A. Rosen, "Competitive non-preemptive call control," *Proceedings of the 5th ACM-SIAM Symposium on Discrete Algorithms*, pp. 312-320, 1994.

[10] B. Awerbuch, Y. Azar, A. Fiat, S. Leonardi, and A. Rosen, "On-line competitive algorithms for call admission in optical networks," *Proceedings of the 4th European Symposium on Algorithms*, pp. 431-444, 1996.

[11] B. Awerbuch, R. Gawlick, F.T. Leighton, and Y. Rabani, "On-line admission control and circuit routing for high performance computing and communication," *Proceedings of Symposium on Foundations of Computer Science*, pp. 412–423, 1994.

[12] S. Azodolmolky, M. Klinkowski, E. Marin, D. Careglio, J.S. Pareta, and I. Tomkos, "A survey on physical layer impairments aware routing and wavelength assignment algorithms in optical networks," *Computer Networks*, vol. 53, no. 7, pp. 926–944, 2009.

[13] Y. Bartal, A. Fiat, and S. Leonardi, "Lower bounds for on-line graph problems with application to on-line circuit and optical routing," *Proceedings of the 28th ACM Symposium on the Theory of Computing*, pp. 531-540, 1996.

[14] K. Benson, B. Birnbaum, E. Molina-Estolano, and R. Libeskind-Hadas, "Competitive analysis of online traffic grooming in WDM rings," *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 984–997, August 2008.

[15] J. Berthold, A.A.M. Saleh, L. Blair, and Jane M. Simmons, "Optical Networking: Past, present, and future," *Journal of Lightwave Technology*, vol. 26, no. 9, pp. 1104-1118, 2008.

[16] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*, Kluwer Academic, 1999.

[17] M. Blanchet, F. Parent, and B. St-Arnaud, "Optical BGP (OBGP): IDRA lightpath provisioning," IETF draft, ietf-draft-parent-obgp-01, Mar. 2001.

[18] A. Bley, "On the complexity of vertex-disjoint length-restricted path problems," *Computational Complexity*, vol. 12, pp. 131-149, 2003.

[19] N. Bouabdallah and H.G. Perros, "Cost-effective single-hub WDM ring networks: A proposal and analysis," *Computer Networks*, vol. 51, no. 13, pp. 3878-3901, 2007.

[20] E. Bouillet and J.-F. Labourdette, "Distributed computation of shared backup path in mesh optical networks using probabilistic methods," *IEEE/ACM Transactions on Networking*, vol. 12, no. 5, pp. 920-930, Oct. 2004.

[21] B. Van Caenegem, W. Van Parys, F. De Turck, and P.M. Demeester, "Dimensioning of survivable WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, Sept. 1998.

[22] D. Campi and C. Coriasso, "Wavelength conversion technologies," *Photonic Network Communications*, vol. 2, no. 1, pp. 85-95, Mar. 2000.

[23] T.J. Carpenter, R.C. Menendez, D.F. Shallcross, J.W. Gannett, J. Jackel, and A.C. Von Lehmen, "Cost-conscious impairment-aware routing," *Proceedings of IEEE/OSA OFC 2004*, Feb. 2004.

[24] B. Chen, G. N. Rouskas, and R. Dutta, "Traffic grooming in WDM ring networks to minimize the electronic port cost," *Optical Switching and Networking*, vol. 2, no. 1, pp. 1–18, May 2005.

[25] S. Chen, I. Ljubic, and S. Raghavan, "The regenerator location problem," *Networks*, vol. 55, no. 3, pp. 209-220, Dec. 2009.

[26] L-W. Chen and E. Modiano, "Efficient routing and wavelength assignment for reconfigurable WDM networks with wavelength converters," *IEEE INFOCOM'03*, 2003.

[27] A. L. Chiu and E. H. Modiano, "Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks," *Journal of Lightwave Technology*, vol. 18, no. 1, pp. 2-12, Jan. 2000.

[28] T. Chow and P. Lin, "The ring grooming problem," *Networks*, vol. 44, pp. 194-202, 2004.

[29] X. Chu, J. Liu, and Z. Zhang, "Analysis of sparse-partial wavelength conversion in wavelength-routed WDM networks," *IEEE INFOCOM'04*, 2004.

[30] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd edition, MIT Press, Cambridge, MA, 2001.

[31] S. De Maesschalc, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, R. Inkret, B. Mikac, and J. Derkacz, "Pan-European optical transport networks: an availability-based comparison," *Photonic Network Communications,* vol. 5, no. 3, pp. 203-225, May 2003.

[32] T. Deng and S. Subramaniam, "Adaptive QoS routing in dynamic wavelength-routed optical networks," *Proceedings of the 2nd International Conference on Broadband Networks*, pp. 184–193, 2005.

[33] M. Flammini, A.M. Spaccamela, G. Monaco, L. Moscardelli, and S. Zaks, "On the complexity of the regenerator placement problem in optical networks," *Proceedings of SPAA'09*, pp. 154-162, 2009.

[34] M. Flammini, G. Monaco, L. Moscardelli, M. Shalom, and S. Zaks, Shmuel, "Optimizing regenerator cost in traffic grooming, *Lecture Notes in Computer Science*, vol. 6490, pp. 443-458, 2010.

[35] A. Fumagalli and L. Valcarenghi, "IP restoration vs. WDM protection: Is There an Optimal Choice?," *IEEE Network*, vol. 14, no. 6, pp. 34–41,. Nov./Dec. 2000.

[36] A. Fumagalli, I. Cerutti, M. Tacca, F. Masetti, R. Jagannathan, and S. Alagar, "Survivable networks based on optimal routing and WDM self-healing rings," *IEEE INFOCOM'99*, vol.2, pp.726-733, Mar. 1999.

[37] M. J. Francisco, L. Pezoulas, C. Huang, and I. Larnbadaris, "End-to-end signaling and routing for optical IP networks," *Proceedings of IEEE ICC*, Apr. 2002.

[38] J.A. Garay, I.S. Gopal, S. Kutten, Y. Mansour, and M. Yung, "Efficient online call control algorithms," *Proceedings of Symposium on Theory of Computing Systems*, pp. 285-293, 1993.

[39] M.R. Garey and D.S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[40] O. Gerstel and R. Ramaswami, "Optical Layer Survivability-An Implementation Perspective," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1885–1923, Oct. 2000.

[41] O. Gerstel and R. Ramaswami, "Fault Tolerant Multiwavelength Optical Rings with Limited Wavelength Conversion," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, Sept. 1998.

[42] O. Gerstel, R. Ramaswami, and G. Sasaki, "Cost-effective traffic grooming in WDM rings," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 618–630, Oct. 2000.

[43] K.P. Gummadi, M.J. Pradeep, and C.S. Ram Murthy, "An efficient primary-segmented backup scheme for dependable real-rime communication in multihop networks," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, Feb. 2003.

[44] J. Harmatos and P. Laborczi "Dynamic routing and wavelength assignment in survivable WDM networks," *Photonic Network Communications*, vol. 4, no. 3/4, pp. 357-376, July 2002.

[45] R. Hayashi, K. Shimizu, I. Inoue, and K. Shiomoto, "Novel impairment-aware traffic engineering in all-optical networks," *Proceedings of PACRIM'09*, pp. 152-157, 2009.

[46] J. He, M. Brandt-Pearce, Y. Pointurier, and S. Subramaniam, "QoT-aware routing in impairment-constrained optical networks," *Proceedings of IEEE GLOBECOM'07*, pp. 2269–2274, 2007.

[47] R. He, B. Lin, and L. Lia, "Dynamic service-level-agreement aware shared-path protection in WDM mesh networks," *Journal of Network and Computer Applications*, vol. 30 pp. 429–444, 2007.

[48] J. Homa and K. Bala, "ROADM architectures and their enabling WSS technology," *IEEE Communications Magazine*, vol. 46, no. 7, pp. 150-154, July 2008.

[49] Y. Huang, J.P. Heritage, and B. Mukherjee, "Connection provisioning with transmission impairment consideration in optical WDM networks with high-speed channels," *Journal of Lightwave Technology*, vol. 23, no. 3, pp. 982-993, March 2005.

[50] S. Huang, R. Dutta, and G. Rouskas, "Traffic grooming in path, star, and tree networks : Complexity, bounds, and algorithms," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 4, pp. 66-82, April 2006.

[51] P. Ho, J. Tapolcai, and H.T. Mouftah, "On achieving optimal survivable routing for shared protection in survivable next-generation Internet," *IEEE Transactions on Reliability*, vol. 53, no. 2, June 2004.

[52] I-S. Hwang, S-N. Lee, Y-F. Chuang, "Multicast wavelength assignment with sparse wavelength converters to maximize the network capacity using ILP formulation in WDM mesh networks," *Photonic Network Communications*, vol. 12, no. 2, pp. 161-172.

[53] J. Iness and B. Mukherjee, "Sparse wavelength conversion in wavelength-routed WDM optical networks," *Photonic Network Communications*, vol. 1, no. 3, pp. 183-205, Nov. 1999.

[54] A. Itai, Y. Perl, and Y. Shiloach, "The complexity of finding maximum disjoint paths with length constraints," *Networks*, vol. 12, no. 3, pp. 277-286, 1982.

[55] P. Iyer, R. Dutta, and C.D. Savage, "Minimizing transceivers in optical path networks," *Journal of Optical Networking*, vol. 8, no. 5, pp. 454-461, 2009.

[56] G. Jeong and E. Ayanoglu, "Comparison of wavelength-interchanging and wavelength-selective cross-connects in multiwavelength all-optical networks," *Proceedings IEEE INFOCOM'96*, vol. 1, pp. 156-163, 1996.

[57] X-H. Jia, D-Z. Du, X-D. Hu, H-J. Huang, and D-Y. Li, "On the optimal placement of wavelength converters in WDM networks," *Computer Communications*, vol. 26, no. 9, pp. 986-995, 2003.

[58] D.B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM*, vol. 24, no. 1, pp. 1-13, Jan. 1977.

[59] D. S. Johnson, "Fast algorithms for bin packing," *Journal of Computers and System Sciences*, vol. 8, no. 3, pp. 272–314, 1974.

[60] R.M. Karp,"On the computational complexity of combinatorial problems," *Networks*, vol. 5, no. 9, pp. 45-68, 1975.

[61] J. Kleinberg and É. Tardos, "Disjoint paths in densely embedded graphs," *Proceedings of the 36th Symposium on Foundations of Computer Science*, pp. 52-61, Oct. 1995.

[62] F.A. Kuipers and P. Van Mieghem, "Conditions that impact the complexity of QoS routing," *IEEE/ACM Transaction on Networking*, vol. 13, no. 4, pp. 717-730, August 2005.

[63] F.A. Kuipers, A.A. Beshir, A. Orda, and P. Van Mieghem, "Impairment-aware path selection and regenerator placement in translucent optical networks," *Proceedings of ICNP 2010*, Kyoto, Japan, October 2010.

[64] K.-C. Lee and V.O.K. Li, "A wavelength-convertible optical network," *Journal of Lightwave Technology*, vol. 11, no. 5/6, May/June 1993.

[65] K.C. Lee and V.O.K. Li, "A wavelength rerouting algorithm in wide-area all-optical networks," *Journal of Lightwave Technology*, vol. 14, no. 6, pp. 1218-1229, June 1996.

[66] C.-L. Li, S. T. McCormick, and D. Simchi-Levi, "The complexity of finding two disjoint paths with min-max objective function," *Discrete Applied Mathematics*, vol. 26, no 1, pp. 105–115, Jan. 1990.

[67] G. Liu, C. Ji, and V. Chan, "On the scalability of network management information for inter-domain light-path assessment," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 160-172, Feb. 2005.

[68] Q. Liu, M.A. Kök a, N. Ghani, and A. Gumaste, "Hierarchical routing in multi-domain optical networks," *Computer Communications*, vol. 30, no. 1, pp. 122-131, Dec. 2006.

[69] K. Manousakis, K. Christodoulopoulos, E. Kamitsas, I. Tomkos, and E.A. Varvarigos, "Offline impairment-aware routing and wavelength assignment algorithms in translucent WDM optical networks," *Journal of Lightwave Technology*, vol. 27, no. 12, pp. 1866-1877, June 2009.

[70] G. Markidis, S. Sygletos, A. Tzanakaki, and I. Tomkos, "Impairment aware based routing and wavelength assignment in transparent long haul networks," *Proceedings of IFIP ONDM*, pp. 48-57, May 2007.

[71] A. Marsden, A. Maruta, and K. Kitayama, "FWM-aware dynamic routing and wavelength assignment for wavelength-routed optical networks," *IEICE Transaction of Communication*, vol. E91-B, no. 7, pp. 2145-2151, July 2008.

[72] Y. Miyao and H. Saito, "Optimal design and Evaluation of Survivable WDM Transport Networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1190-1198, Sept. 1998.

[73] G. Mohan and C. Siva Ram Murthy, "A time optimal wavelength rerouting algorithm for dynamic traffic in WDM networks," *Journal of Lightwave Technology*, vol. 17, no. 3, pp. 406-417, 1999.

[74] D. Monoyios, K. Vlachos, M. Aggelou, and I. Tomkos, "On the use of multi-objective optimization algorithms for solving the Impairment Aware RWA problem," *Proceedings of ICC'09*, 2009.

[75] B. Mukherjee, *Optical WDM Networks*, Springer Optical Networks Series, 2006.

[76] NOBEL: A Next Generation Optical Networks for Broadband European Leadership, "IST Integrated Project," http://www.ist-nobel.org.

[77] C. Ou, K. Zhu, J. Zhang, H. Zhu, B. Mukherjee, H. Zang, and L. Sahasrabuddhe, "Traffic grooming for survivable WDM networks: dedicated protection," *Journal of Optical Networking*, vol. 3, no. 1, pp. 50-74, 2004.

[78] P.M. Pachnicke, S. Paschenda, and T. Krummrich, "Physical impairment based regenerator placement and routing in translucent optical networks," *Proceedings of the Optical Fiber communication/National Fiber Optic Engineers Conference*, pp. 1-3, Feb. 2008.

[79] M. Patel, R. Chandrasekaran, and S. Venkatesan, "Improved Quasi-Path Restoration in Mesh Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, Feb. 2008.

[80] A.N. Patel, C. Gao, J.P. Jue, X. Wang, Q. Zhang, P. Palacharla, and T. Naito, "Traffic grooming and regenerator placement in impairment-aware optical WDM networks," *Proceedings of ONDM'10*, pp. 1-6, Feb. 2010.

[81] G.S. Pavani, L.G. Zuliani, H. Waldman, and M.F. Magalhães, "Distributed approaches for impairment-aware routing and wavelength assignment algorithms in GMPLS Networks," *Computer Networks*, vol. 52, no. 10, pp. 1905-195, July 2008.

[82] Q.V. Phung, D. Habibi, H.N. Nguyen, and K. Lo, "K pairs of disjoint paths algorithm for protection in WDM optical networks," *Proceedings of Asia-Pacific Conference on Communications*, pp. 183-187, Oct. 2005.

[83] C. Pinart and H. Harai, "On using optical-layer link information parameters in distributed impairment constraint-based routing," *Proceedings of the Optical Fiber Communication/National Fiber Optic Engineers Conference*, pp. 1-3, March 2007.

[84] S. Rai, C.-F. Su, and B. Mukherjee, "On provisioning in all-optical networks: an impairment-aware approach," *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1989-2001, 2009.

[85] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *Journal of Lightwave Technology*, vol. 21, no. 4, April 2003.

[86] R. Ramaswami and G. H. Sasaki, "Multiwavelength optical networks with limited wavelength conversion," *IEEE INFOCOM'97*, pp. 490–499, 1997.

[87] S. Sankaranarayanan, S. Subramaniam, H. Choi, and H-Ah Choi, "Survivable traffic grooming in WDM ring networks," *Proceedings of SPIE*, vol. 5285, pp. 80-90, 2003.

[88] C.V. Saradhi, L.K. Wei and M. Gurusamy, "Provisioning fault-tolerant scheduled lightpath demands in WDM mesh networks," *International Conference on Broadband Networks*, pp. 150-159, October 2004.

[89] B.H. Shen, B. Hao, and A. Sen, "On multipath routing using widest pair of disjoint paths," *Workshop on High Performance Switching and Routing*, pp. 134-140, 2004.

[90] G. Shen, W.D. Grover, T.H. Cheng, and S.K. Bose, "Sparse placement of electronic switching nodes for low blocking in translucent optical networks," *OSA Journal of Optical Networking*, vol. 1, no. 12, pp. 424–41, Dec. 2002.

[91] G. Shen and R.S. Tucker, "Translucent optical networks: the way forward," *IEEE Communications Magazine*, vol. 45, no. 2, pp. 48–54, 2007.

[92] N. Shinomiya, T. Hoshida, Y. Akiyama, H. Nakashima, and T. Terahara, "Hybrid link/path-based design for translucent photonic network dimensioning," *Journal of Lightwave Technology*, vol. 25, no. 10, pp. 2931-2941, Oct. 2007.

[93] D. Simchi-Levi, "New worst-case results for the bin-packing problem," *Naval Research Logistics*, vol. 41, pp. 579-585, 1994.

[94] J.M. Simmons, *Optical Network Design and Planning*, Springer Publishing Company, 2008.

[95] M. Sivakumar, R.K. Shenai, and K.M. Sivalingam, "A survey of survivability techniques for optical WDM networks," in *Emerging Optical Network Technologies* (K.M. Sivalingam and S. Subramaniam, editors), pp. 297-331, Springer Publishers. 2005.

[96] L. Song and B. Mukherjee, "On the study of multiple backups and primary-backup link sharing for dynamic service provisioning in survivable WDM mesh networks," *IEEE Journal of Selection Areas in Communications*, vol. 26, no. 6, August 2008.

[97] J. Strand, A.L. Chiu, and R. Tkach, "Issues for routing in the optical layer," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 81–87, Feb. 2001.

[98] J. Strand and A. Chiu, "Impairments and Other Constraints on Optical Layer Routing," RFC 4054, May 2005.

[99] J. Suurballe and R. Tarjan,"A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, pp. 325–336, 1984.

[100] A. Todimala, and B. Ramamurthy, "A dynamic partitioning protection routing technique in WDM networks," *Cluster Computing*, vol. 7, no. 3, pp. 259-269, July 2004.

[101] D. Truong and B. Jaumard, "Recent progress in dynamic routing for shared protection in multidomain networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 112-119, June 2008.

[102] D. L. Truong and B. Jaumard, “Using topology aggregation for efficient segment shared protection solutions in multi-domain networks,” *IEEE Journal of Selection Areas in Communications*, vol. 25, no. 9, pp. 96–107, Dec. 2007.

[103] A. Tucker, “Coloring a family of circular arcs,” *SIAM Journal of Applied Mathematics*, vol. 29, no. 3, pp. 493–502, Nov. 1975.

[104] P. Van Mieghem and F.A. Kuipers, “Concepts of exact quality of service algorithms,” *IEEE/ACM Transaction on Networking*, vol. 12, no. 5, pp. 851-864, Oct. 2004.

[105] Y. Wan and W. Liang, “Wavelength rerouting in survivable WDM networks,” *Proceedings of the 4th International IFIP-TC6 Networking Conference*, pp. 431-442, May 2005.

[106] P-J. Wan, O. Frieder, and L. Liu, “Optimal placement of wavelength converters in trees, tree-connected rings, and tree of rings,” *Computer Communications*, vol. 26, no. 7, pp. 718-722, May 2003.

[107] L. Wang, H. Zhang, and L. Zheng, “Reducing the OBGP protection switching time in WDM mesh networks,” *Proceedings of the Optical Fiber Communication/National Fiber Optic Engineers Conference*, March 2006.

[108] D. Xu, Y. Chen, Y. Xiong, C. Qiao, and X. He, “On the complexity of and algorithms for finding the shortest path with a disjoint counterpart,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, Feb. 2006.

[109] G. Xue, “Optimal lightpath routing and rerouting in WDM networks,” *Proceedings of GLOBECOM’ 01*, vol. 4, pp. 2124-2128, 2001.

[110] B. Yang, S.Q. Zheng, and E. Lu, “Finding two disjoint paths in a network with minsum-minmin objective function,” *Proceedings of the International Conference on Foundations of Computer Science*, Las Vegas, Nevada, Jun. 2007.

[111] X. Yang and B. Ramamurthy, “Sparse regeneration in translucent wavelength-routed optical networks: architecture, network design and wavelength routing,” *Photonic Network Communications*, vol. 10, no. 1, July 2005.

[112] X. Yang and B. Ramamurthy, “Dynamic routing in translucent WDM optical networks,” *Proceedings of ICC’02*, vol. 5, pp. 2796-2802, 2002.

[113] M. Yannuzzi, X. Masip-Bruin, S. Sanchez-Lopez, and E. Marin-Tordera, “OBGP+: An improved path-vector protocol for multi-domain optical networks,” *Optical Switching and Networking*, vol. 6, issue 2, pp. 111-119, Apr. 2009.

[114] M. Yannuzzi, X. Masip-Bruin, G. Fabrego, S. Sanchez-Lopez, A. Sprintson, and A. Orda, "Toward a new route control model for multi-domain optical networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 104-111, June 2008.

[115] W. Yao and B. Ramamurthy, "Survivable traffic grooming in WDM mesh networks under SRLG constraints," *Proceedings of ICC'05*, vol. 3, pp. 1751–1755, 2005.

[116] J.Y. Yen, "Finding the k-shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712-716, July 1971.

[117] E. Yetginer and E. Karasan, "Regenerator placement and traffic engineering with restoration in GMPLS networks," *Photonic Network Communications*, vol. 6, no. 2, pp. 139-149, Sept. 2003.

[118] H. Zang, C. Ou, B. Mukherjee, "Path-protection routing and wavelength assignment (RWA) in WDM mesh networks under duct-layer constraints," *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 248-258 2003.

[119] K. Zhu, H. Zang, and B. Mukherjee, "A comprehensive study on next generation optical grooming switches," *IEEE Journal on Selected Areas in Communications*, vol. 21, pp. 1173–1186, Sep. 2003.

[120] S. Zsigmond, G. Németh, and T. Cinkler, "Mutual impact of physical impairments and grooming in multilayer networks," *Proceedings of IFIP ONDM'07*, pp. 38–47, May 2007.

# Samenvatting (Summary in Dutch)

Optische netwerken die gebruik maken van Wavelength Division Multiplexing (WDM) technologie faciliteren het multiplexen van meerdere onafhankelijke golflengte kanalen in een optische vezel. Aangezien golflengte kanalen onafhankelijk van elkaar opereren op verschillende Gb/s, bieden WDM optische netwerken een enorme capaciteit (in de orde van enkele Tb/s), waardoor ze geschikt zijn voor toekomstige netwerken. Een lichtpad is samengesteld uit golflengte kanalen tussen de bron en de bestemming en dient om een grote hoeveelheid data te versturen. Routeren in WDM netwerken betreft het toewijzen van zowel paden als golflengtes, en heet Routing and Wavelength Assignment (RWA). In WDM netwerken zijn er twee essentiële RWA zaken die veel belangstelling van zowel onderzoekers als netbeheerders hebben gekregen.

1. Survivability (robuustheid/herstelvermogen): Lichtpaden in WDM netwerken transporteren een enorme hoeveelheid gegevens. Als een lichtpad breekt als gevolg van verschillende natuurlijke of door de mens veroorzaakte rampen, kunnen kostbare gegevens verloren raken. Het is daarom belangrijk dat het netwerk, middels opnieuw configureren, de communicatie kan hervatten communicatie.

2. Impairment-aware routing (routeren bij signaalverstoringen): Als een optisch signaal zijn pad doorkruist, komt het ruis en signaal verstoringen onderweg tegen. Deze verstoringen veroorzaken bitfouten, die het signaal onherkenbaar kunnen maken. Om de verstoringen tegen te gaan, moet het signaal worden geregenereerd op tussenliggende knooppunten. In tegenstelling tot traditionele RWA, houdt impairment-aware RWA rekening met het effect van verstoringen. Impairment-aware RWA omvat twee belangrijke elementen:

   (a) Impairment-aware pad selectie: hoe vind je een pad van de bron naar de bestemming?

   (b) Regeneratoren plaatsen: hoeveel regeneratoren zijn nodig en waar moeten we ze plaatsen in het netwerk?

De belangrijkste focus van dit proefschrift is het bestuderen van verschillende problemen in verband met survivability en impairment-aware RWA in WDM netwerken.

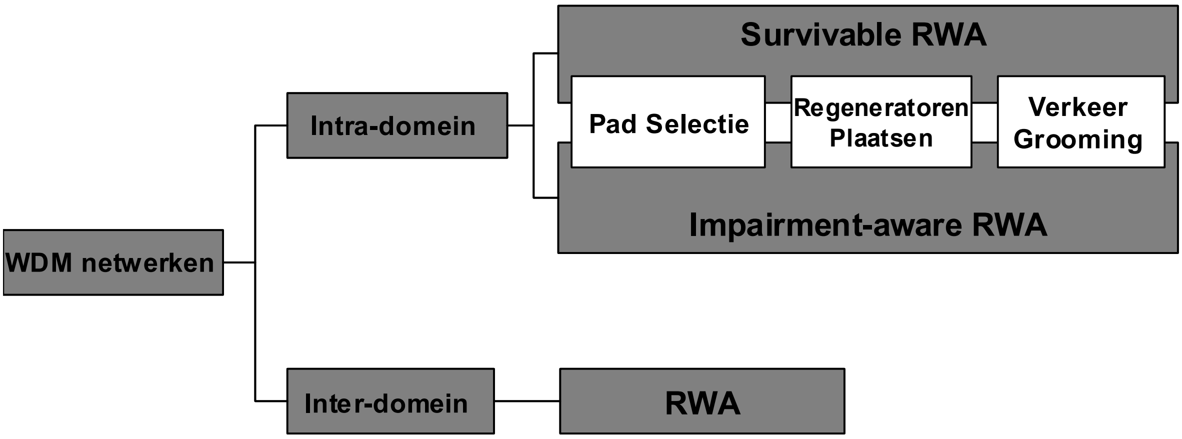



Figure F.1: Een overzicht van het onderzoek in dit proefschrift.

Figuur F.1 geeft een overzicht van het onderzoek in dit proefschrift. Het proefschrift beschouwt zowel intra-domein (dat wil zeggen, binnen in een enkel domein) als inter-domein (dat wil zeggen, tussen domeinen) RWA. Vooral voor intra-domein netwerken, maakt het een gedetailleerde studie van survivable en impairment-aware RWA kwesties, zowel afzonderlijk alsook gecombineerd. Zoals weergegeven in figuur F.1 zijn de belangrijkste onderwerpen (1) pad selectie, dat wil zeggen, het vinden van survivable en/of impairment-aware lichtpaden, (2) regenerator plaatsing voor onbeschermde en beschermde lichtpaden, en (3) survivable en/of impairment-aware verkeer grooming. Het groomen van verkeer helpt de beschikbare netwerkcapaciteit beter te benutten door de samenvoeging van meerdere onafhankelijke lage-snelheids verkeersstromen over hoge-snelheids lichtpaden. Voor de diverse problemen in dit proefschrift, is de complexiteit van de problemen in detail geanalyseerd, en zijn exacte, approximatie of heuristische algoritmes voorgesteld. Daarnaast is een case study van survivable en impairment-aware routeren uitgevoerd op een echt netwerk dat onderzoeks- en onderwijsinstellingen in Nederland met elkaar verbindt. Het werk in dit proefschrift zal niet alleen helpen inzicht te krijgen in de verschillende problemen in WDM netwerken, maar het kan ook toegepast worden op soortgelijke problemen in andere soorten netwerken, of zelfs in andere gebieden. Bijvoorbeeld, impairment-aware routeren heeft betrekking op het benzinestation probleem, waar gegeven een set van knooppunten (steden) met benzinestations, de vraag is om een route te vinden van de ene stad naar de andere zodanig dat een bestuurder niet met een lege tank strandt tussen benzinestations.

# Acknowledgements

This thesis presents the research work I have done for the last four and half years in the Network Architectures and Services (NAS) group of Delft University of Technology. I would like to acknowledge everyone who has helped me achieve this.

I would like to thank Prof. Piet Van Mieghem for giving me the opportunity to pursue my Ph.D. and for his supervision. I am highly indebted to my advisor, Dr. Fernando Kuipers, for all the help and assistance he has given me. His door has always been open for me and he is always ready to discuss and answer my questions. His thorough and dedicated guidance has made me a better researcher.

I have had the honor of collaborating with Prof. Ariel Orda, whose insightful comments and suggestions have contributed a lot to the several papers that we wrote together. I would like to thank him for that. I would also like to extend my thanks to Dr. Marcelo Yannuzzi, with whom I had the chance to work with during my study visit at the Technical University of Catalonia, which resulted in a paper on inter-domain routing in optical networks.

I am grateful to SURFnet, which funded this project. I also appreciate the opportunity I was given to present my work at the various Research on Networks (RON) meetings. My special thanks go to Dr. Roeland Nuijts and Dr. Richa Malhotra, who provided me with invaluable comments from a practical point of view.

I want to say "muchas gracias" to my friend Xavi for helping me a lot in designing the cover. I have had a fun and unforgettable time in the NAS group due to my friends and colleagues: Antonio, Siyu, Yue, Huijuan, Tom, Jasmina, Ebisa, Ruud, Cong, Dajie, Wynand, Norbert, Edgar, Christian, Rob, Jos, Stojan, Jinglong, Javad, Xueli, Jing, Bingjie, Gu, Chang, Przemek, Santpol, Rogier, and all others. Special thanks to my Ethiopian friends Elsa, Milliyon, Solomon (TU), Solomon (IHE), Eyasu, and Alemgena, who have made my stay in the Netherlands extra enjoyable.

The love and support I got from my family has sustained me through thick and thin. My parents have invested a lot for me to reach this point in my life. I know I can never pay them back in full, but I would like to express my appreciation and love. I also want to thank my siblings: Aster, Misrak, Dagim and Yewalashet. Last but not least, I would like to thank my wife, the love of my life. She has stood by me and helped me through the good and difficult times of my Ph.D.

# Curriculum Vitae

Anteneh Ayalew Beshir finished his B.Sc. in Computer Engineering at the Electrical and Computer Engineering Department of Addis Ababa University, Ethiopia, in 2000. He then worked as an assistant lecturer in the same department for 4 years. In September 2004, he came to the Netherlands to do his M.Sc. in Computer Engineering at Delft University of Technology. After finishing his two-year study, he joined the Network Architectures and Services (NAS) group of Delft University of Technology as a Ph.D. student in September 2006 under the supervision of Dr.ir. Fernando Kuipers and Prof.dr.ir. Piet Van Mieghem. His thesis work was funded by the Gigaport project of SURFnet, and focused on routing issues in optical networks, specifically on survivable and impairment-aware routing.

During his Ph.D. study, he was nominated for the best paper award at the International Teletraffic Congress (ITC 21), has guided four masters students, and has reviewed several international journal and conference papers.

## Publications

**P1:** F.A. Kuipers, A.A. Beshir, A. Orda, and P. Van Mieghem, Impairment-aware Path Selection and Regenerator Placement in Translucent Optical Networks, Proc. of ICNP 2010, the 18th IEEE International Conference on Network Protocols, Kyoto, Japan, October 5-8, 2010.

**P2:** A.A. Beshir, M. Yannuzzi, and F.A. Kuipers, Inter-domain Routing in Optical Networks with Wavelength Converters, in Proc. of IEEE ICC 2010, Cape Town, South Africa, May 23-27, 2010.

**P3:** A.A. Beshir and F.A. Kuipers, Variants of the Min-Sum Link-Disjoint Paths Problem, in Proc. of the 16th Annual Symposium on Communications and Vehicular Technology, Louvain-la-Neuve, Belgium, November 19, 2009.

**P4:** A.A. Beshir, F.A. Kuipers, A. Orda, and P. Van Mieghem, On-line Survivable Routing in WDM Networks, in Proc. of the 21st International Teletraffic Congress

(ITC 21), Paris, France, September 15-17, 2009 (nominated for **best paper award**).

**P5:** A.A. Beshir, F.A. Kuipers, A. Orda, and P. Van Mieghem, Sparse Regenerator Placement in Optical Networks, under submission.

**P6:** A.A. Beshir, F.A. Kuipers, A. Orda, and P. Van Mieghem, Survivable Impairment-aware Traffic Grooming in WDM Rings, under submission.

**P7:** A.A. Beshir, R. Nuijts, R. Malhotra, and F.A. Kuipers, Survivable Impairment-aware Traffic Grooming in WDM Networks, under submission.

## Relationship with Thesis

The following table describes the relationship of the afore-mentioned publications with the chapters in the thesis.

| **Pub.** | **Ch. 1** | **Ch. 2** | **Ch. 3** | **Ch. 4** | **Ch. 5** | **Ch. 6** | **Ch. 7** | **Ch. 8** |
|---|---|---|---|---|---|---|---|---|
| **P1** | | | X | X | | | | |
| **P2** | | | | | | | X | |
| **P3** | X | | | | | | | |
| **P4** | | X | | | | | | |
| **P5** | | | | X | X | | | |
| **P6** | | | | | | X | | |
| **P7** | | | | | | | | X |