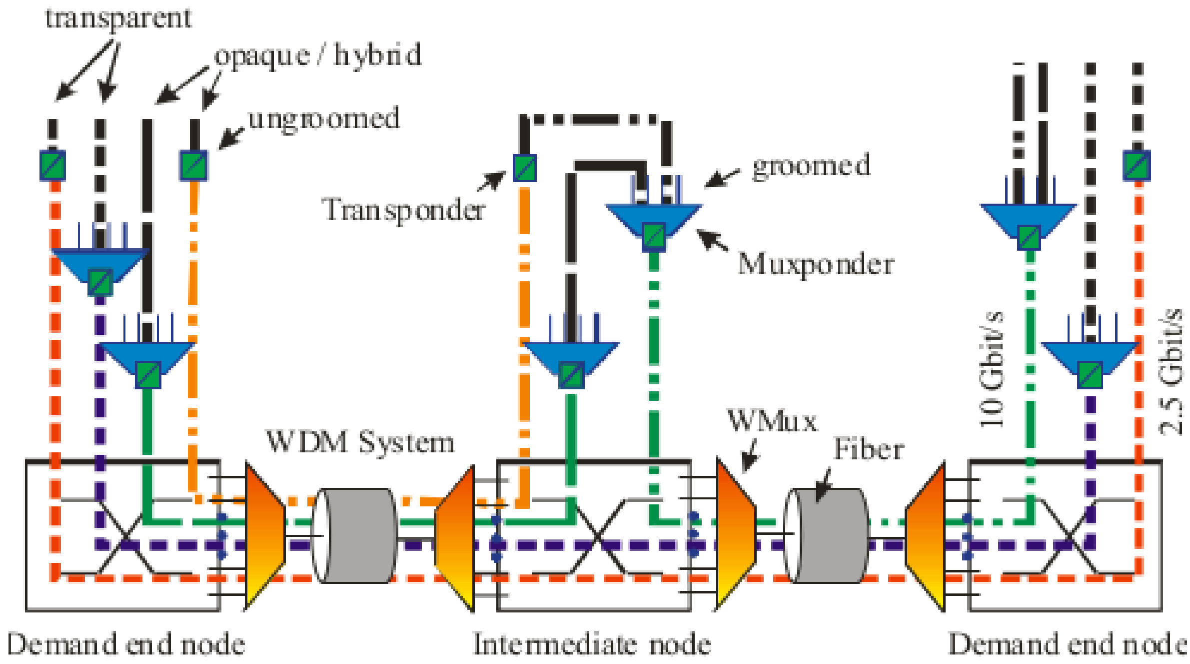


Bruna Antunes Nogueira

Regeneration and Grooming in MPLS over WDM Networks

Setembro de 2016

· U C ·

Universidade de Coimbra

· U C ·

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Regeneration and Grooming in MPLS over WDM networks

Bruna Antunes Nogueira

Coimbra, Setembro 2016

Regeneration and Grooming in MPLS over WDM networks

Supervisor:

Doutora Lúcia Maria Reis Albuquerque Martins

Co-Supervisor:

Doutor João Miguel Lopes dos Santos

Jury:

Doutora Rita Girão Silva

Doutora Lúcia Maria Reis Albuquerque Martins

Doutora Teresa Martinez dos Santos Gomes

Dissertation submitted in partial fulfillment for the degree of Master of Science in Electrical and Computer Engineering.

Coimbra, Setembro 2016

Acknowledgements

Gostaria de agradecer à Professora Lúcia Martins, por me ter convidado a participar neste projecto e pela sua orientação e apoio ao longo do desenvolvimento da tese.

Quero agradecer também ao Dr. João Santos, por estar sempre disponível para esclarecer qualquer dúvida.

Agradeço também à CORIANT Portugal, por terem oferecido uma bolsa para a realização deste projecto.

Quero também deixar uma palavra de agradecimento à Professora Teresa Gomes, por estar sempre disponível para esclarecimentos.

Queria agradecer à minha mãe, Adelina, e à minha irmã, Mariana, pelo apoio constante e incondicional ao longo de todo o meu percurso académico e pessoal.

Agradeço ao Ricardo, que esteve ao meu lado e me apoiou neste trabalho, e com quem posso sempre contar.

Por fim, queria agradecer aos meus amigos, pelo companheirismo e pela força.

Resumo

A evolução da tecnologia de multiplexagem por divisão no comprimento de onda (*Wavelength Division Multiplexing* – WDM) em redes óticas deu origem a um aumento contínuo nas capacidades das redes, como resposta ao crescimento exponencial do tráfego. Este aumento expressivo das capacidades reflectiu-se numa maior complexidade na electrónica associada à multiplexagem e comutação, tornando o custo da electrónica o custo dominante numa rede óptica. Particularmente, os custos envolvidos na agregação de tráfego (*grooming*) e regeneração, dominados pelo custo dos *transponders*, são os mais significativos. Tendo em conta que os pedidos de conexão feitos à rede podem exigir larguras de banda inferiores às do comprimento de onda, as técnicas de *grooming* emergiram no sentido de colmatar a discrepância entre as capacidades dos comprimentos de onda e dos pedidos de conexão. Assim, múltiplos pedidos são multiplexados em caminhos óticos (*lightpaths*) de maior capacidade, para um uso mais eficiente dos recursos da rede. Por outro lado, a regeneração do sinal ótico é necessária devido à degradação que este sofre à medida que se propaga. Desta forma, é necessário um posicionamento eficiente de *transponders* para *grooming* e regeneração, de modo a que todos os pedidos sejam satisfeitos com um custo mínimo.

Este trabalho aborda o problema de Agregação de Tráfego, Encaminhamento e Atribuição de Comprimento de Onda com Regeneração (*Grooming, Routing and Wavelength Assignment with Regeneration* – GRWAR) para redes emalhadas, em cenários de tráfego estático. Dadas a topologia da rede e a matriz de tráfego, o objectivo é encaminhar e agregar pedidos de estabelecimento de ligações de forma a minimizar o número de *transponders*. Para simplificar, o trabalho foca-se primeiramente no problema de Agregação de tráfego e Encaminhamento com Regeneração (*Grooming and Routing with Regeneration – GRR*), ignorando o problema de atribuição de comprimento de onda. Foram desenvolvidos três modelos de optimização usando Programação Linear Inteira (*Integer Linear Programming*–ILP), para redes dirigidas e não-dirigidas, bem como um modelo lexicográfico que minimiza o comprimento dos *lightpaths* para o número ótimo de *transponders*. É proposta uma heurística para o problema GRR como alternativa à abordagem ILP que, numa segunda fase, serve como base de desenvolvimento de duas heurísticas que abordam o problema GRWAR, em articulação com um trabalho sobre Encaminhamento e Atribuição de Comprimento de Onda com Regeneração (*Routing and Wavelength Assignment with Regeneration – RWAR*) desenvolvido no âmbito de outra tese de mestrado.

Os desempenhos da formalização ILP e da heurística no problema GRR foram comparados para redes pequenas, tendo a heurística obtido bons resultados em tempos razoáveis - obtiveram-se um desvio máximo de 11% e um desvio médio de 6% em relação ao número ótimo de *transponders*. No que diz respeito às heurísticas para o problema GRWAR, os resultados indicam que uma das duas heurísticas pode ter um desempenho potencialmente

superior ao da outra para ocupações da rede elevadas, apesar da amostra de resultados não ser suficientemente grande para suportar estatisticamente tal conclusão. Por essa razão, devem ser efectuadas mais experiências de modo a obter conclusões sólidas no que respeita aos benefícios de usar cada uma das heurísticas.

Palavras-Chave

Caminhos mais curtos, Agregação de Tráfego, Regeneração, Transponder, Programação Linear Inteira, Heurística, WDM

Abstract

The deployment and maturing of Wavelength Division Multiplexing (WDM) technology in optical networks has allowed for an increase in network capacities, in response to the exponential growth of demand in communications. Such a massive increase in network capacities translated into greater electronic multiplexing and switching efforts, consequently making the cost of the electronics the dominant cost in a network. Particularly, the costs involved in traffic grooming and regeneration, dominated by transponders, are prevalent. As connection requests to the network typically require sub-wavelength data rates, traffic grooming techniques have emerged to bridge the gap between wavelength channel capacities and connection requests, in which multiple lower-speed traffic streams are multiplexed onto the high-speed wavelength lightpaths, for more efficient use of both network capacity and resources. On the other hand, the regeneration of the optical signal is mandatory due to the degradation it suffers as it propagates through the optical networks facilities. Thus, an efficient placement of transponders for electronic grooming and regeneration is necessary so that all the demands are carried with minimum cost.

This work addresses the Grooming, Routing and Wavelength Assignment with Regeneration (GRWAR) problem for mesh networks, in static traffic scenarios. For a given network topology and a traffic matrix, the goal is to route and groom connection requests in a way that minimizes the number of transponders. To reduce complexity, this work focuses primarily on the Grooming and Routing with Regeneration (GRR) problem, which disregards the wavelength assignment problem. Three Integer Linear Programing (ILP) variants were developed, for directed and undirected networks, as well as a third lexicographical optimization that minimizes the cost of lightpaths for the optimal number of transponders. A GRR heuristic is proposed as an alternative to the ILP approach. The GRR heuristic then serves as a base to two heuristics proposed to address the GRWAR problem, in articulation with a work on Routing and Wavelength Assignment with Regeneration (RWAR) developed in the context of another MSc. thesis.

The ILP and the heuristic for the GRR problem were compared for small networks, with the heuristic providing good results in reasonable times. A maximum deviation of 11% from the optimal number of transponders was observed, with the average difference being of 6%. Regarding the heuristics addressing the GRWAR, results indicate that one of the two heuristics can potentially be superior to the other for high network loads, although the sample of results is not large enough to statistically support such conclusion. For that reason, more experiences must be carried out in order to obtain solid conclusions about the benefits of using each of the approaches.

Keywords

Shortest Path, Traffic Grooming, Regeneration, Transponder, ILP, Heuristic, Wavelength Division Multiplexing

"We really thought we had a purpose
We were so anxious to achieve,
We had hope, the world held promise
For a slave to liberty "

— Joni Mitchell, *Come In From The Cold*

Contents

List of Acronyms

MPLS	Multiprotocol Label Switching
GRR	Grooming and Routing with Regeneration
WA	Wavelength Assignment
RP	Regenerator Placement
RRP	Routing with Regenerator Placement
RWA	Routing and Wavelength Assignment
RWAR	Routing and Wavelength Assignment with Regeneration
GR	Grooming and Routing
GRWA	Grooming, Routing and Wavelength Assignment
GRWAR	Grooming, Routing and Wavelength Assignment with Regeneration
ILP	Integer Linear Programing
WDM	Wavelength Division Multiplexing
DWDM	Dense Wavelength Division Multiplexing
ROADM	Reconfigurable Optical Add-Drop Multiplexer
OADM	Optical Add-Drop Multiplexer
OEO	Optical-electrical-optical
NOBEL	Next Generation Optical Network for Broadband European Leadership
OE	Optical-electrical

EO	Electrical-optical
SONET	Synchronous Optical Networking
LSP	Label Switched Path
CAPEX	Capital Expenditures

List of Figures

List of Tables

List of Algorithms

1 Introduction

The deployment and maturing of Wavelength Division Multiplexing (WDM) technology in optical networks has allowed for an increase in network capacities, in response to the exponential growth of demand in communications [1]. An optical network is composed of several nodes inteconnected with fiber optical links, with the WDM technology enabling the simultaneous transmission of traffic on different non-overlapping wavelength channels within the same fiber. In more recent advances, Dense Wavelength Division Multiplexing (DWDM) technology allows for up to 160 wavelength channels per fiber [2]. As traffic propagates through optical links at different wavelengths, the network can be alternatively thought of as a set of nodes connected by lightpaths. A lightpath is a logical connection that consists of a path of optical links with an allocated wavelength in each link.

Such a massive increase in network capacities translated into greater electronic multiplexing and switching efforts, consequently making the cost of the electronics the dominant cost in a network [3]. Particularly, the costs involved in traffic grooming and regeneration, dominated by transponders, are prevalent [4, 5]. As connection requests to the network may require sub-wavelength data rates, traffic grooming techniques have emerged to face the gap between wavelength channel capacities and connection requests, in which multiple lower-speed traffic streams are multiplexed onto the high-speed lightpaths, for more efficient use of both network capacity and resources. On the other hand, the regeneration of the optical signal is mandatory due to the degradation it suffers as it propagates through the optical network's facilities [6]. The generation and termination of a lightpath require the use of transponders in its originating and terminating nodes. When the lightpath is electronically terminated at a transponder, the optical signal is converted to electronic domain for further processing, for instance, in an Multiprotocol Label Switching (MPLS) router, for the case of MPLS over WDM networks. Once the signal is in the electronic domain, it has access to functionalities such as regeneration, wavelength conversion and traffic grooming. As for regeneration, although the development of all-optical regeneration is in course, electronic regeneration is still the most economic and reliable [7]. Electronic regeneration of a lightpath can be obtained by electronically terminating traffic at a transponder or by a regenerator. While the first approach allows the optical signal to be demultiplexed and have the lower-speed connection requests groomed and switched separately, in the latter, the process is carried exclusively at the wavelength granularity. Traffic grooming alone has the potential to greatly reduce network costs, since multiple connections can share both the bandwidth of lightpaths and, as a consequence, transponders [8]. However, it has been shown that there is a cost advantage in approaching both grooming and regeneration simultaneously [4], since strategical locations for traffic grooming can be automatically used for regeneration purposes and, reciprocally, regeneration needs create opportunities for traffic grooming, facts that sequential network planning approaches overlook.

The issues of traffic grooming, routing, wavelength assignment and regeneration are some of the most important regarding the design of optical networks [5,8]. The problem of grooming and routing consists of determining a set of lightpaths and how to efficiently route and group connection requests over those lightpaths. On the other hand, the wavelength assignment problem consists of assigning wavelengths to lightpaths. Regarding regeneration, the goal is to assure that an optical signal does not degrade above an unrecoverable level. Approaches to traffic grooming differentiate on whether these problems are solved separately or as a whole [8,9,10], addressing the Grooming, Routing and Wavelength Assignment with Regeneration (GRWAR) problem. When taking into account all the constraints involved in the different problems, potentially better results can be achieved. However, the problem is rather complex (NP-Complete), which makes the decomposition into sub-problems an attractive answer to reduce complexity.

1.1 Objectives

The traffic grooming problem, with or without physical impairment constraints, is usually addressed through two main approaches. Integer Linear Programing (ILP) formulations are commonly developed in order to achieve the optimal solution for a given goal. However, they can only optimally solve problems for small network topologies – which are not representative of deployed networks – in reasonable times. Therefore, the second approach comes as natural consequence to this fact, which consists of the development of heuristics seeking good results in a reasonable amount of time.

This work addresses the GRWAR problem for mesh networks, in static traffic scenarios. For a given network topology and traffic matrix, the objective is to route and groom connection requests in a way that minimizes the number of transponders.

The main focus of this work is the Grooming and Routing with Regeneration (GRR) subproblem, which disregards wavelength assignment and consequently the wavelength continuity constraints at the physical level, in order to reduce the complexity of the problem. In this context, the goals of this work are the development of an ILP formulation of the problem, as well as an alternative heuristic for realistically sized networks.

In a later stage, a third goal is to integrate the work in a multi-layer approach, articulating the work developed for the GRR problem with a work on Routing and Wavelength Assignment with Regeneration (RWAR) developed in the context of another Msc. thesis [11], where the combined approach must address the GRWAR problem.

1.2 Main Contributions

In this work, three ILP variants addressing the GRR problem were formulated. These include models for directed networks, undirected networks with bidirectional symmetrical traffic, and a third lexicographical optimization that minimizes the cost of lightpaths (in terms of length) while requiring the optimal number of transponders. A heuristic for the GRR problem was developed as an alternative to the ILP approach. For small networks, it was observed that the heuristic provides an interesting trade-off between the number of transponders and running times. Moreover, it presents interesting results regarding other evaluated metrics.

Regarding the GRWAR problem, two heuristics were developed and compared. A reference heuristic, that consists of the heuristic developed in the first stage followed by a Wavelength Assignment (WA) procedure (GRR-WA heuristic), and a second proposed heuristic which also consists of the heuristic developed in the first stage integrated with a heuristic for the RWAR problem developed in the context of another Msc. thesis [11] (RWAR-Grooming and Routing (GR) heuristic). Results suggest that, for high network load, there may be advantages in using the RWAR-GR approach, although that is not verified for every test, and it is important to carry out more experiences in order to obtain solid conclusions about the benefits of using each of the approaches.

1.3 Document Structure

This document is organized in five chapters. The following chapter presents a brief overview of the research done so far on the problem of traffic grooming in WDM networks. Chapter 3 addresses the developed ILP formulations and heuristics. It begins by presenting the three developed variants of the ILP formulation for the GRR problem, considering directed and undirected networks, and a further lexicographical optimization that achieves the optimal number of lightpaths with minimum cost. As an alternative, a heuristic for the GRR problem is proposed. Finally, two heuristics for the GRWAR problems are presented. In chapter 4, the performance evaluation of the developed heuristics is discussed, where the GRR heuristic is compared with the ILP model and the heuristics addressing GRWAR are compared with each other. Chapter 5 addresses the main conclusions and future work.

2 State Of The Art

2.1 Basic Concepts and Terminology

The purpose of this section is to introduce and define common concepts regarding optical networks and traffic grooming.

WDM networks are composed of nodes interconnected by optical fiber-links, where several non-overlapping high-rate wavelength channels are multiplexed into each fiber link. Today, with DWDM technology, up to 160 wavelengths can be available in an optical fiber, enabling aggregated rates of several Tbps per fiber [2]. Currently, some nodes of optical networks can be Optical Add-Drop Multiplexers (OADMs), which allow the incoming traffic, in a given wavelength, to either be dropped at the node for electronic processing or to optically bypass it. In cases where incoming traffic does not need to be dropped at a node, the optical signal is switched from one wavelength in the incoming link to the same wavelength in the outgoing link, and everything is processed at the granularity of a wavelength. If the WDM signal on an incoming fiber is dropped at a node, it is terminated at an optical terminal where it is demultiplexed into the different wavelength channels, after which each wavelength might terminate at a transponder that performs an Optical-electrical (OE) conversion of the signal. Analogously, for outgoing links, an electrical signal is directed to a transponder where it undergoes a Electrical-optical (EO) conversion and is sent out in a wavelength channel, which is multiplexed with other signals to form the WDM signal that is sent into the outgoing fiber. A transiting signal can be both added and dropped at the node, which is handled by two back-to-back transponders inter-connected in the node. A pair of inter-connected transponders can operate at different wavelengths, and thus provide wavelength conversion [12].

In MPLS over WDM networks, transponders connect to MPLS routers which perform the necessary electronic processing and switching of the signal.

An optical signal is dropped at a node if the traffic is destined to that node, but also if signal regeneration or wavelength conversion are required, or if it is to be groomed with other traffic. The need for regeneration of the optical signal is imposed because its propagation through optical nodes and fiber-links introduces both linear and nonlinear impairments [6], thus degrading it. Above a certain level of degradation, the receiver is unable to recognize the information carried by the signal.

These impairments are usually addressed through electronic regeneration, which is obtained by dropping the optical signal at a node where it undergoes an Optical-electrical-optical (OEO) conversion. This conversion can be addressed using transponders connected to routers, in which case the traffic in the optical signal can be further processed, or through the use of regenerator cards, which operate at the granularity of the wavelength and only provide regeneration and wavelength conversion [4, 5]. Although the development of all-optical regenerators is in course, electronic regeneration is still the most economic and reliable [7]. Wavelength conversion may be necessary to avoid routing collisions. When a signal is optically bypassed, the incoming wavelength must be the same as the outgoing wavelength, which is known as the wavelength continuity constraint. If the signal must be routed through a link where that wavelength is not available, wavelength conversion can be performed so that the signal can use a free wavelength in that link.

The grooming of traffic may also be of interest, as connection requests usually require sub-wavelength data rates, and assigning the full capacity of each wavelength channel in each link to a single request can be extremely wasteful of that bandwidth. In order to efficiently use the available bandwidth and network resources, traffic grooming techniques have to be considered where lower-speed traffic streams are multiplexed onto high-speed wavelength channels, which is the common scenario for MPLS over WDM networks.

The massive increase in network capacities provided by the deployment of DWDM transmission technology calls for much greater electronic multiplexing and switching efforts, with the current dominant cost residing in electronics rather than in optics [3]. Particularly, electronic equipment involved in regeneration and grooming is the most expensive, with transponders representing the dominant cost [4, 5]. Since the traffic that passes through a given node in a given wavelength is, in many cases, neither originated nor destined to that node, it is unnecessary and costly to have each wavelength at each node electronically processed. Network nodes are often subdivided in three categories, according to their transparency [13]: opaque, transparent and translucent nodes. In an opaque node, traffic switching takes place exclusively in the electronic domain, i.e., the traffic of every wavelength has to be dropped at that node to be switched, being automatically regenerated and having access to functions of wavelength conversion and sub-wavelength traffic grooming. In a transparent node, all the switching is done in the optical domain. In a translucent node, some of the optical channels are switched in the optical domain and others in the electronic domain, presenting selective transparency.

According to this categorization, the concept of transparency is often extended to define three types of network architectures, with a transparent network being one in which all nodes are transparent, an opaque being one where all nodes are opaque, and a translucent being one that may have nodes of all three categories. Optical networks are transitioning from traditional opaque networks to transparent networks [5], but there are still issues that have to be addressed electronically, like signal regeneration, for instance. Translucent networks provide a compromise between opaque and transparent networks by combining the strengths of both architectures, typically reducing over 60% of the required transponders when compared to an opaque architecture, showing advantages in cost, space, power, and heat dissipation [12]. Note that in an opaque network, two transponders are necessary for each wavelength of each link.

Regardless of the architecture, the optical network can be thought of as having two layers [8]:

- a physical layer, which consists of the nodes and fiber links of the network.
- a virtual layer, which consists of lightpaths and the nodes connected by them.

The concept of lightpath varies among research works, but it is generally defined as a logical connection between two nodes that corresponds to a physical path in the optical network and carries a bandwidth equal to that of a wavelength in each link. Some definitions admit that a lightpath may use one or more wavelengths throughout its physical route, if wavelength conversion in the intermediate nodes is available [14]. A lightpath can also be defined in terms of transparency. A transparent lightpath is one in which the optical signal is optically bypassed from source to destination, and thus the constraint of wavelength continuity must be assured from source to destination, as well as the maximum distance it can travel without regeneration. On the contrary, a translucent lightpath, or simply referred to as lightpath, refers to one in which wavelength conversion and/or regeneration is possible in intermediate nodes of its route. A translucent lightpath is a sequence of transparent segments where the wavelength continuity constraint has to be assured [7].

Concerning traffic grooming, there are two common modes of low-rate traffic grooming: single-hop and multi-hop traffic grooming [15]. In single-hop traffic grooming, a traffic request is routed through a single lightpath from source to destination, and in multi-hop traffic grooming, it is routed through a sequence of lightpaths, possibly being groomed with different connections requests along its route.

2.2 Literature Review

As already mentioned, network cost is currently dominated by the cost of transponders. For that reason, it is important that this equipment is efficiently used, such that traffic can be supported with minimum cost. Traffic grooming can be of great help by allowing different connection requests to share both transponders and wavelength capacities. If traffic is dropped for regeneration instead of using regenerator cards, both grooming and regeneration can be made in the same place. Thus, as regeneration presents itself as a necessity, it can create opportunities for traffic grooming and, on the other hand, places of interest for traffic grooming provide signal regeneration. The strategic placement of transponders and regenerators is commonly addressed as a problem of translucent network design, where the goal is to obtain a translucent network that can achieve a performance equivalent to that of an opaque network while using a much smaller amount of strategically placed transponders/regenerators. Two approaches stand out [7]: translucent networks with sparsely placed opaque nodes, where only a subset of nodes provide wavelength conversion, regeneration and traffic grooming for all wavelengths, and translucent networks with translucent nodes, where the objective is to minimize the total number of transponders through the strategical placement of such equipment at a wavelength basis. While the first minimizes the number of active nodes, the second minimizes the total number of transponders in the network, which is more adequate if the objective is to minimize the cost of transponders. Traditionally, the problems of traffic grooming and regenerator placement are handled sequentially, although there are a few recent works that explore the combination of both [2, 4, 16].

In previous works concerning optical networks, the grooming of traffic was ignored, assuming a traffic demand took up an entire wavelength channel. The problem of routing, assigning wavelengths and network resources to traffic demands under such assumption is well known as the Routing and Wavelength Assignment (RWA) problem [15]. The RWA problem applies to transparent networks and, in its pure form, does not account for wavelength conversion [13]. Demands must be routed end-to-end in a way such that if the physical paths assigned to them share an edge, the wavelengths assigned to those paths are different [13]. Typically, the metrics to minimize were the number of wavelengths, congestion, or a combination of the two [17]. When grooming is considered, the problem can be divided into a series of subproblems [4], which are solved in order to meet a network design or operational goal:

1. Finding a virtual topology with a set of lightpaths.
2. Routing of traffic demands in the virtual topology.
3. Routing and wavelength assignment of lightpaths over the physical topology.

Most studies on the traffic grooming problem deal with all of the mentioned subproblems, addressing the Grooming, Routing and Wavelength Assignment (GRWA) problem. However, some works focus only on the virtual layer subproblems [10], addressing the GR problem, and some study both the complete problem and some of the subproblems individually [8]. Note that the routing part of the GR problem can refer to routing only in the virtual layer or go deeper into the physical layer, disregarding only the wavelength assignment. When such problem is mentioned, the context of routing will be specified. Moreover, as works on both impairment-aware and non-impairment-aware traffic grooming are referred, the problems RWA, GRWA and GR with regeneration constraints are herein referred to as RWAR, GRWAR and GRR. The RWA problem is NP-complete [4], and since it is an integrating part of the GRWA/GRWAR problems, these problems are also NP-complete [4]. Much of the work in traffic grooming attempts to solve the problem through two main approaches [16]: ILP formulations or heuristic approaches. The first consists of the formulation of optimization models that, although providing optimal solutions, are only applicable to small problem instances. The other approach focuses on the development of heuristics that are able to solve larger problems in a reasonable amount of time, reducing complexity at the expense of the quality of results. In terms of objectives, there are two main interests on traffic grooming research: throughput maximization for existing networks or network design with minimization of resources or costs for given topologies and traffic matrices [2].

Early research on traffic grooming focused on ring topologies due to the time's widespread use of Synchronous Optical Networking (SONET) ring-based networks, representing the most immediate practical interest [3]. This work focuses on the problem of traffic grooming for mesh topologies in a static traffic scenario and, as such, approaches considering ring networks and dynamic traffic models are not explored. For more information, two comprehensive surveys on the traffic grooming subject that include detailed reviews on works focusing on ring networks can be found in [13] and [3].

The growth of Internet traffic led to an increasing number of networks arranged in mesh topologies [3], and as such the focus of traffic grooming research has recently transitioned to mesh networks. It has been shown that mesh topologies present a compelling cost advantage, are more resilient to network failures, and are more efficient in accommodating traffic changes (see [18], [19] and references therein) when compared to ring networks.

Zhu and Mukherjee [15] address the GRWA problem in a WDM mesh network considering a static traffic scenario, where the objective is the maximization of network throughput. In this work, it is not required that all demands are satisfied. It is further assumed that network nodes do not provide wavelength conversion capability and that connection requests cannot be divided and diversely routed through multiple lower-speed connections (splitting). A couple of variants of the problem are formulated as ILP optimization problems: assuming single-hop traffic grooming, multi-hop traffic grooming, and the use of wavelength tunable or fixed transceivers at the network nodes. The performances of single and multi-hop traffic grooming approaches are compared on a small network with six nodes and with randomly generated traffic. Results show that multi-hop leads to higher throughput than single-hop and that, for network throughput maximization, connections between the same source and destination nodes tend to be packed together in an end-to-end single-hop lightpath. Based on this observation, they propose two heuristic algorithms which assign single-hop groomable connections the highest priority, differing mainly in the metric evaluated for the order in which connection requests are served. In one of them, the connections between the node pairs with higher aggregate traffic to be carried are served first, whereas in the other, lightpaths are established first for packed connections with higher resource utilization values. Both heuristic algorithms present reasonable performance when compared to the ILP solutions.

Hu and Leida [8] also focus on the GRWA problem for a WDM mesh network, but with the objective of minimizing the number of transponders. A complete ILP formulation for the GRWA problem is presented, but since the problem becomes computationally infeasible for large networks, a decomposition method is proposed to reduce its complexity. The GRWA problem is divided into two sub-problems: the grooming and routing problem GR, where the grooming and routing (in the physical links) of traffic demands are considered with the objective of minimizing the number of lightpaths, and the wavelength assignment problem WA, which goal is to find a feasible wavelength assignment solution given the WA capacity constraints derived in the GR problem. Although the decomposition method leads, in general, to approximate solutions, the authors provide a sufficient condition under which the obtained solution is optimal. A relaxation of some of the integer constraints in the GR problem is done in order to further reduce complexity. Results show that the decomposition method with the relaxed GR ILP produces good results in reasonably small times.

Konda and Chow [10] also propose to minimize the number of transponders. It is assumed that, between any pair of nodes, any number of lightpaths can be implemented. The work focuses only on the virtual topology, concentrating on the GR problem, and ignores physical topology details. Full-duplex traffic is ensured at the virtual topology level. An ILP formulation of the transponder minimization problem is first introduced, and a heuristic is proposed to be used for large networks. The heuristic is based on a duality property that transforms the problem into an equivalent one. It assigns a large set of lightpaths to carry the given traffic demands (at least the known upper bound on the number of lightpaths between each node pair), beginning by assigning end-to-end lightpaths to every connection request. In a second stage, the heuristic reroutes as many lightpath streams through surplus capacities in the network as possible, deleting lightpaths with increasing occupied bandwidth as long as there is still enough surplus capacity in the network for such rerouting. The results are near the optimal. For large networks, results lie between known bounds.

Zhu et. al [20] focus on the design of a WDM sparse-grooming mesh network, for a static traffic scenario, where only a fraction of the nodes possess grooming capability - the grooming nodes (or opaque nodes). A grooming node is capable of grooming a given amount of connections for every wavelength. Thus, this is a problem of design of a translucent network with opaque nodes. The aim of this work is to strategically position a restricted number of nodes with grooming capabilities in the network in a way that the resulting network performance is similar to that of a full-grooming network, thus reducing network cost. An ILP formulation of the problem is presented, considering two alternative objectives: maximizing network throughput for a given amount of grooming nodes (Ng) and number of wavelengths (W) per fiber, or minimizing network cost while carrying all network requests, where Ng and W become variables. A two-step heuristic is presented, which starts by the selection of Ng nodes as grooming nodes according to the evaluation of a cost function for each node, and follows with the routing of traffic requests on the network while obeying resource limitations. The requests are maintained on a request list where the entries are randomly permuted N times, generating N lists. For each list, the traffic requests are serviced sequentially. For the second objective function, the heuristics can start with small values for Ng and W and increase them gradually until all requests are satisfied. Results show that the efficient selection of a limited number of grooming nodes can lead to good network performance and significant cost reductions.

In [9], a generic graph model for traffic grooming in heterogeneous WDM mesh networks is proposed, which can be used for both static and dynamic traffic scenarios. Parameters as the number of transceivers at each node, the number of wavelengths in each fiber link, the grooming and wavelength-conversion capabilities of each node are represented as edges of an auxiliary graph. Through careful weight assignment to these edges, using different grooming policies, different objectives (such as minimizing the number of traffic hops, minimizing the number of lightpaths and minimizing the number of wavelengths) can be attained simply by using shortest path routing.

The previous works do not account for physical impairments, and as such the problem of regeneration would be addressed in sequence. A few recent studies have addressed the problem of impairment-aware traffic grooming in WDM networks [2,4,16] where the problem is to route traffic and lightpaths and to place regenerators and electronic grooming equipment in a way that minimizes network cost. Although there are multiple sources of impairments [21], for a design problem like regenerator placement, a single impairment metric is sufficient, which could correspond to the worst of all impairments, or alternatively the metric of distance as it represents a determinant role in the signal quality [22].

Patel et. al [4] show that combining the grooming problem with the placement of regenerators reduces the network cost significantly when compared to the case where grooming and regenerator placement are treated independently. In that work, the GRWAR problem is addressed. A detailed Reconfigurable Optical Add-Drop Multiplexer (ROADM) node architecture and the associated cost model are considered. It is assumed that regeneration can be obtained either through the placement of regenerators (regenerator cards) or electronic grooming equipment. An ILP model for a directed network is formulated. An auxiliary-graph based heuristic is proposed for non-blocking scenarios. A threshold in aggregated bandwidth is used to decide whether each given demand is a good candidate to be groomed with others, and thus determines whether grooming equipment or only a regenerator should be used when regeneration is needed. The heuristic uses several values for the threshold and returns the result which yields the minimum network cost. The performance of the heuristic is evaluated through direct comparison with the ILP solutions for small networks, and with a derived lower bound for larger networks. It is further compared with three other heuristics, which handle grooming-only, regeneration-only, and sequential grooming and regeneration. The proposed heuristic outperforms the remaining heuristics, falling within a 2% error from the optimum, for small networks. For larger networks, the proposed heuristic outperforms the remaining heuristics again, with the performance staying close to the lower bound and approaching it as the network load increases.

Scheffel et. al [16] address the GRWAR problem, proposing a "path over path" concept supported by a three-layer network model which is integrated in an ILP formulation for the minimization of Capital Expenditures (CAPEX) in network design. The costs of several equipments are considered, based on a model defined by the European research project Next Generation Optical Network for Broadband European Leadership (NOBEL). The ILP formulation accounts for protected and unprotected traffic, assumes bidirectionality for every path, and solves grooming and routing in the physical layer. The wavelength assignment is performed in a subsequent optimization step by another ILP. Scalability of the problem size is enabled through the limitation of the solution space by a priori selection of eligible grooming and physical paths. Results show that resource limitations and different cost ratios between transponders and muxponders significantly influence solutions.

Plunkte et. al [2] also study the GRWAR problem, in an attempt to minimize network design cost, formulating three ILP formulations, one for each of the network architectures: opaque, transparent and translucent networks, having in mind the cost implications of the equipments used in each of them. All three formulations address both unprotected and protected traffic.

3 Traffic Grooming over WDM Networks

3.1 Notation

Let the graph $G = (N, A)$ represent a physical network, where N and A are the sets of nodes and arcs in the network, respectively. Throughout this work, the terms graph and network will be used interchangeably.

A directed arc connecting two nodes m and n, with $m, n \in N$ and $m \neq n$, is represented as the ordered pair (m, n), and its cost is given by c_{mn} and it is an additive cost in the context of this work. The arc adjacency list of node m, which consists of the set of outgoing arcs of that node, is defined as $A^+(m) = \{(m, n) \in A : n \in N\}$, and thus the outdegree of m equals $|A^+(m)|$. Analogously, the set of incoming arcs to node m is represented as $A^-(m) = \{(n, m) \in A : n \in N\}$, and the indegree of m is given by $|A^-(m)|$. An arc can support W wavelengths with each one providing a capacity of C Gbps.

A path p in G is defined as an alternate sequence of nodes and arcs. A path originating at node i and terminating at node j can be represented in the form $p_{ij} =< i, (i, k), k, \cdots, m,$ $(m, j), j >$, with $i, j, k, m \in N$. The sets of nodes and arcs of p_{ij} are represented as $N(p_{ij})$ and $A(p_{ij})$. The cost of a path is given by the sum of the costs of the arcs which compose it, and it's denoted by $c(p_{ij}) = \sum_{(m,n) \in A(p_{ij})} c_{mn}$.

The concatenation of two paths is represented as, $p_{ij} = p_{ik} \diamondsuit p_{kj}$. This operation results in a larger path p_{ij} corresponding to the union of the operand sub-paths without the repetition of the last node of the left operand and first node of the right operand.

The low-speed connection requests are given in a traffic matrix $\Lambda = \{\Lambda_y\}$, that is, Λ is a set of traffic matrices of distinct service classes characterized by their bandwidths, which are denoted by y. The set of demands between a source node s and a destination node d is represented as Λ_{sd}. The subset of Λ_{sd} which only contains demands of granularity y is denoted by $\Lambda_{y,sd}$. A specific demand is represented as $\Lambda^v_{y,sd}$, with v meaning that this is the v^{th} demand of the set $\Lambda_{y,sd}$, $v \in \{1, \cdots, |\Lambda_{y,sd}|\}$.

The term lightpath will be used to refer to a translucent lightpath, a communication channel with a bandwidth equal to a full wavelength, at the optical layer [14], which may use different wavelengths throughout its route (thus relaxing the traditional wavelength continuity constraint), using wavelength converters, if necessary. A lightpath can carry any bandwidth less or equal to C. The set of lightpaths originating at node i and terminating at node j is given by l_{ij}, with l_{ij}^{t} referring to the t^{th} lightpath of l_{ij}, $t \in \{1, \cdots, |l_{ij}|\}$. A lightpath is defined by an optical path together with an associated occupied capacity $l_{ij}^{t} = (p_{l_{ij}^{t}}, o_{l_{ij}^{t}})$, with $0 \leq o_{l_{ij}^{t}} \leq C$. The concatenation of lightpaths is represented as $l_{ik}^{t_1} \textcircled{\diamond} l_{kj}^{t_2}$ and stands for an ordered sequence of lightpaths. Let $l(mn)$ represent the set of translucent lightpaths in the arc (m, n).

A transparent lightpath is denoted as $l_{ij,\lambda}^{k}$, where λ is its assigned wavelength, and is analogously defined by an optical path together with an associated occupied capacity $l_{ij,\lambda}^{k} = (p_{l_{ij,\lambda}^{k}}, o_{l_{ij,\lambda}^{k}})$, with $0 \leq o_{l_{ij,\lambda}^{k}} \leq C$.

In the context of this work the optical signal regeneration problem is reduced to the simplest case where the considered impairment is only due to the distance of propagation, and thus the optical signal needs regeneration after traveling a distance equal to the impairment threshold, Δ, in kilometers .

If the arcs in A are undirected, they consist of unordered pairs of distinct nodes $[m, n]$. In this case, there is no distinction between incoming and outgoing arcs of a node. An undirected arc may also be referred to as a link. An undirected graph can be converted to a directed graph through the replacement of each undirected arc by two symmetrical directed arcs.

3.2 The GRR Problem

3.2.1 Problem Definition

The purpose of this section is to approach the GRR problem. The aim is to minimize the total number of transponders connected to MPLS routers while granting the fulfillment of all the requests in a given set of static traffic demands, respecting the regeneration impairment threshold Δ. It is assumed that both optical signal regeneration and wavelength conversion capabilities are obtained through transponders. For signal regeneration purposes, two additional transponders must be used when the optical signal has traveled at most Δ. The minimization of the number of transponders must take combined advantage of both the grooming of low-speed connection requests into high-capacity lightpaths and the lightpaths regeneration needs. In this context, it is considered that a transponder operates in both directions of a wavelength in an optical link.

This work focuses solely on the grooming problem, leaving aside the wavelength assignment problem. Thus, the wavelength continuity constraint is not imposed, with the demands being routed through translucent lightpaths. At the optical level, it is considered that only the number of free wavelength channels in each fiber is known.

It is further considered, regarding the topology and technological requirements of networks this study applies to, that the nodes of the optical network consist of ROADMs which are collocated with MPLS routers in the network. A pair of optical nodes is connected through a single (bidirectional) optical link with W wavelengths and one fiber for each direction. The distance between any two optical nodes is known. Regarding traffic grooming, it is assumed that a connection request cannot be divided into a set of diversely routed lower-speed connections, i.e., the traffic cannot be splitted.

Being in a static traffic context, it is assumed that the networks are able to provide the necessary capacity to route the given set of traffic demands, for which reason this study focuses on non-blocking scenarios.

3.2.2 ILP Formulation

ILP formulations for both directed and undirected networks are presented next. Furthermore, a formulation with a lexicographical method that minimizes the length of lightpaths for the optimal number of transponders is also presented.

Notation

In order to provide a clearer understanding of the mathematical formulation of the problem, the following notation is used:

For a given directed network $G = (N, A)$,

- m and n denote the end nodes of an optical link. The two constituent fibers are represented by two symmetrically directed arcs.
- i and j denote the origin and termination nodes of lightpaths.

- s and d specify the source and destination nodes of an end-to-end low-speed connection request.

Having in mind the previous notation, the problem formulations are presented below. Given the inputs $G = (N, A)$, Λ, c_{mn}, $\forall(m, n) \in A$, W, C, and Δ, we intend to determine a virtual topology $G' = (V, L)$ where the nodes correspond to the nodes in the physical topology $(V = N)$ and the arcs correspond to unidirectional lightpaths. The variables to the problem are:

1. V_{ij} - The number of lightpaths between nodes i and j, $|l_{ij}|$.

2. $P_{mn}^{ij,t}$ - It is 1 if l_{ij}^t is routed through arc (m, n), and 0 otherwise. As the number of necessary lightpaths is only obtained after the optimization process, in this formulation, an upper bound to the number of lightpaths between two nodes i and j ($|l_{ij}|$) is estimated. In practice, this number is limited by W times the number of possible disjoint optical paths between i and j. For higher values of W, this translates into a large unnecessary limit and waste of memory for the creation of variables and constraints. So the boundary $LP = |\Lambda|$ has been defined, which means $t \in \{1, \cdots, LP\}$. As a result, in most cases, not all LP lightpaths are necessary, and for some values of t, l_{ij}^t has no real significance, for it has no optical path p_{ij} associated to it.

3. $\lambda_{ij,t}^{sd,y,v}$ - It is 1 if $\Lambda_{y,sd}^v$ uses l_{ij}^t as an intermediate virtual link, and 0 otherwise.

Given the aforementioned input values and variables, a virtual topology has to be determined such that the total number of transponders is minimized. Thus, the objective function of the problem can be easily derived: since two transponders are needed per lightpath, the minimization of the number of transponders that are necessary to grant the fulfillment of a given set of traffic demands can be obtained through the minimization of the number of lightpaths that have to be established in order to carry those requests.

ILP Formulation For Directed Networks

The ILP formulation of the traffic grooming problem was based on the approach presented in [15], considering several adaptations:

- the objective function is the minimization of the total number of lightpaths obtained to satisfy all the connection requests, as opposed to the maximization of total network throughput.
- The costs of the arcs represent the Euclidian distance between the nodes they connect.
- The wavelength continuity constraint is discarded so as to incorporate the possibility of wavelength conversion in the lightpaths (translucent lightpaths).
- The number of transponders is not known and is directly related to the minimization objective.
- A distance impairment of the optical signal is considered.
- The exact routing of each lightpath in the optical layer has to be known, as well as the exact routing of each demand over the lightpaths.

Objective:

$$\min \sum_{i,j} V_{ij}, \tag{3.1}$$

Constraints:

$$0 \leq \sum_{i,j,t} P_{mn}^{ij,t} \leq W, \qquad \forall (m,n) \in A \tag{3.2}$$

$$\sum_{m} P_{mk}^{ij,t} = \sum_{n} P_{kn}^{ij,t}, \qquad \forall i,j,t,k, \quad with\ i \neq j, i \neq k, j \neq k \tag{3.3}$$

$$\sum_{n,t} P_{in}^{ij,t} = V_{ij}, \qquad \forall i,j \tag{3.4}$$

$$\sum_{m,t} P_{mj}^{ij,t} = V_{ij}, \qquad \forall i,j \tag{3.5}$$

$$\sum_{m,t} P_{mi}^{ij,t} = 0, \qquad \forall i, j \tag{3.6}$$

$$\sum_{n,t} P_{jn}^{ij,t} = 0, \qquad \forall i, j \tag{3.7}$$

$$P_{mn}^{ij,t} \leq \sum_{k} P_{ik}^{ij,t}, \qquad \forall i, j, t, (m, n), m \neq i, k \neq i \tag{3.8}$$

$$P_{mn}^{ij,t} \leq \sum_{k} P_{kj}^{ij,t}, \qquad \forall i, j, t, (m, n), n \neq j, k \neq j \tag{3.9}$$

$$\sum_{k} P_{mk}^{ij,t} \leq 1, \qquad \forall m \neq k \tag{3.10}$$

$$\sum_{i,t} \lambda_{id,t}^{sd,y,v} = 1, \qquad \forall \Lambda_{y,sd}^{v} \tag{3.11}$$

$$\sum_{j,t} \lambda_{sj,t}^{sd,y,v} = 1, \qquad \forall \Lambda_{y,sd}^{v} \tag{3.12}$$

$$\sum_{i,t} \lambda_{is,t}^{sd,y,v} = 0, \qquad \forall \Lambda_{y,sd}^{v} \tag{3.13}$$

$$\sum_{j,t} \lambda_{dj,t}^{sd,y,v} = 0, \qquad \forall \Lambda_{y,sd}^{v} \tag{3.14}$$

$$\sum_{i,t} \lambda_{ik,t}^{sd,y,v} = \sum_{j,t} \lambda_{kj,t}^{sd,y,v}, \qquad \forall \Lambda_{y,sd}^{v} \quad with\ k \neq s, k \neq d \tag{3.15}$$

$$\sum_{v,i,t} \lambda_{id,t}^{sd,y,v} = |\Lambda_{y,sd}|, \qquad \forall \Lambda_{y,sd} \tag{3.16}$$

$$\sum_{v,j,t} \lambda_{sj,t}^{sd,y,v} = |\Lambda_{y,sd}|, \qquad \forall \Lambda_{y,sd} \tag{3.17}$$

$$\sum_{s,d,y,v} y \times \lambda_{ij,t}^{sd,y,v} \leq C, \qquad \forall i, j, t \tag{3.18}$$

$$\sum_{(m,n)} c_{mn} P_{mn}^{ij,t} \leq \Delta, \qquad \forall i, j, t \tag{3.19}$$

$$\lambda_{ij,t}^{sd,y,v} \leq \sum_{(m,n)} P_{mn}^{ij,t}, \qquad \forall \Lambda_{y,sd}^{v}, i, j, t \tag{3.20}$$

$$int \quad V_{ij}, \qquad P_{mn}^{ij,t} \in \{0,1\}, \qquad \lambda_{ij,t}^{sd,y,v} \in \{0,1\} \tag{3.21}$$

- Equation (3.1) presents the objective function: the minimization of the total number of lightpaths.
- Equation (3.2) ensures that the number of lightpaths routed through an arc is constrained by the number of wavelengths it supports.
- Equation (3.3) guarantees that, for an intermediate node k of a lightpath l_{ij}^{t}, the number of incoming lightpaths is equal to the number of outgoing lightpaths.
- Equations (3.4) and (3.5) force V_{ij} to be equal to the number of lightpaths that start at node i and terminate at node j.
- Equations (3.6) and (3.7) ensure that a lightpath $l_{ij}^{t} \in l_{ij}$ cannot be routed through an arc which terminates at node i, nor through an arc which originates at node j.
- Equations (3.8) and (3.9) state that a lightpath l_{ij}^{t} has to begin in node i and terminate in node j.
- Equation (3.10) ensures that a lightpath cannot branch (use more than one outgoing arc of a node).
- Equation (3.11) guarantees that a low-speed request $\Lambda_{y,sd}^{v}$ employs one and only one lightpath terminating at the demand's destination node d. Analogously, Equation (3.12) ensures that one and only one lightpath originating at the demand's origin node s is used.
- Equations (3.13) and (3.14) ensure that a given request $\Lambda_{y,sd}^{v}$ does not use any lightpath terminating at the source node s nor originating at the destination node d.

- Equation (3.15) ensures that the connection request $\Lambda^{v}_{y,sd}$ routed through an intermediate lightpath terminating at node k is continued by the employment of a new lightpath originating at node k.
- Equations (3.16) and (3.17) force the fulfillment of all connection requests.
- Equation (3.18) ensures that the aggregate traffic in a lightpath does not exceed the wavelength capacity.
- Equation (3.19) ensures that each lightpath is routed in the optical layer through a path with a cost of at most Δ km.
- Equation (3.20) guarantees that a demand only uses a given lightpath l^{t}_{ij} if this lightpath exists, that is, if it is associated to a physical path.

Some notes must be added to provide a better understanding of the need of certain constraints. Constraints (3.4) and (3.5) force the variable V_{ij} (the number of lightpaths between i and j) to be equal to the number of lightpaths between i and j that start at i and terminate at j. This however does not forbid that there are more instances of $P^{ij,t}_{mn}$, for different i, j, t, that do not start at i or terminate at j, as long as the flow conservation (3.3) is verified, which means in practice there can be lightpaths that are supposedly between i and j but which don't pass through either of those nodes, and consist of an isolated loop (that is, there are $P^{ij,t}_{mn} = 1$ such that all $m \neq i, \forall m$ and $n \neq j, \forall n$). That is why equations (3.8) and (3.9) are added. Equation (3.10) also plays an important role in solving these isolated loop problems. Besides preventing branching from happening, it also prevents the occurrence of possible solutions where there are lightpaths between i and j correctly starting at i and terminating at j, but due to branching allied with loops, the path that originates in i does not meet with the path that terminates at j.

ILP Formulation For Undirected Networks

Herein, the problem is formulated for undirected networks. There are situations in which problems can be analyzed for a directed network and, from those results, the relevant information can be extrapolated for the undirected case, with little or no adaptations to account for bidirectionality. In this case, such a simple way to adapt the model without explicitly adressing bidirectionality and symmetry was not found.

The adaptation from the previous model to this one was made by considering that each undirected link can be represented by two symmetrical directed arcs. For each traffic demand, the connection is bidirectional and symmetrical. In practice, unidirectional lightpaths will be used to carry traffic in each direction of a bidirectional connection request. To account for lightpath bidirectionality, for each lightpath l_{ij}^t, there will be a symmetrical lightpath l_{ji}^t routed through symmetrical physical arcs. Analogously, to account for the bidirectionality of each connection request, for each request $\Lambda_{y,sd}^v$, a symmetrical request of the same order v, $\Lambda_{y,ds}^v$, routed through symmetrical lightpaths, will be considered.

The formulation for undirected networks consists of the model for directed networks with two additional constraints:

Additional Constraints:

$$P_{mn}^{ij,t} = P_{nm}^{ji,t}, \qquad \forall (m,n), \forall i,j,t \tag{3.22}$$

$$\lambda_{ij,t}^{sd,y,v} = \lambda_{ji,t}^{ds,y,v}, \qquad \forall \Lambda_{y,sd}^v, \forall i,j,t \tag{3.23}$$

- Equation (3.22) ensures that if a lightpath l_{ij}^t is routed through an arc (m,n), there is a symmetrical lightpath l_{ji}^t routed through the symmetrical link (n,m).
- Equation (3.23) ensures that if a connection request $\Lambda_{y,sd}^v$ uses lightpath l_{ij}^t as an intermediate virtual link, there will be a symmetrical request $\Lambda_{y,ds}^v$ using the symmetrical lightpath l_{ji}^t.

Lexicographical Method

Throughout the development of the model, it was observed that some solutions provided unnecessarily long lightpaths, presenting possibilities to reduce their lengths while maintaining the optimal number of lightpaths. To obtain better results, a lexicographical method can be used where, in a first step, the number of lightpaths is minimized, and in a second step, the costs of the lightpaths are minimized while maintaining the minimum number of lightpaths. A lexicographical method orders a set of objective functions with respect to their importance in a given problem. Let $F_i(\mathrm{x})$ denote the i^{th} objective function of a problem, $i = \{1, \cdots, n\}$, where n is the number of objective functions under consideration. The problem is solved in n iterations, and for each $F_i(\mathrm{x})$, the optimal values found for $F_j(\mathrm{x})$ with $j \in \{1, \cdots, i-1\}$ have to be met. If the optimization problem is a minimization, the aim is to minimize all the objective functions $F_i(\mathrm{x})$ while ensuring that $F_j(\mathrm{x}) \leq F_j^*$ for $j \in \{1, \cdots, i-1\}$, where F_j^* refers to the solution obtained in the j^{th} iteration of the problem.

The second step of the optimization includes all the constraints in both formulations, for directed and undirected networks, but the objective function now minimizes the overall cost of lightpaths and a new constraint is added to ensure that the number of lightpaths is the one that resulted from the first optimization. Let V refer to the optimal number of lightpaths obtained in the first step of the optimization process:

Objective:

$$\min \sum_{(m,n),i,j,t} c_{mn} P_{mn}^{ij,t} \tag{3.24}$$

Additional Constraint:

$$\sum_{i,j} V_{ij} = V \tag{3.25}$$

- Equation (3.24) presents the second objective function, which minimizes the cost of the selected lightpaths.
- Equation (3.25) forces the number of lightpaths to be equal to the optimal result obtained in the first step of the optimization.

Alternatively, a linear combination of the two objective functions could be used, where the weights of each of them must be correctly chosen in order to allow obtaining the optimal number of transponders.

Loop Removal

The presented formulation, disregarding the lexicographical optimization, allows for the formation of lightpaths which consist of a correct path together with isolated loops, as mentioned before. The flow conservation constraint (3.3) makes it possible. A post-processing script was developed in order to remove such loops from the solutions provided by the optimizer. When the lexicographical method is employed in the optimization, this processing is no longer necessary as the cost of the lightpaths is minimal only in the absence of loops.

3.2.3 Heuristic

Herein, we propose a heuristic for the traffic grooming problem.

Notation

An intermediate step of this heuristic consists of the creation of a logical auxiliary graph that represents the upper layer of this two-layer problem. Because the algorithm involves both layers, some extra notation was included in order to make the separation between both layers clearer. The logical reachability graph is denoted by $G' = (N, L)$ where N is the set of nodes (the same set of nodes of G) and L is the set of logical arcs. A logical arc between i and j in G' has a direct correspondence to a path between i and j in the underlying physical network G. Such arc may be referred to simply as $(i, j)'$, but the notation $la_{ij}^{u,t}$ is used more frequently so as to distinguish logical arcs that connect the same pair of nodes from one another. The value of u discriminates the u^{th} logical arc between i and j, and the value of t relates to a particular existing lightpath. As it will be detailed in the description of the heuristic, a logical arc may result from an existing lightpath or not. If $t \neq 0$, t implies that $la_{ij}^{u,t}$ had origin in lightpath l_{ij}^{t}. Otherwise, it did not result from an existing lightpath. Each logical arc is characterized by a pair $la_{ij}^{u,t} = (p_{la_{ij}^{u,t}}, c_{la_{ij}^{u,t}})$, where $p_{la_{ij}^{u,t}}$ denotes the specific underlying physical path of the arc in G, and $c_{la_{ij}^{u,t}}$ represents the cost of the arc in G'.

A path in G' is denoted similarly to a path in G but with an appended apostrophe $'$. p'_{ij} represents a path between i and j with $p'_{ij} =< i, (i, k)', k, \cdots, m, (m, j)', j >$, with $i, j, k, m \in N$. The path in G that corresponds to a specific p'_{ij} in G' can be obtained through the concatenation of the physical paths underlying each logical arc, that is, $p_{ij} = p_{la_{ik}^{u_1,t_1}} \diamondsuit \cdots \diamondsuit\ p_{la_{mj}^{u_2,t_2}}$. For simplicity, let $p_{ij} \dashleftarrow p'_{ij}$ represent this operation. Note that underlying paths of different logical arcs may share one or more physical arcs and, as such, the physical path that results from this operation may contain loops.

Algorithm Description

The heuristic is described at a higher level in algorithm 1, but further details concerning the implementation of certain steps are provided in algorithms 2 and 3.

The output of the heuristic is the set of lightpaths established to carry the requests in Λ, referred to as L_{LP}. The number of transponders will be twice the number of lightpaths in this list.

Algorithm 1 GRR

Require: $G(N, A)$, Λ, Δ,W, C

Ensure: L_{LP} ▷ Set of lightpaths to carry the demands

$D_{sp} \leftarrow lexicoSort_{y,sp}(\Lambda)$

$D_{lp} \leftarrow lexicoSort_{y,lp}(\Lambda)$

$minLP \leftarrow \infty$ ▷ auxiliar variable to store minimum number of lightpaths

for all $D \in \{D_{sp}, D_{lp}\}$ **do** ▷ for two different traffic selection orders

  $w \leftarrow W$ ▷ w denotes the number of wavelengths per fiber

  while $w > 0$ **do**

    $FFDemands \leftarrow 0$

    $L'_{LP} \leftarrow \emptyset$

    for all $v \in \{1, \cdots, |D|\}$ **do** ▷ for each demand

      $G'(N, L) \leftarrow generateLogicalGraph(G(N, A), \Lambda^v_{y,sd}, \Delta, L'_{LP}, W, C)$

      $p'_{sd} \leftarrow shortestPath(G'(N, L), s, d, y)$

      if $p'_{sd} \neq \emptyset$ **then**

        $\{L'_{LP}, FFDemands\} \leftarrow grooming(G'(N, L), \Lambda^v_{y,sd}, L'_{LP}, p'_{sd}, FFDemands)$

      end if

    end for

    if $FFDemands = |\Lambda|$ and $|L'_{LP}| < minLp$ **then** ▷ store the best solution

      $minLp \leftarrow |L'_{LP}|$

      $L_{LP} \leftarrow L'_{LP}$

    else if $FFDemands \neq |\Lambda|$ **then** ▷ w is too small to support all demands

      break

    end if

    $w \leftarrow w - 1$

  end while

end for

Algorithm 1 is composed of two main sections: an outer combinatory section that conditions an inner core section.

The core section (steps 7-15) is responsible for the whole process of grooming and routing of a given set of demands, providing as an output the set of lightpaths established to carry the demands in that set. This section is explained with detail later on in this text.

The combinatory facet, which includes the remaining steps, results from two main observations during the development of the heuristic: 1) the order in which demands present themselves for routing can have significant influence on the overall routing results; 2) the variation of the number of wavelengths per arc, because of different capacity conditions, creates different opportunities for demand routing. For those reasons, different orders D of the demands in the traffic matrix and different numbers of wavelengths per arc w are used to run the core section. For each order D, the number of wavelengths per fiber w is decremented one by one from the maximum value (and actual physical number of wavelengths per link) W. Let $\{D, w\}$ define a particular combination of a reordered traffic matrix and a particular number of wavelengths per arc. For each $\{D, w\}$, the core algorithm is run. The decreasing of the number of wavelengths per link goes on until the network does not have enough capacity and the heuristic fails to route all the demands. Each time a minimum number of lightpaths provided by the core algorithm is registered, the list of lightpaths and routing information for each demand is stored. The output variable L_{LP} corresponds to the set of minimum number of lightpaths needed for routing all the demands.

Note that D represents two different orders of Λ, D_{sp} and D_{lp}. Both result, in a first step, from the ordering of Λ by decreasing order of aggregated bandwidth. That is, the first demands to be serviced are those between the node pair (s, d) presenting the highest value for the sum of bandwidth requests. For requests of different pairs (s, d) for which the aggregated traffic has the same value, the tiebreaker is the length of the shortest path from s to d, computed with no capacity considerations. D_{sp} gives priority to the demands which yield the shortest paths and D_{lp} gives priority to the demands which yield the longest paths. This two step ordering is a lexicographical ordering, hence the names $lexicoSort_{Y,x}$ where Y represents the bandwidth objective and x the path length objective with $x \in \{sp, lp\}$.

Having explained the outer combinatory section, herein follows the explanation of how the core section operates internally until the set of lightpaths for a given set of demands is obtained. It begins by setting the number of routed demands $FFDemands$ to zero and the set of lightpaths L'_{LP} that carried the demands in D to an empty set. Then, each demand $\Lambda^{v}_{y,sd} \in D$ is orderly taken, and three essential operations are performed:

Generation of an auxiliary graph $G'(N, L)$

The generation of the auxiliary graph G' is detailed in algorithm 2. The aim is to build a reachability graph, which is a quite common procedure in impairment-aware problems. An auxiliary arc $la_{ij}^{u,t}$ between nodes i and j, with $i, j \in N$, will exist if:

1) L_{LP} contains a lightpath between i and j with sufficient spare capacity for the demand.
2) The condition in 1) is not verified and the shortest path between i and j in G[1], p_{ij}, exists such that $c(p_{ij}) \leq \Delta$ and that every $(m, n) \in A(p_{ij})$ has capacity for a new lightpath.

A lightpath is created when a logical arc resulting from case 2) is a part of the chosen end-to-end path for a demand, which is explained later on when looking into algorithm 3. Thus, whenever a logical arc $la_{ij}^{u,t}$ is created, in both cases 1) and 2), it is associated to an underlying physical path $p_{la_{ij}^{u,t}}$ that corresponds to the current shortest path between i and j in G with enough capacity for the demand. However, there is a difference in the costs assigned to both types of logical arcs. In case 1), to force the reutilization of already existing lightpaths through the grooming of several requests, the cost assigned to the logical arc is much inferior to the one assigned in case 2). In the latter case, if such an arc is chosen to be a part of the path p'_{sd}, it will imply the creation of a new lightpath and thus the deployment of two more transponders. This way, using in the final path a pre-existing lightpath or a set of pre-existing lightpaths will contribute, in almost all situations, to a lower overall cost of p'_{sd}. The cost assigned to a logical arc $la_{ij}^{u,t}$ deriving from a lightpath l_{ij}^t is a function of the physical arcs the lightpath/logical arc spans. More specifically, it is set to the number of arcs of the lightpath, $|A(p_{l_{ij}^t})|$, divided by the total number of arcs in the network, $|A|$. In the other case, the cost of the logical arc is set to unity. Note that the cost of a path resulting exclusively from the reutilization of pre-existing lightpaths would only cost more than the creation of a new lightpath if it used more than the number of arcs in the network. This could happen only in very long paths with several cycles.

[1] A detailed description of Dijkstra's algorithm for the shortest paths, used in this work, can be found in Appendix A.

Algorithm 2 generateLogicalGraph

Require: $G(N,A)$, $\Lambda^v_{y,sd}$, Δ , L_{LP},W, C

Ensure: $G'(N,L)$

$G'(N,L) \leftarrow (N,\emptyset)$
for all node pair $(i,j) \in N^2$, $i \neq j$ **do**
    $u \leftarrow 1$
    $LP_{ij} \leftarrow \{l^t_{ij} \in L_{LP} : C - o_{l^t_{ij}} \geq y\}$ ▷ lightpaths between i and j with enough capacity
    if $LP_{ij} \neq \emptyset$ **then**
        for all $l^t_{ij} \in LP_{ij}$ **do** ▷ for each lightpath, create a logical arc
            $p_{la^{u,t}_{ij}} \leftarrow p_{l^t_{ij}}$
            $c_{la^{u,t}_{ij}} \leftarrow \frac{|A(p_{l^t_{ij}})|}{|A|}$
            $L \leftarrow L \cup \{la^{u,t}_{ij}\}$
            $u \leftarrow u+1$
        end for
    else ▷ if no such lightpath exists, search for a lightpath candidate in G
        $A' \leftarrow \{(m,n) \in A : |l(mn)| < W\}$ ▷ consider only arcs with free wavelengths
        $p_{ij} \leftarrow shortestPath(G(N,A'),i,j)$
        if $p_{ij} \neq \emptyset \wedge c(p_{ij}) \leq \Delta$ **then**
            $p_{la^{1,0}_{ij}} \leftarrow p_{ij}$
            $c_{la^{1,0}_{ij}} \leftarrow 1$
            $L \leftarrow L \cup \{la^{1,0}_{ij}\}$
        end if
    end if
end for

Computation of the shortest path between s and d in G', p'_{sd}

After obtaining G', the shortest path from s to d in G', p'_{sd}, is computed using Dijkstra's algorithm for the shortest path. This path is a sequence of logical arcs and, since logical arcs are associated to an underlying physical path, p'_{sd} can be easily expanded to obtain the corresponding physical path p_{sd} in G, as it was already explained. If the network faces high load, even if just locally, there may be lack of capacity to reach the destination, in which case no path p'_{sd} will be found.

Routing and grooming of $\Lambda^t_{y,sd}$ through p'_{sd}

Assuming a path p'_{sd} is found, the next step is to route the demand, taking advantage of traffic grooming for sharing existing lightpaths if possible. $(\Lambda^v_{y,sd})_{LP}$ denotes the ordered sequence of lightpaths selected to carry the demand $\Lambda^v_{y,sd}$. The path p'_{sd} is a sequence of logical arcs which may exist as lightpaths or not. Different logical arcs may share common physical resources and, as such, it is possible that the concatenation of two logical arcs translates into the occurrence of loops at the physical layer. When a lightpath is established, it is advantageous that it is reused. For that reason, when p'_{sd} contains one or more lightpaths, we choose not to interfere with them for loop removal. However, in parts of p'_{sd} that consist of a logical arc or of a concatenation of logical arcs that have no allocated resources, loops at the physical level will be removed. After loop removal, new logical arcs are computed to create new lightpaths. This helps greatly at reducing loops but does not eliminate completely the possibility of occurrence, because of the existence of a lightpath in p'_{sd} with allocated demands. In a more technical explanation, that is what is done in algorithm 3.

Path p'_{sd} is analyzed, one logical arc $la^{u,t}_{ij}$ at a time, from s to d. If a lightpath l^t_{ij} is not found (step 6), the underlying physical path $p_{l^t_{ij}}$ is concatenated to an auxiliary path variable p_{xy} (steps 15-16). This concatenation proceeds for every logical arc until a logical arc for which a lightpath exists is found. Then, two actions must be performed:

1) The lightpath is reused to route the demand and the capacity of network elements must be updated (steps 13-14).
2) Remove loops in p_{xy} and create corresponding lightpaths (steps 8-11).

Algorithm 3 performs these two actions in reverse order, so that the final path of the demand is obtained in the right order of lightpaths. In case 2), if p_{xy} is not empty,

Algorithm 3 groomingAndRouting

Require: $G'(N, L)$, $\Lambda^v_{y,sd}$, L_{LP}, p'_{sd}, $FFDemands$, W, C

Ensure: L_{LP}, $FFDemands$

$p_{sd} \dashleftarrow p'_{sd}$
$(\Lambda^v_{y,sd})_{LP} \leftarrow \emptyset$ $\triangleright$ ordered sequence of lightpaths used by the demand
$newLPs \leftarrow \emptyset$
$p_{xy} \leftarrow \emptyset$ $\triangleright$ variable to hold subpaths of p_{sd}
for all $p_{la^{u,t}_{ij}} \in p'_{sd}$ **do**
    if $t \neq 0$ **then** $\triangleright$ if $t \neq 0$, $la^{u,t}_{ij}$ had origin in lightpath l^t_{ij}
        if $p_{xy} \neq \emptyset$ **then** $\triangleright$ p_{xy} consists of a concatenation of lightpath candidates
            remove possible existing loops in p_{xy}
            $\{L_{LP}, newLPs\} \leftarrow createNewLightpaths(p_{xy}, L_{LP}, y)$
            $(\Lambda^v_{y,sd})_{LP} \leftarrow (\Lambda^v_{y,sd})_{LP} \mathbin{\textcircled{$\diamond$}} newLPs$
            $p_{xy} \leftarrow \emptyset$
        end if
        $(\Lambda^v_{y,sd})_{LP} \leftarrow (\Lambda^v_{y,sd})_{LP} \mathbin{\textcircled{$\diamond$}} l^t_{ij}$
        $o_{l^t_{ij}} \leftarrow o_{l^t_{ij}} + y$
    else if $|l(mn)| < W, \quad \forall (m, n) \in A(p_{la^{u,t}_{ij}})$ **then**
        $p_{xy} \leftarrow p_{xy} \diamondsuit p_{la^{u,t}_{ij}}$
    else $\triangleright$ there is not sufficient capacity due to loops
        $\{L_{LP}\} \leftarrow deallocateDemandResources((\Lambda^v_{y,sd})_{LP}, L_{LP})$
        return
    end if
end for
if $p_{xy} \neq \emptyset$ **then**
    Remove possible existing loops in p_{xy}
    $\{L_{LP}, newLPs\} \leftarrow createNewLightpaths(p_{xy}, L_{LP}, y)$
    $(\Lambda^v_{y,sd})_{LP} \leftarrow (\Lambda^v_{y,sd})_{LP} \mathbin{\textcircled{$\diamond$}} newLPs$
end if
$FFDemands \leftarrow FFDemands + 1$

it will contain a portion of the physical path p_{sd} that corresponds to a portion of p'_{sd} where no lightpaths were found. This portion will be processed for loop removal. After that, new logical arcs for which the concatenation is free of loops are obtained. This process is explained with more detail in algorithm 4: each new logical arc is obtained

Algorithm 4 createNewLightpaths

Require: p_{xy}, L_{LP}, y

Ensure: L_{LP}, $newLPs$

1: $newLPs \leftarrow \emptyset$
2: $P_\Delta \leftarrow \{p_{rq} \in p_{xy} :\ c(p_{rq}) \leq \Delta \wedge c(p_{rq}) + c_{qk} > \Delta\}$ ▷ Let k be the successor of q in p_{xy}
3: **for all** $p_{rq} \in P_\Delta$ **do**
4: $\quad l_{rw}^{|l_{rq}|+1} \leftarrow (p_{rq}, y)$
5: $\quad L_{LP} \leftarrow L_{LP} \ \cup\ \{l_{rq}^{|l_{rq}|+1}\}$
6: $\quad newLPs \leftarrow newLPs \ \Diamond\ l_{rq}^{|l_{rq}|+1}$
7: $\quad$ **for all** $(m, n) \in A(p_{rq})$ **do**
8: $\quad\quad |l(mn)| \leftarrow |l(mn)| + 1$
9: $\quad\quad$ Place Transponders for $l_{rq}^{|l_{rq}|+1}$ in nodes r and q
10: $\quad$ **end for**
11: **end for**

by scanning the new loop free auxiliary path p_{xy} and creating a list of unregenerated segments, that is, a list of subpaths of p_{xy}, p_{rq}, such that their length is maximum without surpassing Δ (starting at node x) (step 2). For each unregenerated segment, a new lightpath will be created, transponders will be placed at its end nodes, and the capacity of network elements will be updated (steps 3-11).

After all unregenerated segments have undergone this process, p_{xy} will be reset and algorithm 3 proceeds until all logical arcs of p'_{sd} are scanned.

This process of creating new lightpaths from p_{xy} has to be repeated after the outmost for loop, in case no lightpaths are encountered in p'_{sd} or if p'_{sd} does not terminate with a lightpath (steps 22 - 26).

Note that the logical path p'_{sd} is calculated with a shortest path algorithm having in mind that the underlying physical arcs have enough capacity for the demand. The problem is that due to the already mentioned possibility of loop occurrence at the physical layer, if a demand uses the same arc twice, then it would need twice the capacity that was searched for. If the case happens where a certain part of the path only has capacity to support a single new lightpath, and it has to be used more than once (by more than one logical arc), then the routing of the demand through p'_{sd} is not feasible. If so, the lightpaths, capacity and transponders allocated for the demand that is being routed have to be deallocated. This process is explained in algorithm 5.

Algorithm 5 deallocateDemandResources

Require: $(\Lambda^v_{y,sd})_{LP}$, L_{LP}

Ensure: L_{LP}

1: **for all** $l^t_{ij} \in (\Lambda^v_{y,sd})_{LP}$ **do**

2: $\quad o_{l^t_{ij}} \leftarrow o_{l^t_{ij}} - y$

3: $\quad$ **if** $o_{l^t_{ij}} = 0$ **then** $\qquad \triangleright$ lightpath purposely created for this demand

4: $\quad\quad L_{LP} \leftarrow L_{LP} \setminus l^t_{ij}$

5: $\quad\quad$ Remove Transponders for l^t_{ij} from nodes i and j

6: $\quad$ **end if**

7: **end for**

8: $(\Lambda^v_{y,sd})_{LP} \leftarrow \emptyset$

3.3 The GRWAR Problem

The purpose of this section is to approach the GRWAR problem. The problem definition is similar to the one given for the GRR problem, with the inclusion of wavelength assignment and continuity constraints. Furthermore, it is assumed that both optical signal regeneration and wavelength conversion capabilities are obtained either through transponders or regenerators. Analogously to transponders, it is considered that a single regenerator provides regeneration and wavelength conversion in both directions of each wavelength in an optical link.

3.3.1 Heuristics for the GRWAR Problem

One of the objectives of this work consists on the integration of the developed heuristic for the GRR problem with a heuristic for the RWAR problem, having in mind the development of an appropriate heuristic to solve the GRWAR multi-layer problem. Incorporating the wavelength continuity constraint in the presented ILP formulation would lead to an even higher complexity. For that reason, two algorithms are compared: a reference GRR-WA heuristic and the integrated RWAR-GR heuristic.

GRR-WA Reference Heuristic

The reference heuristic consists of the previously presented GRR heuristic followed by wavelength assignment. In other words, the output set of lightpaths obtained from the GRR heuristic is the input to a wavelength assignment algorithm which assigns wavelengths to each lightpath. When wavelength continuity cannot be met throughout the entire underlying physical path of a lightpath, the algorithm places regenerators for wavelength conversion. The wavelength assignment algorithm was developed in [11].

RWAR-GR Heuristic

The multi-layer heuristic here proposed begins by solving the RWAR problem followed by the GR problem, relying on different heuristics for each of the two purposes. The RWAR heuristic that precedes grooming and routing can be found in [11]. The GR part of this heuristic resulted from the presented GRR heuristic after a few adaptations. Algorithm 6 presents an overall view of the combined heuristic, where the modified version of the GRR heuristic is referred to as GRR*.

Algorithm 6 RWAR-GR

Require: $G(N, A)$, Λ, Δ,W, C

Ensure: L_{LP}

$w \leftarrow W$ ▷ w denotes the number of wavelengths per fiber

while $w > 0$ **do**

  $L_{TLP} \leftarrow RWAR(G(N, A), \Delta, \Lambda)$

  $sortByIncreasingCost(L_{TLP})$

  $L_{LP} \leftarrow GRR^{*}(G(N, A), \Lambda, \Delta, L_{TLP})$

  $w \leftarrow w - 1$

end while

The RWAR algorithm takes the physical network, the impairment threshold and the traffic matrix, and routes each demand through a sequence of unregenerated transparent lightpaths. In summary, the optical end-to-end path that is assigned to a given demand is segmented in transparent lightpaths, each associated to a given wavelength. The segmentations occur where there is the need to place regenerators for regeneration and/or wavelength conversion. In this implementation of RWAR, information on the capacity in network elements at each moment strongly influences route determinations with the intent to distribute the traffic in order to avoid local blocking.

It should be mentioned that the RWAR algorithm used in this heuristic has been adapted to be used in this work. Contrary to the algorithm presented in [11], this variation does not actually instantiate regenerators, but exclusively provides the transparent lightpaths the regenerators would bound. Furthermore, the order of demand service was altered. In the traffic matrix, each demand pair (s, d) may be associated to more than one traffic requests. After ordering the traffic matrix by decreasing values of bandwidth, demand service is alternated between node pairs, that is, if the first demand to be served is between a node pair (s_i, d_i), then a second demand between that same node pair can only be served after one traffic request has been attended for all (s_j, d_j), $i \neq j$.

Concerning the GRR section, the loop that gradually constrained the capacity in each arc of the network by decrementing the number of wavelengths has been moved from algorithm 1 to the outside of the RWAR and GRR heuristics, as it is shown in algorithm 6, in order to also have influence on the transparent lightpaths provided by RWAR.

For simplicity, let the set of transparent lightpaths be addressed as L_{TLP}. The main difference between the GRR* section of this heuristic and the original GRR heuristic resides in the candidate paths to logical arcs and lightpaths. In the original GRR heuristic a lightpath between two nodes i and j is formed on the shortest path from i to j which arcs still have at least one free wavelength. In this version, a lightpath between i and j is formed on the shortest transparent lightpath between i and j, which will be the first lightpath between i and j found in L_{TLP} given that it is ordered by increasing cost. That is, the set of lightpath candidates is reduced to the set of transparent lightpaths provided by RWAR. For that reason, algorithm 2 suffers a very small change where the steps 13-14, in which the shortest path between i and j is computed to create a logical arc, are replaced by a single one:

$$p_{ij} \leftarrow p_{l^k_{ij,\lambda}} \text{ where } l^k_{ij,\lambda} \text{ is the first lightpath between } i \text{ and } j \text{ encountered in } L_{TLP}$$

From the RWAR point of view, each demand takes a whole wavelength, which means that the set of provided transparent lightpaths still consists of a large set of options for the GRR* part of the heuristic, which will in most cases need much less capacity.

Algorithm 3 is the one with more changes, and for that reason, the new version is described in algorithm 7. In order to strictly use the transparent lightpaths provided by the RWAR algorithm as lightpath candidates, the routines for loop removal of concatenated logical arcs and for creation of new lightpaths from the loop free path segments are not necessary. When a transparent lightpath is on the base of the creation of a new lightpath, it is removed from L_{TLP}. As the RWAR algorithm creates the transparent lightpaths for the different demands having under consideration the current capacities in network elements, insufficient capacity due to loops in the physical level in a determined logical path p'_{sd} is not a possibility, hence the dismissal of the deallocation procedure.

The output of the heuristic is the set of transparent lightpaths established to carry the set of demands by taking advantage of traffic grooming.

Note that the reason behind demand alternating in RWAR is a consequence of the wavelength number decrement, for two main reasons: first, as RWAR uses much more capacity per demand than the GRR heuristic, there may be situations of shorter capacity where RWAR can only serve the top demands of the traffic matrix. If many of these demands are between the same node pairs, it would end up providing a set of transparent lightpaths that are not as diverse and adequate to the whole set of demands. Second, the lightpaths provided for each RWAR request can carry multiple requests when grooming is accounted for.

Algorithm 7 groomingAndRouting

Require: $G'(N, L)$, $\Lambda^v_{y,sd}$, L_{LP}, p'_{sd}, $FFDemands$, W, C, L_{TLP}

Ensure: L_{LP}, $FFDemands$

$p_{sd} \dashleftarrow p'_{sd}$

$(\Lambda^v_{y,sd})_{LP} \leftarrow \emptyset$

for all $p_{la^{u,t}_{ij}} \in p'_{sd}$ **do**

    if $t \neq 0$ **then** ▷ if $t \neq 0$, $la^{u,t}_{ij}$ had origin in lightpath l^t_{ij}

        $(\Lambda^v_{y,sd})_{LP} \leftarrow (\Lambda^v_{y,sd})_{LP} \textcircled{$\Diamond$} l^t_{ij}$

        $o_{l^t_{ij}} = o_{l^t_{ij}} + y$

    else

        $l^{|l_{ij}|+1}_{ij} \leftarrow (p_{la^{1,0}_{ij}}, y)$

        $L_{LP} \leftarrow L_{LP} \cup \{l^{|l_{ij}|+1}_{ij}\}$

        $(\Lambda^v_{y,sd})_{LP} \leftarrow (\Lambda^v_{y,sd})_{LP} \textcircled{$\Diamond$} l^{|l_{ij}|+1}_{ij}$

        $TLP_{ij} \leftarrow \{l_{ij,\lambda^k} \in L_{TLP}, \forall k, \lambda\}$ ▷ unused transparent lightpaths between i,j

        $L_{TLP} \leftarrow L_{TLP} \setminus \{TLP_{ij}(1)\}$ ▷ where $TLP_{ij}(1)$ is the first element of TLP_{ij}.

        for all $(m, n) \in A(p_{l^{|l_{ij}|+1}_{ij}})$ **do**

            $|l(mn)| \leftarrow |l(mn)| + 1$

        end for

        Place Transponders for $l^{|l_{ij}|+1}_{ij}$ in nodes i and j

    end if

end for

$FFDemands \leftarrow FFDemands + 1$

4 Results and Discussion

4.1 Experimental Setup

Modified versions of networks provided in the library SNDlib1.0 [23] were used. In this library, both topology and end-to-end requests (one for each pair of nodes) of several undirected networks can be found. To reduce the demands to two classes of service, the average bandwidth of a set of requests of a network is computed. Then, for each request, if the bandwidth requirement is below the average value, it is set to 10 Gbps; otherwise, it is set to 40 Gbps. To increase the network load, the set of demands was replicated a certain number of times.

With respect to optimization, the Java API of IBM ILOG CPLEX Optimization Studio V12.6.1 was used for the ILP, and the heuristics were developed in Java. The optimizer was bound to run on 4 cores and for at most 24 hours, after which an upper bound is obtained.

The tests were performed on a computer with a Intel(R) Xeon(R) CPU X5660 @ 2.80GHz processor and a 48 GB RAM.

.

4.2 GRR Results

The results obtained for the models addressing undirected networks, with and without lexicographical optimization, and for the GRR heuristic, are presented in this section. Regarding the model for directed networks, there was not enough time to complete all tests, and the obtained results can be found in Appendix B.

4.2.1 Undirected Networks

Table 4.1 shows the results obtained for the GRR problem for both ILP and heuristic approaches. The tests were run for the ILP formulation for undirected networks without the lexicographical method and followed by loop removal. The considered networks were *polska* and *abilene*. Due to the complexity of the problem, the optimizer only provided solutions for small networks within the specified time bound. For that reason, most of the results, in particular those concerning *polska*, were obtained from subnetworks of the original network for different numbers of nodes, links and demands. The original network contains 12 nodes, 18 links and 66 demands. The files regarding the subnetworks used in the following tests can be found in Appendix C.

Let $|N|$, $|A|$, $|\Lambda|$, Δ and W denote, respectively, the number of nodes, links, demands, the impairment threshold and the number of wavelengths per fiber considered for each test. For each network, and for both ILP and heuristic approaches, the table provides: the average length and average free capacity per each of the established lightpaths; the percentual network occupation, given by the ratio between the used bandwidth and the total available bandwidth in the network ($W * C * |A|$); the maximum number of used wavelengths in a link; the number of required transponders, and the elapsed time until the solution is met. In case the optimizer reaches the time limit, the value "time-out" appears on the elapsed time field, and the solution obtained by the ILP may or not be optimal, thus representing an upper bound to the optimal number of transponders.

Input data						**Cplex / Heuristic**					
Network	$\|N\|$	$\|A\|$	$\|\Lambda\|$	Δ	W	**Avg Lp Length (km)**	**Avg Free Lp Capacity (Gbps)**	**Capacity Usage (%)**	**W$_{max}$**	**Transp. + Reg.**	**Δ T(s)**
polska	6	6	15	1000	48	412.18 / 435.23	11.43 / 21.43	3.51 / 3.51	3 / 4	14 / 14	113.81 / 0.56
			30			518.04 / 414.73	10.0 / 21.81	6.60 / 5.21	6 / 4	20 / 22	800.97 / 0.72
			45			525.83 / 430.08	9.37 / 11.87	10.73 / 8.26	8 / 6	32 / 32	time-out / 1.31
			60			619.28 / 422.77	1.11 / 9.47	15.49 / 10.41	12 / 8	36 / 38	time-out / 1.25
	7	8	21			728.77 / 431.21	7.78 / 20.0	7.11 / 3.80	6 / 5	18 / 20	566.76 / 0.76
			42			545.58 / 412.60	12.14 / 16.0	7.66 / 6.09	8 / 5	28 / 30	23657 / 1.18
	8	10	28			644.4 / 459.83	14.16 / 13.85	6.02 / 4.77	7 / 5	24 / 26	45934.19 / 1.33
abilene	12	15	66	3000	48	1689.07 / 1789.62	11.21 / 15.31	8.05 / 7.71	7 / 7	66 / 64	time-out / 5.73

Table 4.1: Results obtained for the GRR problem, comparing cplex and heuristic performances.

Results show that, for small networks, the heuristic approximates the optimal solution, presenting an average relative error of 6%, with a maximum deviation of 11% from a result obtained with the optimizer. Concerning the seven *polska* subnetworks, the results of the heuristics match those obtained in the ILP for two of them – although one of the provided solutions could be sub-optimal – and present two more transponders than the ILP for the remaining five. Note that two transponders are needed per lightpath, which means that any non-optimal solution will differ from the optimal by a value that is a multiple of 2.

Regarding *abilene*, the heuristic was able to reach a solution which required a number of transponders below the sub-optimal provided by the ILP. In terms of running times, the heuristic clearly outperforms the ILP, running in fractions of seconds to units of seconds, while the ILP runs in hundreds of seconds to tenths of hours, even reaching the 24 hour bound in three of the tests.

Other metrics of interest should be examined: as the heuristic forms lightpaths through shortest path routing – and, on the contrary, the optimizer does not assign a preference to longer or shorter lightpaths as long as the solution yields the same number of transponders – the average length of lightpaths is, in most cases, inferior for the heuristic than for the ILP. Regarding the average free capacity of lightpaths, the ILP shows a more efficient use of lightpath bandwidths, for the majority of the tests. The overall network capacity usage is more contained when using the heuristic, as would be to expect due to the use of shorter lightpaths. The number of wavelengths required to carry all the demands is also a metric that could be of interest when addressing the minimization of installation costs. The results show that neither of the approaches is systematically better, although the heuristic asks for less wavelengths in five of the eight tests.

Overall, the heuristic provides a very interesting trade-off between results and running times. Moreover, the lower average length of lightpaths, network capacity usage, and also a potentially lower number of wavelengths, when compared to the ILP, are very attractive features of this approach.

4.2.2 Undirected Networks with Lexicographical Optimization

Input data						**Cplex Lexi**					
Network	$\|N\|$	$\|A\|$	$\|\Lambda\|$	Δ	W	**Avg Lp Cost (km)**	**Avg Free Lp Capacity (Gbps)**	**Capacity Usage (%)**	**W$_{max}$**	**Transp. + Reg.**	**+ Δ T(s)**
polska	6	6	15	1000	48	326.93	15.71	2.67	3	14	51,68
			30			394.85	6.0	5.28	4	20	951,39
			45			-	-	-	-	-	time-out
			60			-	-	-	-	-	time-out
	7	8	21			309.25	5.56	3.12	3	18	583,83
			42			380.67	1.43	5.89	4	28	time-out
	8	10	28			318.92	6.67	3.29	3	24	time-out
abilene	12	15	66	3000	48	-	-	-	-	66	time-out

Table 4.2: Results obtained for the GRR problem using the lexicograhical formulation for undirected networks.

Table 4.2 shows the results obtained for the GRR problem using the lexicographical optimization for undirected networks. The field $+\Delta T$ refers to the time of execution of only the second part of the optimization, as tests were performed by introducing the optimal/suboptimal results obtained in table 4.1 as a constraint for the maximum number of lightpaths.

It can be seen that using the lexicographical model demands for considerably larger execution times when compared to the previous model. For some cases, the time bound was reached without any solution, and thus no conclusions can be drawn regarding these networks.

Regarding the average length of lightpaths, results show that the lexicographical optimization outperforms both the previous model and the heuristic, which would be to expect since the objective of the second optimization is to minimize the total length of lightpaths.

Regarding the average free capacity of lightpaths, the lexicographical model systematically leads to a more efficient use of lightpath bandwidths, with only one exceptional case.

In terms of network capacity usage, this model shows better results than the previous one, generally achieving results similar to those obtained by the heuristic.

Furthermore, results show that the lexicographical model clearly requires less wavelength channels per link.

In summary, adding the lexicographical optimization leads to better performances regarding all metrics at the expense of considerably larger execution times.

4.3 GRWAR Results

Table 4.3 shows the results obtained for the GRWAR problem in three networks: *polska*, *abilene* and *nobel_germany*. A significant difference between both heuristics is that one of them distributes the traffic according to the arcs occupations. In cases of low network capacity usage, they are expected to respond with similar performances. To explore the effect of such difference in performance, tests were carried out subjecting both algorithms to high network loads. The referred tests were carried out with $W \in \{48, 96\}$. For each network, tests start for $W = 48$ and with the original set of demands. The set of demands is then sequentially replicated until one of the heuristics fails to route all the demands. When that happens, the demand set replication continues from where it stopped, repeating the procedure for $W = 96$. Note that the number of transponders could be different if the tests for $W = 96$ were performed for all sets of demands, because of the external loop that varies the number of wavelengths per link to save the best solution. However, the purpose here is to compare both heuristics under the same conditions.

Regarding the objective function, it is observed that both heuristics perform similarly for low network loads. For network capacity usages under 37%, in 72% of the cases the heuristics

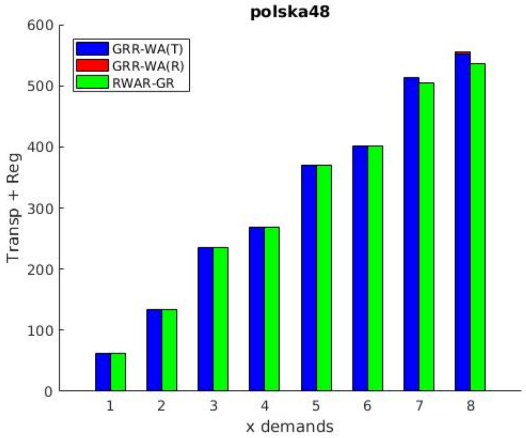
Input data						GRR-WA / RWAR-GR					
Network	$\|N\|$	$\|A\|$	$\|\Lambda\|$	Δ	W	**Avg Lp Length (km)**	**Avg Free Lp Capacity (Gbps)**	**Capacity Usage (%)**	$\mathbf{W}_{max}$	**Transp. + Reg.**	**Δ T(s)**
polska	12	18	66	1000	48	508.13 / 519.92	10.64 / 11.61	7.15 / 7.01	6 / 8	62 + 0 / 62	7.12 / 54.42
			132			498.012 / 508.61	20.0 / 20.0	13.61 / 13.15	16 / 13	134 + 0 / 134	7.38 / 98.65
			198			502.89 / 511.62	22.03 / 22.03	23.66 / 22.91	27 / 22	236 + 0 / 236	12.15 / 117.12
			264			498.01 / 512.11	20.0 / 20.0	27.22 / 26.30	32 / 24	268 + 0 / 268	9.92 / 129.40
			330			501.12 / 514.66	21.30 / 21.30	37.26 / 36.06	43 / 33	370 + 0 / 370	11.71/ 101.68
			396			498.01 / 514.74	20.0 / 20.0	40.83 / 39.44	48 / 35	402 + 0 / 402	8.79 / 93.60
			462			535.57 / 520.88	21.09 / 20.95	54.31 / 49.68	48 / 44	514 + 0 / 504	6.40 / 58.79
			528			493.56 / 523.41	20.0 / 20.0	52.59 / 53.33	48 / 45	552 + 4 / 536	3.28 / 40.84
			594		96	499.82 / 513.75	20.75 / 20.75	32.24 / 31.18	75 / 57	638 + 0 / 638	49.64 / 484.40
			660			498.01 / 514.13	20.0 / 20.0	34.02 / 32.87	80 / 59	670 + 0 / 670	37.85 / 470.39
			726			499.50 / 516.53	20.62 / 20.62	39.05 / 37.84	91 / 67	772 + 0 / 772	45.44 / 428.87
			792			498.01 / 516.49	20.0 / 20.0	40.83 / 39.54	96 / 69	804 + 0 / 804	32.29 / 384.52
			858			516.33 / 518.94	20.53 / 20.53	46.92 / 44.51	96 / 77	906 + 0 / 906	34.24 / 311.41
			924			520.63 / 518.69	20.0 / 20.0	49.12 / 46.20	96 / 79	938 + 0 / 938	21.91 / 284.87
			990			491.38 / 521.58	20.67 / 20.46	51.23 / 51.27	96 / 88	1070 +0 / 1040	17.29 / 218.15
			1056			493.56 / 522.02	20.0 / 20.0	52.59 / 53.05	96 / 90	1104 + 10 / 1072	10.28 / 152.63
			1122			496.01/ 527.91	20.46 / 20.41	58.63 / 58.75	96 / 95	1226 + 30 / 1180	6.95 / 45.97
abilene	12	15	66	3000	48	1789.62 / 1849.87	15.31 / 7.42	7.71 / 8.49	7 / 6	64 + 0 / 62	5.73 / 31.35
			132			1811.72 / 1870.61	10.75 / 11.64	16.97 / 17.05	21 / 18	134 + 0 / 134	8.86 / 43.75
			198			1740.71 / 1768.25	15.56 / 15.74	27.40 / 26.93	34 / 28	230 + 0 / 230	13.23 / 42.59
			264			1769.16 / 1823.86	16.92 / 16.94	36.72 / 36.61	48 / 38	312 + 0 / 314	7.15 / 29.51
			330		96	1736.75 / 1790.20	17.63 / 18.26	23.27 / 23.26	55 / 45	414 + 0 / 414	49.74 / 172.45
			396			1769.16 / 1806.60	16.92 / 16.92	27.54 / 27.22	72 / 55	468 + 0 / 468	44.53 / 165.70
			462			1773.84 / 1810.48	19.53 / 19.27	35.46 / 33.87	96 / 69	608 + 0 / 606	34.71 / 121.72
			528			1782.39 / 1804.76	19.53 / 19.54	39.5 / 38.61	96 / 80	688 + 0 / 698	11.88 / 96.29
			594			1780.22 / 1791.45	20.35 / 20.15	44.77 / 44.30	96 / 84	800 + 4 / 816	6.62 / 60.58
nobel_germany	17	26	121	1000	48	354.19 / 509.17	10.58 / 10.19	6.99 / 9.45	5 / 9	102 + 0 / 104	30.67 / 398.27
			242			411.94 / 463.49	7.11 / 7.91	16.49 / 16.44	19 / 17	180+ 0 / 182	47.54 / 591.34
			363			442.36 / 515.76	22.87 / 21.85	36.18 / 39.26	48 / 42	446 + 0 / 432	13.59 / 342.73
			484			441.35 / 520.10	20.0 / 20.0	43.59 / 44.68	48 / 43	520 + 2 / 484	7.26 / 260.34
			605		96	437.45 / 492.30	21.55 / 21.55	29.89 / 29.26	96 / 66	672 + 0 / 672	81.74 / 2154.56
			726			447.65 / 496.66	20.0 / 20.0	32.88 / 32.11	96 / 71	726 + 0 / 726	53.84 / 2009.30
			847			445.35 / 516.18	21.26 / 21.14	39.87 / 41.06	96 / 83	954 + 0 / 914	43.69 / 1305.14
			968			441.35 / 520.20	20.0 / 20.0	43.59 / 44.13	96 / 86	1040 + 4 / 968	25.57 / 1112.10

Table 4.3: Results obtained for the GRWAR problem, comparing GRR-WA and RWAR-GR heuristic performances.

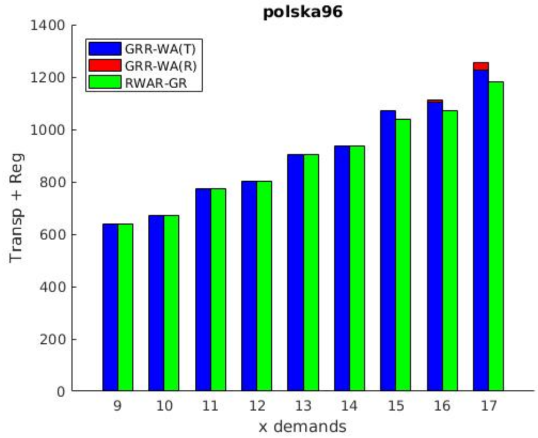
present a matching number of transponders, while in the remaining 28%, there is a variation of two transponders from one to the other. For higher values of network capacity usage, the differences between the results gradually accentuate. Figure 4.1 illustrates the results concerning the number of transponders and regenerators for all the three networks and for both values of W. Concerning the GRR-WA heuristic, R denotes the number of regenerators and T denotes the number of transponders. In the horizontal axis, the factor of replication of the original set of demands is presented. For both *polska* and *nobel_germany*, as the network load increases, the GRR-WA heuristic requires a considerably larger number of transponders and regenerators for wavelength conversion than the RWAR-GR heuristic. As in [4], if it is assumed that a regenerator is twice the cost of a transponder, it is observed that, in the case of *polska*, the RWAR-GR approach obtains results 2-9% better than the other approach. Similarly, improvements from 3-8% are obtained for *nobel_germany*. However, the reverse happens for *abilene*, achieving lower costs with the GRR-WA approach, although with only up to 1.5% of improvement. These values are computed taking the GRR-WA heuristic as a reference. Thus, results indicate that there may be cost advantages when using the proposed RWAR-GR heuristic, although in some cases an improvement is not guaranteed and, as observed for *abilene*, the GRR-WA heuristic may obtain better results. In the two heuristics, the candidate lightpaths are different, not only because they are computed with different algorithms but also because one accounts for wavelength continuity within lightpaths and the other does not. These factors lead to the creation of different virtual topologies with lightpaths of different lengths, where the route selection of one demand can significantly affect the routes of the remaining, and ultimately the number of transponders and the benefits of one or other approach, for a given topology and traffic matrix, may vary.

Running times are clearly more favorable for the reference heuristic. This is caused by the RWAR section of the RWAR-GR heuristic. The algorithm computes alternative paths using K-shortest paths and uses occupation-based arc costs, several times. These tests were performed without imposing a limit to the number of alternative paths, so the limiting factor is the capacity of the network at each moment. The external loop for wavelength variation further increases execution times. A possibility to reduce the running times of the RWAR-GR heuristic would be to limit the number of alternative paths. Note that the execution times decrease as the network load increases, because the external wavelength cycle of both heuristics terminates sooner due to lack of capacity.

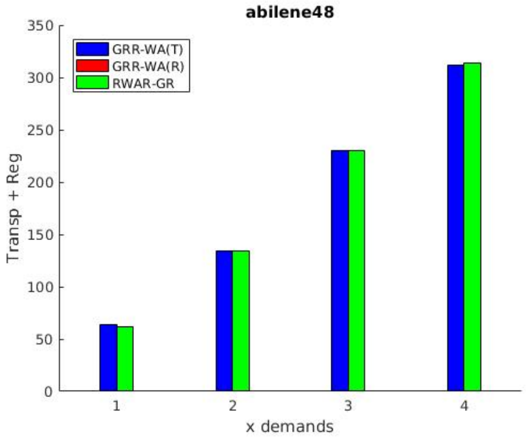
The maximum number of wavelengths in a link is consistently larger for the GRR-WA heuristic, with only two tests of the whole set showing the opposite. Results show that



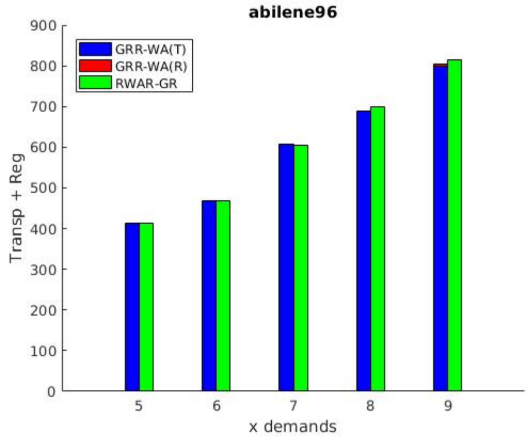
(a) *polska*, $W = 48$



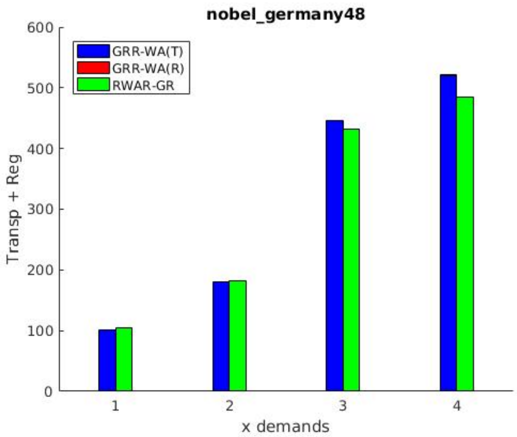
(b) *polska*, $W = 96$



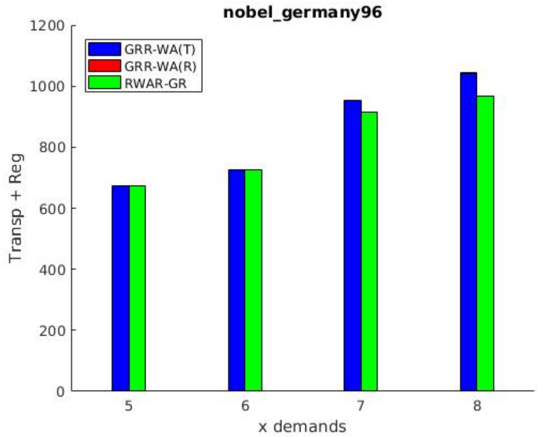
(c) *abilene*, $W = 48$



(d) *abilene*, $W = 96$



(e) *nobel_germany*, $W = 48$



(f) *nobel_germany*, $W = 96$

Figure 4.1: Evolution of the number of transponders and regenerators with the increase of the network load for the tests performed with the RWAR-GR and GRR-WA heuristics.

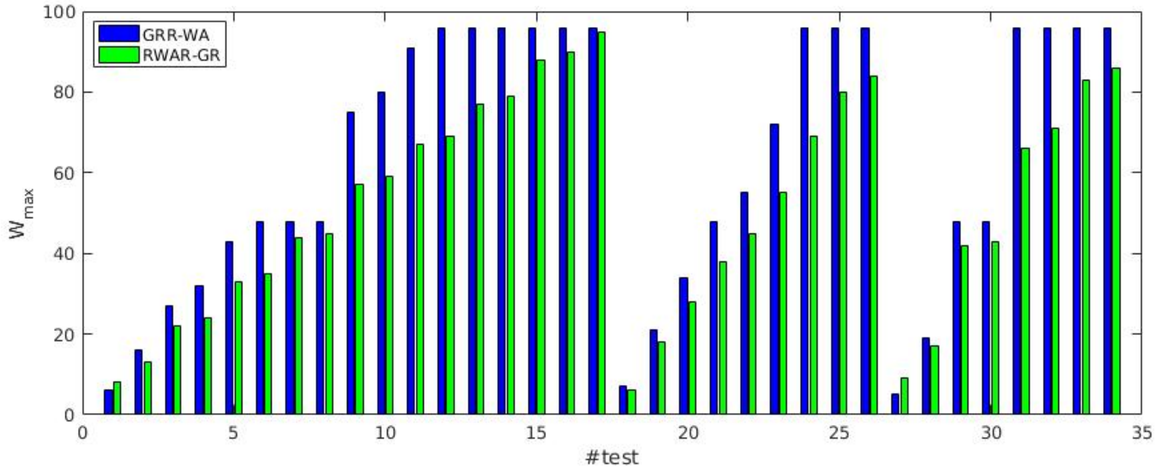


Figure 4.2: Maximum number of wavelengths required for each test presented in table 4.3.

this heuristic is always the first to reach the maximum number of wavelengths per fiber, saturating at that value while the other heuristic slowly rises towards the same value. This behavior can be observed in figure 4.2, and is explained by the already mentioned traffic distribution performed in the RWAR section of the RWAR-GR heuristic. The GRR-WA heuristic uses in average 13% more wavelengths than RWAR-GR.

Regarding average lightpath length, the GRR-WA heuristic uses shorter lightpaths in 94% of the tests. This can also be explained by the use of K-shortest paths in RWAR-GR while GRR-WA always uses shortest path routing. The average difference between the lightpath lengths of both heuristics is of 7%. For average free lightpath capacity, there is not a pattern. The values match for 53% of the tests, and for the remaining, the heuristics with the highest and lowest values alternate almost evenly. In these cases, the difference between them is lower or up to 1Gbps, except for one test that reaches a 7.9 Gbps difference. For the network capacity usage, there is not a pattern either, with the difference in results being of about 10% or lower for all tests except the one regarding *nobel_germany* with $W = 48$.

In summary, results show that the two heuristics perform identically, regarding the cost of transponders and regenerators, until the network load increases to a certain level. From that level up, results indicate that there may be advantages in using the RWAR-GR approach, although that does not happen for all tests. The set of results is not large enough to infer statistics that support a conclusion regarding this observation. In addition, the GRR-WA heuristic is clearly less time-consuming, but requires a higher number of wavelengths per optical link.

5 Conclusions and Future Work

This work addresses the problem of grooming and regeneration in WDM networks, with the objective of minimizing the number of transponders and regenerators. The work was divided in two main stages, one that focused on the GRR problem and a second that addressed the GRWAR problem.

In the first stage, ILP formulations for the GRR problem were developed for both directed and undirected networks, as well as a third variant that considered a lexicographical optimization with the goal of minimizing the cost of lightpaths while requiring the minimum number of transponders. The complexity of the ILP makes it computationally infeasible for large problems and, to address this issue, a GRR heuristic is proposed as an alternative.

Tests were performed for small networks, where the number of transponders and the execution times were compared along with other metrics that could be of interest. Considering undirected networks, both with and without the lexicographical optimization, results show that the heuristic provides an interesting trade-off between results and running times, with an average deviation of 6% and a maximum of 11% from the optimizer results. When the lexicographical optimization is disregarded, it presents a lower average cost of lightpaths, network capacity usage, and a potentially lower number of wavelengths, when compared to the ILP. Adding the lexicographical optimization to the model for undirected networks improves performance regarding all metrics at the expense of considerably larger execution times.

Concerning the model for directed networks, there was not enough time to carry out all the experiments. However, results obtained so far suggest that using this model as an approximation to the one for undirected networks is not an advantageous approach, since not only the solutions were suboptimal, but there were also no time gains.

In the second stage, the GRWAR problem was considered, where two heuristics addressing the problem were developed and compared. One of the heuristics, the reference heuristic, consists of the GRR heuristic followed by a WA algorithm, which assigns wavelengths to lightpaths and places regenerators for wavelength conversion where necessary. The second heuristic, RWAR-GR, consists of an integration of a heuristic that addresses the RWAR problem, developed in the context of another MSc. thesis, with the GRR heuristic. The RWAR heuristics provides a set of transparent paths which are used as lightpath candidates in the GRR heuristic. Both the WA and RWAR algorithms were developed in [11]. Results show that, considering the number of transponders and regenerators, the two heuristics perform similarly until the network load reaches a certain level. For higher network load, the results suggest that there may be advantages in using the RWAR-GR approach, although that is not guaranteed for all the cases. Assuming a regenerator costs twice the price of a transponder, improvements of up to 9% were registered. The GRR-WA heuristic is far less time-consuming, but requires a higher number of wavelengths per optical link. Most of the observed differences are mainly a consequence of the traffic distribution employed in the RWAR heuristic.

From the collaboration between this work and the other previously mentioned MSc. thesis, a paper was written and accepted [24], which can be found in [11].

5.1 Future Work

Regarding the ILP for undirected networks, it was not possible to obtain results for realistically sized networks. The possibility of relaxation of some constraints could be explored in order to reduce its complexity. The use of a restricted set of precomputed physical and logical paths in the ILP can also be explored to reduce the solution space and execution times.

Furthermore, the lexicographical problem should be further explored. For complexity reduction, it could also be approached with a relaxation of the transponder optimality constraint, resulting in a trade-off optimization. Another possibility could rely on the implementation of a linear combination of the two objective functions. These procedures would allow for an evaluation of the performance of the GRR heuristic in larger problems.

Concerning the heuristics for the GRWAR problem, it is important to carry out more experiments in order to obtain solid conclusions about the benefits of using each of them.

One of the directions of this work is its integration in survivability contexts. In fact, the RWAR-GR heuristic has been extended to include a lightpath-based recovery mechanism [11], at the optical level. In the future, it is intended to address survivable impairment-aware traffic grooming using also MPLS recovery techniques.

Appendices

Appendix A

Dijkstra's Algorithm

A.1 Heap Implementations

A heap is a data structure that allows for efficiently storing and manipulating a set of elements, where each element has an associated key. It has its elements structured in a rooted tree where the arcs define a predecessor - successor relationship between the pair of nodes they connect. In a d-heap, each node has a maximum of d successors, with the depth of each node being defined as the number of arcs in the unique path to the root node.

The order of insertion of elements in a heap is done from left to right and in increasing order of depth. A crucial property of the d-heap is the *heap order property*, that states that the key of node i in the heap is never greater than the key of each of its successors. This implies that the root node of the d-heap will hold the minimum key.

This data structure is stored in an array organized in a manner that conveys a rather efficient manipulation. With this storage, the position of the predecessors and successors of each node are well defined in function of the value d.

The basic operations required for the manipulation of a heap data structure comprehend its creation , the insertion and removal of elements, the alteration of the key of elements in the heap and the retrieval and deletion of the element possessing the smallest key (that is, the element on the root node).

The insertion, deletion, and the alteration of keys of preexisting elements in the heap will, most certainly, lead to a situation where the *heap order property* is violated. In these situations, sorting operations have to be performed in order to restore the heap order.

As for the operations involving the element holding the minimum key, they will always address the element in the root node, although the deletion of that element will require the procedure of heap order restoration.

Heap data structures are widely used in graph algorithms. One of such applications is the Dijkstra's algorithm, where the elements of the heap represent the nodes of the network.

A.2 Dijkstra's Algorithm

Dijkstra's algorithm offers a solution for a particular application of the minimum cost problem: the shortest path problem. In this class of problems, the goal is to find the minimum cost path from a certain source node to a certain destination node, given a set of individual costs c_{mn} associated to each arc (m, n) in the network.

Dijkstra's algorithm addresses this problem, making it possible to find the shortest paths from a source node to the remaining nodes in the network.

A fundamental aspect of the algorithm lies in the division of the set of nodes in the network into two subsets: permanently labeled and temporarily labeled nodes. At each moment, there is a distance label $d(m)$ associated to each node m in the network. This label can be permanent or temporary: the permanently labeled nodes are the ones for which the shortest path from the source has already been determined, while the temporarily labeled nodes are the remaining, for which the current distance label represents an upper bound to the true shortest distance. In each iteration of the algorithm, the temporarily labeled node m holding the minimum value of the distance label is marked as permanent - *node selection operation* -, which means the shortest path to that node has been found and its length is $d(m)$. The algorithm proceeds with the cost evaluation of each arc leaving the selected node, $(m, n) \in A^+(m)$. The distance labels of the n nodes will be updated if the cost of the shortest path from the source to the selected node ($c(p_{sm})$) together with the cost of the arc (m, n) under evaluation (c_{mn}), is lower than the current distance label of n node, $d(n)$ - *distance update operation*.

During initialization, the source node s is assigned a distance label of $d(s) = 0$, and the remaining nodes are given infinite distance labels. Thus, the first selected node will be the source node, once $d(s) < d(n)$, $\forall n$. Then, all the nodes directly connected to s by the arcs in $A^+(s)$ will have their distance labels evaluated and updated to the values of the costs of the corresponding arcs $d(n) = d(s) + c_{sn} = c_{sn}$. The node presenting the shortest distance to the source will be selected in the next operation, and so on. Once a node is marked as permanent, arcs containing such node will not be scanned while evaluating the adjacency list of any node. For each selected node, the predecessor node (previously selected node) in the shortest path has to be saved so as to store the necessary information for determination of the node sequence of each of the shortest paths. The algorithm will terminate when all the nodes are marked as permanent.

Regarding computational complexity, Dijkstra's classic algorithm presents a $O(|N^2|)$ computational time bound - with total node selections taking $O(|N^2|)$ time and total distance label updates consuming $O(|A|)$ time [25]. This means that Dijkstra's algorithm runs in $O(|N|^2)$ time, which is the best result for completely dense networks but can be improved when that is not the case. In this work, a heap structure was used to store the data on which Dijkstra's algorithm relies to determine the shortest paths. Given the purpose of the shortest path algorithm, the key of each node will be the corresponding distance label. The node to be labeled as permanent in each iteration will be removed from the root node of the heap. A binary ($d = 2$) heap has been implemented. With the binary heap implementation, Dijkstra's algorithm runs in $O(|A| \log |N|)$ time [25].

Algorithm 8 describes the implemented solution at a high level. Here, $bHeap$ refers to the implemented binary heap and *solution* represents the data structure consuming the output results from the algorithm, storing the shortest path value for each node on the network as well as it's predecessor in that same path. Let the nodes preceding and succeeding a node m in a path be denoted by $pred(m)$ and $suc(m)$, respectively.Three allusions to heap implemented operations are made:

- *insert(node)* - deals with the insertion of a node in the heap. This operation must assure the heap order property, and as such must perform a heap order restoration operation each time a node is inserted.

- *decreaseKeyValue(node, decreasedKeyValue)* - responsible for decreasing the distance label (and the key attribute) of a heap element. This too calls for a heap order restoration operation;

- *deleteMin()* - removes and retrieves the element holding the minimum distance label from the heap, that is, the root node of the heap.

Algorithm 8 Dijkstra

```
d(j) ← ∞ for all j ∈ N;
d(s) ← 0, pred(s) ← 0;
bHeap = createHeap();
solution = createArrayList();
bHeap.insert(s);

while bHeap.size() > 0 do
    minNode = bHeap.deleteMin();
    solution.insert(minNode);
    for each j ∈ A(minNode) do
        if j ∉ solution then
            candidateCost ← d(minNode) + c_minNodej;
            if d(j) = ∞ then
                d(j) ← candidateCost, pred(j) = minNode;
                bHeap.insert(j);
            else if d(j) > candidateCost then
                d(j) ← candidateCost, pred(j) = minNode;
                bHeap.decreaseKeyValue(j, candidateCost);
            end if
        end if
    end for
end while
```

Appendix B

Additional Results

B.1 ILP Formulation for Directed Networks

Table B.1 presents the results obtained for the ILP formulation for directed networks, without lexicographical optimization. In the undirected model, traffic is bidirectional and symmetrical, and lightpaths are minimized having that consideration. An approximation to the solution can be obtained with the model for directed networks, duplicating the obtained number of lightpaths. The main intent was to analyze whether the elimination of the bidirectional constraints would lead to faster executions while providing a solution close to the optimal. Results suggest that there is no gain in execution times, and all the obtained results are suboptimal considering the number of lightpaths, which means that there was no advantage in using this model as an approximation.

Input data						**Cplex Directed**				
Network	$\vert N\vert$	$\vert A\vert$	$\vert\Lambda\vert$	Δ	W	**Avg Lp Cost (km)**	**Avg Free Lp Capacity (Gbps)**	**Capacity Usage (%)**	**Transp. + Reg.**	**Δ T(s)**
polska	6	6	15	1000	48	629.86	18.75	2.97	8	149.4
			30			565.66	14.54	3.75	11	1681.85
	7	8	21			458.90	15.0	2.47	10	1081.72
			42			515.35	13.33	3.48	15	25355.32

Table B.1: Results obtained for the GRR problem using the formulation for directed networks.

Appendix C

Subnetworks

C.1 polska_6_6_15

```
# network polska

# NODE SECTION
#
# <node_id> [(<longitude>, <latitude>)]


NODES (
  Gdansk ( 18.60 54.20 )
  Bydgoszcz ( 17.90 53.10 )
  Lodz ( 19.40 51.70 )
  Poznan ( 16.80 52.40 )
  Warsaw ( 21.00 52.20 )
  Wroclaw ( 16.90 51.10 )
)

# LINK SECTION
#
# <link_id> ( <source> <target> ) <pre_installed_capacity> <pre_installed_capacity_cost> <routing_cost> <setup_cost>
( {<module_capacity> <module_cost>}* )

LINKS (
  Link_0_10 ( Gdansk Warsaw ) 0.00 0.00 0.00 156.00 ( 155.00 156.00 622.00 468.00 )
  Link_1_7 ( Bydgoszcz Poznan ) 0.00 0.00 0.00 186.00 ( 155.00 186.00 622.00 558.00 )
  Link_1_10 ( Bydgoszcz Warsaw ) 0.00 0.00 0.00 272.00 ( 155.00 272.00 622.00 816.00 )
  Link_6_10 ( Lodz Warsaw ) 0.00 0.00 0.00 165.00 ( 155.00 165.00 622.00 495.00 )
  Link_6_11 ( Lodz Wroclaw ) 0.00 0.00 0.00 305.00 ( 155.00 305.00 622.00 915.00 )
  Link_7_11 ( Poznan Wroclaw ) 0.00 0.00 0.00 195.00 ( 155.00 195.00 622.00 585.00 )
)

# DEMAND SECTION
#
# <demand_id> ( <source> <target> ) <routing_unit> <demand_value> <max_path_length>

DEMANDS (
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
```

```
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
)
```

C.2 polska_6_6_30

```
# NODE SECTION
#
# <node_id> [(<longitude>, <latitude>)]

NODES (
  Gdansk ( 18.60 54.20 )
  Bydgoszcz ( 17.90 53.10 )
  Lodz ( 19.40 51.70 )
  Poznan ( 16.80 52.40 )
  Warsaw ( 21.00 52.20 )
  Wroclaw ( 16.90 51.10 )
)

# LINK SECTION
#
# <link_id> ( <source> <target> ) <pre_installed_capacity> <pre_installed_capacity_cost> <routing_cost> <setup_cost>
( {<module_capacity> <module_cost>}* )

LINKS (
  Link_0_10 ( Gdansk Warsaw ) 0.00 0.00 0.00 156.00 ( 155.00 156.00 622.00 468.00 )
  Link_1_7 ( Bydgoszcz Poznan ) 0.00 0.00 0.00 186.00 ( 155.00 186.00 622.00 558.00 )
  Link_1_10 ( Bydgoszcz Warsaw ) 0.00 0.00 0.00 272.00 ( 155.00 272.00 622.00 816.00 )
  Link_6_10 ( Lodz Warsaw ) 0.00 0.00 0.00 165.00 ( 155.00 165.00 622.00 495.00 )
  Link_6_11 ( Lodz Wroclaw ) 0.00 0.00 0.00 305.00 ( 155.00 305.00 622.00 915.00 )
  Link_7_11 ( Poznan Wroclaw ) 0.00 0.00 0.00 195.00 ( 155.00 195.00 622.00 585.00 )
)

# DEMAND SECTION
#
# <demand_id> ( <source> <target> ) <routing_unit> <demand_value> <max_path_length>

DEMANDS (
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
```

```
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
)
```

C.3 polska_6_6_45

```
# NODE SECTION
#
# <node_id> [(<longitude>, <latitude>)]

NODES (
  Gdansk ( 18.60 54.20 )
  Bydgoszcz ( 17.90 53.10 )
  Lodz ( 19.40 51.70 )
  Poznan ( 16.80 52.40 )
  Warsaw ( 21.00 52.20 )
  Wroclaw ( 16.90 51.10 )
)

# LINK SECTION
#
# <link_id> ( <source> <target> ) <pre_installed_capacity> <pre_installed_capacity_cost> <routing_cost> <setup_cost>
( {<module_capacity> <module_cost>}* )

LINKS (
  Link_0_10 ( Gdansk Warsaw ) 0.00 0.00 0.00 156.00 ( 155.00 156.00 622.00 468.00 )
  Link_1_7 ( Bydgoszcz Poznan ) 0.00 0.00 0.00 186.00 ( 155.00 186.00 622.00 558.00 )
  Link_1_10 ( Bydgoszcz Warsaw ) 0.00 0.00 0.00 272.00 ( 155.00 272.00 622.00 816.00 )
  Link_6_10 ( Lodz Warsaw ) 0.00 0.00 0.00 165.00 ( 155.00 165.00 622.00 495.00 )
  Link_6_11 ( Lodz Wroclaw ) 0.00 0.00 0.00 305.00 ( 155.00 305.00 622.00 915.00 )
  Link_7_11 ( Poznan Wroclaw ) 0.00 0.00 0.00 195.00 ( 155.00 195.00 622.00 585.00 )
)

# DEMAND SECTION
#
# <demand_id> ( <source> <target> ) <routing_unit> <demand_value> <max_path_length>

DEMANDS (
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
```

```
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
)
```

C.4 polska_6_6_60

```
# NODE SECTION
#
# <node_id> [(<longitude>, <latitude>)]



NODES (
  Gdansk ( 18.60 54.20 )
  Bydgoszcz ( 17.90 53.10 )
  Lodz ( 19.40 51.70 )
  Poznan ( 16.80 52.40 )
  Warsaw ( 21.00 52.20 )
  Wroclaw ( 16.90 51.10 )
)

# LINK SECTION
#
# <link_id> ( <source> <target> ) <pre_installed_capacity> <pre_installed_capacity_cost> <routing_cost> <setup_cost> ( {<m

LINKS (
  Link_0_10 ( Gdansk Warsaw ) 0.00 0.00 0.00 156.00 ( 155.00 156.00 622.00 468.00 )
  Link_1_7 ( Bydgoszcz Poznan ) 0.00 0.00 0.00 186.00 ( 155.00 186.00 622.00 558.00 )
  Link_1_10 ( Bydgoszcz Warsaw ) 0.00 0.00 0.00 272.00 ( 155.00 272.00 622.00 816.00 )
  Link_6_10 ( Lodz Warsaw ) 0.00 0.00 0.00 165.00 ( 155.00 165.00 622.00 495.00 )
  Link_6_11 ( Lodz Wroclaw ) 0.00 0.00 0.00 305.00 ( 155.00 305.00 622.00 915.00 )
  Link_7_11 ( Poznan Wroclaw ) 0.00 0.00 0.00 195.00 ( 155.00 195.00 622.00 585.00 )
)

# DEMAND SECTION
#
# <demand_id> ( <source> <target> ) <routing_unit> <demand_value> <max_path_length>

DEMANDS (
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
```

```
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED

)
```

C.5 polska_7_8_21

```
# NODE SECTION
#
# <node_id> [(<longitude>, <latitude>)]
```

```
NODES (
  Gdansk ( 18.60 54.20 )
  Bydgoszcz ( 17.90 53.10 )
  Katowice ( 18.80 50.30 )
  Lodz ( 19.40 51.70 )
  Poznan ( 16.80 52.40 )
  Warsaw ( 21.00 52.20 )
  Wroclaw ( 16.90 51.10 )
)

# LINK SECTION
#
# <link_id> ( <source> <target> ) <pre_installed_capacity> <pre_installed_capacity_cost> <routing_cost> <setup_cost>
( {<module_capacity> <module_cost>}* )

LINKS (
  Link_0_10 ( Gdansk Warsaw ) 0.00 0.00 0.00 156.00 ( 155.00 156.00 622.00 468.00 )
  Link_1_7 ( Bydgoszcz Poznan ) 0.00 0.00 0.00 186.00 ( 155.00 186.00 622.00 558.00 )
  Link_1_10 ( Bydgoszcz Warsaw ) 0.00 0.00 0.00 272.00 ( 155.00 272.00 622.00 816.00 )
  Link_3_6 ( Katowice Lodz ) 0.00 0.00 0.00 181.00 ( 155.00 181.00 622.00 543.00 )
  Link_3_11 ( Katowice Wroclaw ) 0.00 0.00 0.00 208.00 ( 155.00 208.00 622.00 624.00 )
  Link_6_10 ( Lodz Warsaw ) 0.00 0.00 0.00 165.00 ( 155.00 165.00 622.00 495.00 )
  Link_6_11 ( Lodz Wroclaw ) 0.00 0.00 0.00 305.00 ( 155.00 305.00 622.00 915.00 )
  Link_7_11 ( Poznan Wroclaw ) 0.00 0.00 0.00 195.00 ( 155.00 195.00 622.00 585.00 )
)

# DEMAND SECTION
#
# <demand_id> ( <source> <target> ) <routing_unit> <demand_value> <max_path_length>

DEMANDS (
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_3 ( Gdansk Katowice ) 1 174.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_3 ( Bydgoszcz Katowice ) 1 117.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_3_6 ( Katowice Lodz ) 1 110.00 UNLIMITED
  Demand_3_7 ( Katowice Poznan ) 1 132.00 UNLIMITED
  Demand_3_10 ( Katowice Warsaw ) 1 126.00 UNLIMITED
  Demand_3_11 ( Katowice Wroclaw ) 1 100.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
)
```

C.6 polska_7_8_42

```
# NODE SECTION
#
# <node_id> [(<longitude>, <latitude>)]

NODES (
  Gdansk ( 18.60 54.20 )
  Bydgoszcz ( 17.90 53.10 )
  Katowice ( 18.80 50.30 )
```

```
  Lodz ( 19.40 51.70 )
  Poznan ( 16.80 52.40 )
  Warsaw ( 21.00 52.20 )
  Wroclaw ( 16.90 51.10 )
)

# LINK SECTION
#
# <link_id> ( <source> <target> ) <pre_installed_capacity> <pre_installed_capacity_cost> <routing_cost> <setup_cost>
( {<module_capacity> <module_cost>}* )

LINKS (
  Link_0_10 ( Gdansk Warsaw ) 0.00 0.00 0.00 156.00 ( 155.00 156.00 622.00 468.00 )
  Link_1_7 ( Bydgoszcz Poznan ) 0.00 0.00 0.00 186.00 ( 155.00 186.00 622.00 558.00 )
  Link_1_10 ( Bydgoszcz Warsaw ) 0.00 0.00 0.00 272.00 ( 155.00 272.00 622.00 816.00 )
  Link_3_6 ( Katowice Lodz ) 0.00 0.00 0.00 181.00 ( 155.00 181.00 622.00 543.00 )
  Link_3_11 ( Katowice Wroclaw ) 0.00 0.00 0.00 208.00 ( 155.00 208.00 622.00 624.00 )
  Link_6_10 ( Lodz Warsaw ) 0.00 0.00 0.00 165.00 ( 155.00 165.00 622.00 495.00 )
  Link_6_11 ( Lodz Wroclaw ) 0.00 0.00 0.00 305.00 ( 155.00 305.00 622.00 915.00 )
  Link_7_11 ( Poznan Wroclaw ) 0.00 0.00 0.00 195.00 ( 155.00 195.00 622.00 585.00 )
)

# DEMAND SECTION
#
# <demand_id> ( <source> <target> ) <routing_unit> <demand_value> <max_path_length>

DEMANDS (
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_3 ( Gdansk Katowice ) 1 174.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_3 ( Bydgoszcz Katowice ) 1 117.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_3_6 ( Katowice Lodz ) 1 110.00 UNLIMITED
  Demand_3_7 ( Katowice Poznan ) 1 132.00 UNLIMITED
  Demand_3_10 ( Katowice Warsaw ) 1 126.00 UNLIMITED
  Demand_3_11 ( Katowice Wroclaw ) 1 100.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_3 ( Gdansk Katowice ) 1 174.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_3 ( Bydgoszcz Katowice ) 1 117.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_3_6 ( Katowice Lodz ) 1 110.00 UNLIMITED
  Demand_3_7 ( Katowice Poznan ) 1 132.00 UNLIMITED
  Demand_3_10 ( Katowice Warsaw ) 1 126.00 UNLIMITED
  Demand_3_11 ( Katowice Wroclaw ) 1 100.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
```

```
)
```

C.7 polska_8_10_28

```
# NODE SECTION
#
# <node_id> [(<longitude>, <latitude>)]

NODES (
  Gdansk ( 18.60 54.20 )
  Bydgoszcz ( 17.90 53.10 )
  Kolobrzeg ( 16.10 54.20 )
  Lodz ( 19.40 51.70 )
  Poznan ( 16.80 52.40 )
  Szczecin ( 14.50 53.40 )
  Warsaw ( 21.00 52.20 )
  Wroclaw ( 16.90 51.10 )
)

# LINK SECTION
#
# <link_id> ( <source> <target> ) <pre_installed_capacity> <pre_installed_capacity_cost> <routing_cost> <setup_cost>
 ( {<module_capacity> <module_cost>}* )

LINKS (
  Link_0_10 ( Gdansk Warsaw ) 0.00 0.00 0.00 156.00 ( 155.00 156.00 622.00 468.00 )
  Link_0_2 ( Gdansk Kolobrzeg ) 0.00 0.00 0.00 272.00 ( 155.00 272.00 622.00 816.00 )
  Link_1_2 ( Bydgoszcz Kolobrzeg ) 0.00 0.00 0.00 156.00 ( 155.00 156.00 622.00 468.00 )
  Link_1_7 ( Bydgoszcz Poznan ) 0.00 0.00 0.00 186.00 ( 155.00 186.00 622.00 558.00 )
  Link_1_10 ( Bydgoszcz Warsaw ) 0.00 0.00 0.00 272.00 ( 155.00 272.00 622.00 816.00 )
  Link_2_9 ( Kolobrzeg Szczecin ) 0.00 0.00 0.00 237.00 ( 155.00 237.00 622.00 711.00 )
  Link_6_10 ( Lodz Warsaw ) 0.00 0.00 0.00 165.00 ( 155.00 165.00 622.00 495.00 )
  Link_6_11 ( Lodz Wroclaw ) 0.00 0.00 0.00 305.00 ( 155.00 305.00 622.00 915.00 )
  Link_7_9 ( Poznan Szczecin ) 0.00 0.00 0.00 142.00 ( 155.00 142.00 622.00 426.00 )
  Link_7_11 ( Poznan Wroclaw ) 0.00 0.00 0.00 195.00 ( 155.00 195.00 622.00 585.00 )
)

# DEMAND SECTION
#
# <demand_id> ( <source> <target> ) <routing_unit> <demand_value> <max_path_length>

DEMANDS (
  Demand_0_1 ( Gdansk Bydgoszcz ) 1 195.00 UNLIMITED
  Demand_0_2 ( Gdansk Kolobrzeg ) 1 158.00 UNLIMITED
  Demand_0_6 ( Gdansk Lodz ) 1 158.00 UNLIMITED
  Demand_0_7 ( Gdansk Poznan ) 1 182.00 UNLIMITED
  Demand_0_9 ( Gdansk Szczecin ) 1 175.00 UNLIMITED
  Demand_0_10 ( Gdansk Warsaw ) 1 122.00 UNLIMITED
  Demand_0_11 ( Gdansk Wroclaw ) 1 114.00 UNLIMITED
  Demand_1_2 ( Bydgoszcz Kolobrzeg ) 1 179.00 UNLIMITED
  Demand_1_6 ( Bydgoszcz Lodz ) 1 198.00 UNLIMITED
  Demand_1_7 ( Bydgoszcz Poznan ) 1 189.00 UNLIMITED
  Demand_1_9 ( Bydgoszcz Szczecin ) 1 142.00 UNLIMITED
  Demand_1_10 ( Bydgoszcz Warsaw ) 1 137.00 UNLIMITED
  Demand_1_11 ( Bydgoszcz Wroclaw ) 1 163.00 UNLIMITED
  Demand_2_6 ( Kolobrzeg Lodz ) 1 128.00 UNLIMITED
  Demand_2_7 ( Kolobrzeg Poznan ) 1 195.00 UNLIMITED
  Demand_2_9 ( Kolobrzeg Szczecin ) 1 105.00 UNLIMITED
  Demand_2_10 ( Kolobrzeg Warsaw ) 1 173.00 UNLIMITED
  Demand_2_11 ( Kolobrzeg Wroclaw ) 1 157.00 UNLIMITED
  Demand_6_7 ( Lodz Poznan ) 1 169.00 UNLIMITED
  Demand_6_9 ( Lodz Szczecin ) 1 196.00 UNLIMITED
  Demand_6_10 ( Lodz Warsaw ) 1 193.00 UNLIMITED
```

```
  Demand_6_11 ( Lodz Wroclaw ) 1 151.00 UNLIMITED
  Demand_7_9 ( Poznan Szczecin ) 1 125.00 UNLIMITED
  Demand_7_10 ( Poznan Warsaw ) 1 194.00 UNLIMITED
  Demand_7_11 ( Poznan Wroclaw ) 1 194.00 UNLIMITED
  Demand_9_10 ( Szczecin Warsaw ) 1 181.00 UNLIMITED
  Demand_9_11 ( Szczecin Wroclaw ) 1 195.00 UNLIMITED
  Demand_10_11 ( Warsaw Wroclaw ) 1 141.00 UNLIMITED
)
```

Bibliography

[1] H. Wang and G. N. Rouskas, "Traffic grooming in optical networks: Decomposition and partial linear programming (LP) relaxation," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, pp. 825–535, Aug 2013.

[2] C. Pluntke, M. Menth, and M. Duelli, "Capex-aware design of survivable DWDM mesh networks," in *Proceedings of the 2009 IEEE International Conference on Communications*, ICC'09, (Piscataway, NJ, USA), pp. 2348–2353, IEEE Press, 2009.

[3] J.-Q. Hu and E. Modiano, *Traffic Grooming in WDM Networks*, pp. 245–264. Boston, MA: Springer US, 2005.

[4] A. N. Patel, C. Gao, J. P. Jue, X. Wang, Q. Zhang, P. Palacharla, and T. Naito, "Cost efficient traffic grooming and regenerator placement in impairment-aware optical WDM networks," *Optical Switching and Networking*, vol. 9, no. 3, pp. 225 – 239, 2012. ONDM 2010.

[5] A. Beshir, R. Nuijts, R. Malhotra, and F. Kuipers, "Survivable impairment-aware traffic grooming," in *Networks and Optical Communications (NOC), 2011 16th European Conference on*, pp. 208–211, July 2011.

[6] J. Strand, A. L. Chiu, and R. Tkach, "Issues for routing in the optical layer," *IEEE Communications Magazine*, vol. 39, pp. 81–87, Feb 2001.

[7] G. Shen and R. S. Tucker, "Translucent optical networks: the way forward [topics in optical communications]," *IEEE Communications Magazine*, vol. 45, pp. 48–54, Feb 2007.

[8] J. Q. Hu and B. Leida, "Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, p. 501, March 2004.

[9] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," *IEEE/ACM Trans. Netw.*, vol. 11, pp. 285–299, Apr. 2003.

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

[11] R. D. S. C. Mendes, "Impairment-Aware Minimization Of The Number Of Regenerators in Optical Transport Networks," Master's thesis, University of Coimbra, Portugal, 2016. Submitted.

[12] J. Berthold, A. A. M. Saleh, L. Blair, and J. M. Simmons, "Optical networking: Past, present, and future," *Journal of Lightwave Technology*, vol. 26, pp. 1104–1118, May 2008.

[13] R. Dutta and G. N. Rouskas, "Traffic grooming in wdm networks: past and future," *IEEE Network*, vol. 16, pp. 46–56, Nov 2002.

[14] G. Shen and R. S. Tucker, "Sparse traffic grooming in translucent optical networks," *Journal of Lightwave Technology*, vol. 27, pp. 4471–4479, Oct 2009.

[15] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 122–133, Jan 2002.

[16] M. Scheffel, R. G. Prinz, C. G. Gruber, A. Autenrieth, and D. A. Schupke, "Optimal routing and grooming for multilayer networks with transponders and muxponders," in *GLOBECOM*, 2006.

[17] R. Dutta, S. Huang, and G. N. Rouskas, "On optimal traffic grooming in elemental network topologies," 2003.

[18] L. Cox, J. Sanchez, and L. Lu, "Cost savings from optimized packing and grooming of optical circuits: Mesh versus ring comparisons," *Optical Networks Magazine*, vol. 2, no. 3, pp. 72–90, 2001.

[19] W. Grover, J. Doucette, M. Clouqueur, D. Leung, and D. Stamatelakis, "New options and insights for survivable transport networks," *IEEE Communications Magazine*, vol. 40, pp. 34–41, Jan 2002.

[20] K. Zhu, H. Zang, and B. Mukherjee, "Design of WDM mesh networks with sparse grooming capability," in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 3, pp. 2696–2700 vol.3, Nov 2002.

[21] J. Strand and A. Chiu, "RFC 4054, impairments and other constraints on optical layer routing." IETF Network Working Group, May 2005.

[22] A. Beshir, F. Kuipers, A. Orda, and P. V. Mieghem, "Survivable routing and regenerator placement in optical networks," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on*, pp. 684–690, Oct 2012.

[23] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "SNDlib 1.0–Survivable Network Design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010. http://sndlib.zib.de.

[24] R. Mendes, B. Nogueira, T. Gomes, L. Martins, and J. Santos., "Impairment-aware optimization strategies to dimension optical transport networks with minimal regeneration requirements," in *8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Oct 2016. Accepted paper to be presented.

[25] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.