Technische Universität München
Lehrstuhl für Kommunikationsnetze

Efficient Optimization Methods for Communication Network Planning and Assessment

Dipl.-Ing. Univ. Moritz B. Kiese

Vollständiger Abdruck der von der Fakultät Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. Ralph Kennel
Prüfer der Dissertation:	1.	Univ.-Prof. Dr.-Ing. Jörg Eberspächer
	2.	Prof. Thomas K. Stidsen, Ph.D. Technical University of Denmark

Die Dissertation wurde am 01.10.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 04.02.2010 angenommen.

I have always found that plans are useless,
but planning is indispensable.

– Dwight D. Eisenhower

Abstract

In this work, we develop efficient mathematical planning methods to design communication networks. First, we examine future technologies for optical backbone networks. As new, more intelligent nodes cause higher dynamics in the transport networks, fast planning methods are required. To this end, we develop a heuristic planning algorithm. The evaluation of the cost-efficiency of new, adapative transmission techniques comprises the second topic of this section. In the second part of this work, we optimally plan and assess numerous protection methods with column-generation and branch-and-price. The former method forms also the foundation of an extensive planning method to determine optimal bounds on the connectivity of wireless mesh networks with beamforming antennas, which we develop in the last part of this work and employ it to assess existing distributed heuristics.

Kurzfassung

In dieser Arbeit werden effiziente mathematische Planungsmethoden zum Entwurf von Telekommunikationsnetzen entwickelt. Zunächst werden zukünftige Technologien für optische Transportnetze untersucht. Da neue, intelligentere Netzknoten eine höhere Dynamisierung der Transportnetze zur Folge haben, werden schnelle Planungsmethoden benötigt. Hierfür wird ein heuristisches Verfahren entwickelt. Die Untersuchung der Kosteneffizienz von neuartigen adaptiven Übertragungsverfahren im Gesamtnetz bildet den weiteren Schwerpunkt dieses Abschnitts. Verschiedene Fehlerschutzmechanismen werden im zweiten Teil der Arbeit mittels Column-Generation und Branch-and-Price-Verfahren optimal geplant und bewertet. Erstere Methode bildet auch die Grundlage eines umfangreichen Planungsverfahren, das Referenzwerte zur Konnektivitätsanalyse von drahtlosen Mesh-Netzen mit Richtantennen liefert, das im letzten Teil der Arbeit entwickelt und zur Bewertung bestehender verteilter Heuristiken eingesetzt wird.

Acknowledgements

During the course of this thesis, I was fortunate enough to be supported by superiors, colleagues, and partners, many of whom have become far more for me than their formal role might imply.

First and foremost, I am indebted to Prof. Dr.-Ing. Eberspächer who created the Institute for Communication Networks or “LKN”, which was at the least my academic home during the last four years. Without his steady support and astonishing confidence in my abilities and the resulting freedom to follow my own ideas, this thesis simply would not have been possible.

During INFORMS Telecommunication Conference in spring 2006, I was introduced to Prof. Thomas K. Stidsen, PhD, who showed great interest in my work with Claus Gruber. While being a visiting researcher at Nokia Siemens Networks in fall 2006, we started to investigate the possibilities of branch-and-price algorithms for network planning problems. The then newly started cooperation between TU München and DTU financed a number of mutual research visits, during which we delved deeper and deeper into the actual implementation of branch-and-price algorithms. I am more than grateful for his offer to be the second examiner of my thesis.

I am thanking Prof. Dr.-Ing. Ralph Kennel for presiding my thesis committee.

Thanks to Prof. Eberspächer’s efforts, at LKN I found many partners for discussions about networks, computers, programming, science in general, students, coffee and all other topics of vital importance for upcoming scientists. A very special part of the culture at LKN is the lively contact to our alumni (or a bit less flattering, but with similar affection “oldies”). Some of these discussions resulted in new research, others in barbecues, and other were “just” an exchange of ideas, but I utterly enjoyed all of them. In this sense I am specially indebted to Dr. Claus Gruber, Dr. Christian Hartmann, Dr. Martin Maier, Dr. Carmen Mas Machuca, Dr. Dominic Schupke, Dr. Robert Vilzmann, Jan Ellenbeck, Oliver Hanka, Julius Kammerl, Silke Meister, Bernd Müller-Rathgeber, Robert Nagel, Robert Prinz, Matthias Scheffel, and Christoph Spleiss.

During the first three years, I participated in the BMBF-funded research project EIBONE in a subcontract of Siemens Corporate Technology and later on Nokia Siemens Networks. During this time, Dr. Achim Autenrieth and Thomas Engel were important partners for discussion and started to build my knowledge of optical networks. Furthermore their constant interest and constructive criticism helped me enormously not to drift away into the domain of purely mathematically interesting problems and to keep useful applications in mind. Apart from this main team, I would like to express my gratitude towards Dr. Marco Hoffmann, Sascha Kallin, Dr. Thomas Michaelis, and Matthias Schuster.

I furthermore would like to thank my students Adriana Bocoi, Juan Camilo Cardone Restreppo, Elisabeth Georgieva, Julian Lamberty, Velislava Marcheva, and Michal Markowski. It was a pleasure to be your advisour.

Although I spent quite some time at institute, I am indebted to my friends outside the institute, who provided distraction whenever I needed some (which might have become obvious to me only afterwards). I am lucky to have you.

Just as my academic advisours showed great trust in my abilities, my family did the same and has been even before I started my journey in the academic world. Without your support, encouragement and wisdom this thesis would have never happened. Last but not least, thank you my dear Caro for being there whenever I needed you.

Contents

1 Introduction

Since the very beginning of telephony and the experience that connecting one customer to every other customer in a phone network is not the most economical way of providing a phone service, *planning* has been an important aspect for network operators. A. K. Erlang's ground breaking work in this field can be viewed as the stepping stone for the scientific interest in communication network planning.

Unfortunately, planning methods, costs and cost-savings of large (commercial) communication networks are well-kept secrets, making it hard to assess the impact of planning on the business of network operators. To German researchers however the fruits of network planning are evident every day: German universities and research institutes are connected via a fast yet cost-effective networks which could not have been realized without mathematical planning methods **[EKW08]**. In a more commercial context, in 2000 AT&T claimed to have saved "hundreds of millions of dollars" [ACD+00] using an advanced method to plan backup capacities in their US backbone network.

Given the fact that communication network providers obviously already behave much more rational than their counterparts in other industries (as for example the railway industry where mathematical planning has only reached awarenes [BGJ08]), why is communication network planning still interesting as a research topic? The answer to this question can be condensed to two words: *constant change*. To fully grasp the impact of this statement, it is important to understand that we plan real-life networks, which do not easily follow abstract models. Network technologies however are constantly evolving. As a consequence, the planning problems we have to solve are changing as well. On top of that, our planning capabilities are improving: Taking *Linear Optimization* (which we will strongly use in this thesis) as an example, mathematical improvements can claim a speed-up of a factor of 1000 in the last 20 years, **not** taking the speed-ups of the computer hardware into account [Bix02]. This factor of 1000 is available within general purpose solver software such as ILOG CPLEX [CPL09]. Even larger improvements have been made with problem-specific algorithms. Consequently, we need a thorough understanding of both — the underlying technologies and the mathematical methods to be able to come to solutions for realistic problems.

The desire to find good solutions is high: On the one hand, Service Providers are forced to offer a service as cheap as possible to be able to survive in a highly competitive market. On the other hand hardware vendors have to be able to provide cost-efficient solutions to get tenders in a global competition. We want to emphasize this last point: Since hardware vendors have to provide a cost-efficient solution, not just a good price for a given list of hardware, a considerable interest to assess different hardware options from a network perspective already arises during the development process.

1.1 Contributions

From this perspective, we can summarize the contributions of this thesis into three key aspects:

- We present and solve a number of planning problems that result from possible future hardware developments (for example Optical Orthogonal Frequency Division Multiplex (OOFDM) or beam-forming antennas in wireless mesh networks). Our provable optimal solutions can serve as a reference point in the evaluation of upcoming hardware – before the actual hardware has been built and deployed.
- We present and employ state-of-the-art planning algorithms (column-generation and branch-and-price) for planning problems which are interesting from a network engineering point of view. In order to facilitate the further use of these algorithms we use a consistent formulation which highlights the similarities between seemingly completely different scenarios.
- Based on optimal solutions we propose enhancements to current practices (such as dual-homing protection or failure localization) and identify promising directions for future developments (distributed heuristics in wireless mesh networks with beamforming antennas).

1.2 Structure of This Thesis

The remainder of this thesis is organized as follows:

In *Chapter 2* we will give an overview of today's communication networks and present the basic network architecture used throughout the remainder of this thesis. Subsequently we discuss according to which criteria a network can be assessed. Building upon this foundation, we will introduce our definition of network planning which embraces a considerably larger set of planning problems as "network planning".

Chapter 3 will serve as the foundation stone of the methodology used in the main chapters. We will introduce Linear- and Mixed Integer Programming, present state of the art solution algorithms. A presentation of the basic network models and their formulation concludes this chapter.

In *Chapter 4*, we focus on capacity planning in optical backbone networks:

- In Section 4.1, we propose a heuristic approach to finding good solutions quickly in traffic grooming problems arising in translucent optical backbone networks and resulting multi-layer scenarios.
- Section 4.2 evaluates possible next-generation adaptive transmission technologies for optical networks from a networking perspective. We solve the arising

planning problems as Mixed-Integer Linear Program (MIP) using column generation techniques and will demonstrate that adaptive technologies offer a decisive cost-advantage over conventional Dense Wavelength Division Multiplex (DWDM) transponders.

In *Chapter 5* we shift our focus from the optical layer to resilience methods and failure reaction:

- Having identified Shared Backup Path Protection (SBPP) as a promising approach to reducing the protection costs in backbone networks during a short overview of existing resilience schemes in Section 5.1, we will investigate the arising planning problem in Section 5.2. In order to solve the resulting MIP-formulation despite the difficult structure of the optimization problem, we develop a new branch-and-price algorithm, which we prove to be efficient in the concluding case study.
- Dual-link failure protection might become an important issue with the rising importance of communication methods and growing communication networks. Hence we will investigate the use of *dual-homing* in Section 5.3.
- In the last section of this chapter 5.4, we present a MIP-based planning algorithm for the placement of monitoring equipment in transparent or translucent optical networks. Failure reports of the end-nodes can only localize a failure to be between these very nodes, which can be more than 1000 km apart with current DWDM equipment, thus making extra monitoring equipment necessary.

In *Chapter 6*, we will examine the connectivity in wireless mesh networks. In a case study presented in Section 6.1 based on real-life measurements, we can show that connectivity is a challenge in urban environments with current technologies. In order to improve this situation, beam-forming antennas have been proposed in previous works. We will thoroughly study the effects of the new degree of freedom (the direction of the main-beam) and develop a planning algorithm capable of providing optimal bounds on the connectivity of a given scenario. Extensive simulations to assess the efficiency of existing distributed heuristics and the influence of shadow fading on this scenario conclude this section.

We will summarize our findings and highlight some of the possible key-issues of future network planning problems in *Chapter 7*.

2 Network Design

"Form Follows Function."[1]

Within the first part of this chapter, we will examine the network structure of large communication networks. Starting with a brief review of the history of the Internet, we will highlight those basic principles, which still have a significant influence on the architecture of current backbone networks, which we will demonstrate in a short overview of such a (possible) architecture, resulting in the basic architecture used in this thesis. Subsequently, we will discuss how other networks or users are connected to these backbone networks. On this basis, we will show how different solutions for a similar problem can be assessed, thereby discussing various cost models and other important criteria towards "good" network design. In the last section we will give an outline of *network planning* as an important step of a network design process.

2.1 Network Architectures

2.1.1 Historical Background

The Internet has undergone tremendous changes from its very modest beginnings in 1969 (when it was a two node network) to its incredible size of today. Nevertheless, the last technological update requiring an update for *every* user was the switch to IPv4[Def81] on January 1st, 1983 — more than 25 years ago. Considering that for the International Standards Organization (ISO), the Internet of the end-70s appeared to be a mere "academic toy network" [HL96, p. 247], this has to be conceived as quite an achievement.

What made this impressive technological longevity possible? In order to answer this question, let us take a look at the environment the Internet was developed in. Even in the most simple network imagineable (consisting of the two hosts cited in the beginning), two different computers were involved and with the growth in the subsequent years, this scenario became more and more complex[2]. Hence, we can safely state that a certain degree of *hardware independence* was absolutely necessary.

[1]Louis Sullivan (1856-1924) in "The tall office building artistically considered", *Lippincott's Magazine*, March 1896

[2]By the end of the 1970s, some of the most prominent computer systems in academia were Digital's PDP/10 running TOPS-10, TOPS-20 or ITS, PDP/11 running UNIX or VMS, IBM S/370 running OS/MVT and many, many more. All of these machines had significant differences in both software and hardware.

Another important point is that the experimental network was also the working network: There was simply no dedicated test network at that time. Although this might appear to be simply an inconvenience, it enforced *modularity*, as it was hard to require large portions of software to be changed to add or improve some functionality because this would have affected many users. Furthermore, for example a mail protocol that would take down the entire node in case of an error would not gain much popularity either.

The last point already hints at the last observation we want to highlight: *implementation*, i. e. in order to gain acceptance protocols had to have a showcase or a proof of concept. This was a large departure to the approach taken by the ISO and other standardization bodies. When they presented the Open Systems Interconnect (OSI)-model, it was little more than a design and the first full implementation known to the author was demonstrated in 1987 with DECnet Phase V.

Modularity and hardware-independence are usually considered good practice in software engineering, however as we tried to illustrate above, there might have been less intention behind this design than pressure from the environment. To summarize, the Internet protocol suite offered a practical solution to a problem (connect computers) and was sufficiently well-designed that there was never enough pressure to restart from scratch despite the more and more evident shortcomings.
We should further note that although ISO took quite a while to design the OSI-model, their model is far from being flawless and exposed some of the errors (political decisions instead of technical ones[3]) that caused severe trouble for Asynchronous Transfer Mode (ATM) some years later. Nevertheless the *OSI-model* is considerably more general than the model behind TCP/IP and thus (although sometimes modified as in [Tan03]) still in widespread use.

One of the more obvious aspects, we can easily discover in today's Internet (and many other communication networks) is the notion of layers. In this showcase of moularity, one layer offers a set of capabilities (which sometimes are also called "service") to the next layer above. With this simple mechanism, changes in one layer will only affect the next upper layer. Naturally, this layering comes with the usual price of abstraction: More overhead (caused by additional headers, . . .) in every layer and a certain *loss of information* from layer to layer. Since one layer will not provide all information it has to the upper layer (otherwise we will not gain any modularity), we will loose possibly useful information (for example IP cannot directly provide any information about the physical channel quality).

In the remainder of this thesis we will use the term *layer* not in its strict ISO/OSI sense, but in a more general meaning that will be probably self-explanatory after the next section. The rule of thumb for this use is that every protocol which has its own hardware platform is considered as a layer: For example IP is a layer, TCP not (since we have IP routers).

[3]For example Tanenbaum [Tan03] attributes the seven layers in the OSI-model to politics rather than to technical thoughts. Otherwise two overfull and two almost empty layers would be hard to explain.

2.1.2 Vertical Layers

This notion of *layering* is still prominent in networking as it was some 25 years before. Modern *backbone-networks* are not built exclusively any more for one application any more, they have to transport a number of different services and thereby serve many different kinds of customers at the same time. An *example* backbone vertical layer structure (which is part of our network architecture) is depicted in Figure 2.1. Starting from below, we can see the following layers:

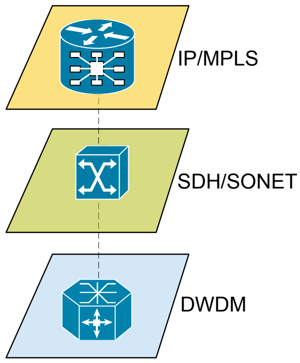


Figure 2.1: Example vertical layer structure

DWDM

At the very bottom we have an optical DWDM network. DWDM networks build the foundation stone of today's backbone network offering huge data-rates at viable costs. Modern transmission systems can multiplex up to 160 channels with data-rates of up to 40 Gbit/s each over distances of up to 3000 km. It is worth noting however that the boundaries between "typical" metro/regional and backbone equipment are becoming more and more fuzzy and might merge to one "super-platform" in the future [CS07]. A typical transmission system is shown in Figure 2.2.

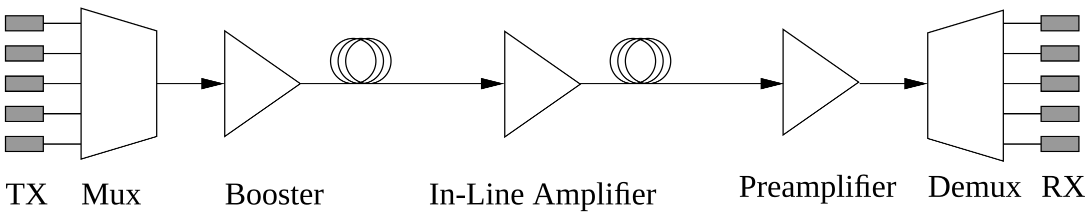


Figure 2.2: DWDM Transmission System

A client signal, which is usually referred to as a *grey* signal, arrives at the DWDM *transponder* TX. Typical client interfaces are (Carrier-Grade) Ethernet or SDH/SONET. The transponder "translates" this signal to a DWDM signal with a fixed bandwidth (typically 2.5 Gbit/s, 10 Gbit/s or 40 Gbit/s) and transmits this signal on one of the eligible wavelengths (which we can also interpret as the *colour* of the signal) to the *multiplexer*. The multiplexer Mux can be seen as an "inverse prism": It multiplexes all the coloured signals from the transponders on the client side on one fibre on the trunk side. Each of the different wavelengths may be used with a different data-rate, which is – considering the large price differences between the necessary transponders – an important cost-factor. As a matter of fact, conversion from the optical to the electrical domain or vice-versa, may it be for light-path termination or regeneration is one of the most important influences for the resulting equipment costs of a network. Besides differing in data-rate and range, transponders can be tunable or limited to one fixed wavelength. While the former offers greater flexibility and greatly simplifies the keeping of spare parts, the latter can be significantly cheaper.

On the way to the opposite transmission system, the signal has to be amplified in regular intervals due to signal degradation. Since amplifiers cannot differentiate between signal and noise, they amplify both, which imposes an upper limit on the transmission length. This again depends on the signal quality generated by the transponder in the source node. If this limit is reached, the signal will have to go through so-called *3R regeneration* (reamplifying, reshaping, retiming) which is usually performed via two transponders in a back-to-back configuration. The costs of this full O/E/O-conversion (optical/electrical/optical) are quite notable, which leads to ongoing research for an all-optical 3R regeneration [TBFC+04, SST07].

The termination of the light-path is symmetric to its source: A demultiplexer Demux spreads the different wavelengths on a fiber and a transponder RX translates the trunk signal back to a client signal.

Only customers requiring very high data-rates and having other requirements (privacy, special protocols, etc.) would buy a DWDM-connection (i. e. a wavelength) directly. We will examine in Section 4.1 heuristic planning methods for DWDM-networks with increasing cross-connection capabilities, in Section 4.2 the influences of new, adaptive transmission methods and in Section 5.4 failure localization in transparent or translucent DWDM-networks.

Before we move on to the node architectures of the next layer, let us first inspect the topology from the perspective of the next layer, which we illustrate in Figure 2.3. We can see two paths on the DWDM-layer: A *transparent* light-path between nodes A_{DWDM} and D_{DWDM} and an *opaque* connection between D_{DWDM} and E_{DWDM} respectively. The transparent light-path is one single wavelength, which passes transparently through the intermediate nodes, i. e. these nodes do not alter the signal in any way. The consequence of this configuration is that in the next layer, node A_{SDH} appears to be directly connected to D_{SDH} just as D_{SDH} and E_{SDH}. This hiding of information about the underlying topology is typical for layered networks. In this work, we will address the cost-efficient construction of such DWDM layers in Chapter 4 and will examine some of the consequences for failure localiziation in Section 5.4.

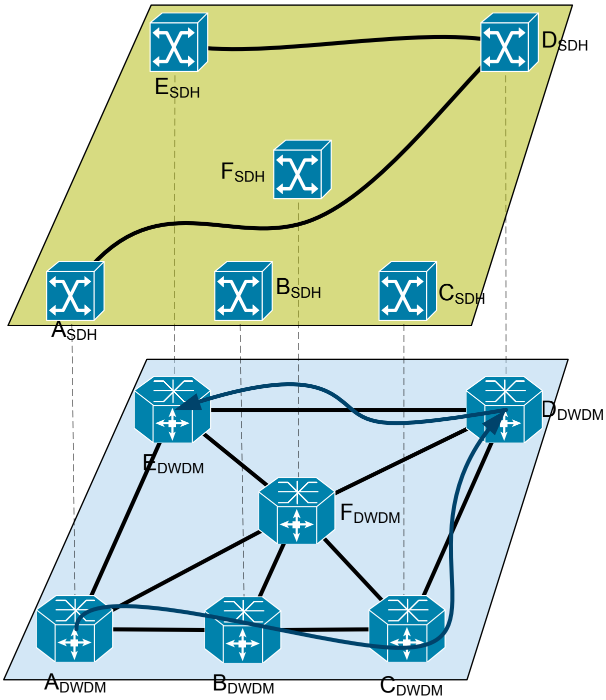


Figure 2.3: Interaction between the WDM and the SDH layer

SONET/SDH

The next layer in our example architecture is Synchronous Optical Networking (SONET) (in the US) or Synchronous Digital Hierarchy (SDH) (in Europe). Although it contains specifications for optical interfaces (which in return are used on DWDM-transponders as client interfaces), SONET/SDH works on the electrical domain, which of course increases the complexity, but also offers more flexibility than mere wavelengths. If SONET/SDH is used in conjunction with DWDM, then one coloured signal usually corresponds to one SONET/SDH connection, for example a 2.5 Gbit/s transponder has an OC-48 client interface, a 10 Gbit/s transponder an OC-192 and 40 Gbit/s transponder an OC-768 interface, respectively. SONET/SDH supports a very flexible multiplexing scheme, which is capable of multiplexing signals from 155 Mbit/s into the above streams without fully demultiplexing the entire traffic stream (which was necessary with its predecessor Plesiosyncronous Hierarchy (PDH)). The flexibility becomes even more eminent once we take into account, that slower bitrate PDH connections can be transported via SONET/SDH as well.
In addition to this, SONET/SDH can offer far more advanced switching, protection and automation facilities than DWDM. Naturally this flexibility comes at the cost of far more complex electronic components thereby resulting in higher equipment costs.

IP/MPLS

In order to understand the fundamental difference between technologies such as DWDM, ATM, SONET/SDH on the one side and Internet Protocol (IPv4) on the other side, it is helpful to understand the difference between *connection-oriented* and *connection-less* networks. In DWDM for example, a connection has to be created before two nodes can communicate, i. e. a wavelength has to be chosen. In the case of a translucent network, the intermediate nodes have to be configured to pass the chosen wavelength through, etc. Once the connection has been established, only these two nodes can use the resources, the again the usage of this resource is *guaranteed* to them. All in all, this procedure is comparable to Plain Old Telephone Service (POTS), where one has to dial a connection, and the dialing establishes the connection. A IPv4 *packet-network* on the other hand, decides node-by-node how to forward a packet just based on the destination which is contained in the packet header. A subsequent packet with an identical destination may take a completely different path.

Multiprotocol Label Switching (MPLS) is an approach to take ideas from *connection-oriented* technologies (such as ATM) to the *connection-less* domain of IP. This technology was expected by many experts coming from data-networking (as opposed to those coming from phone-networking) to fulfil the promises ATM made but never met. When traditional telecommunications-providers deployed ATM-based backbone networks, they faced severe problems[4] with early ATM switches [Ste96]. The upside of the ATM switches was, that they could handle larger bandwidths than IP-routers[5]. Thus the next step was quite logical: Use comparably cheap ATM switches and provide the possibility to control their switching fabrics via IP routing protocols. Three companies (Cisco, IBM and Toshiba) followed the pioneering start-up Ipsilon Inc. (acquired by Nokia in 1997) with similar ideas, which eventually resulted in the standardization through the Internet Engineering Task Force (IETF) under the name MPLS.

The technology as it is used and standardized today is based on fixed-length *labels*, which are used to determine the next hop, when a packet has to be forwarded. By adding such a label to the incoming data (i. e. an IP packet) at the border-routers (i. e. the routers at the edge of the backbone), the inner routers do not need to consult the full Internet routing table to determine the next hop – as long as it stays inside the network of an Internet Service Provider (ISP), the label contains sufficient information[6]. Due to this simplification, forwarding an MPLS packet is relatively easy compared to IPv4 routing. On the other hand, this label allows the creation of *paths* through the ISP backbone: When every switch along the path is configured to similarly forward packets with a given label, the path an IP packet takes through an MPLS network is determined by the label it receives at the edge of the network. This abil-

[4]Which were caused by the fact, that the buffers in the ATM switches were too small and traffic did not behave like the Poisson distribution used in the dimensioning of the buffers, but more following the observation of Leland et al. [LTWW94].

[5]provided that not too many of the features of ATM were actually used [Ste96]

[6]Extensions to allow this routing/switching scheme across multiple domains exist as well [FAV09].

ity introduces *connections* to Integer (Linear) Program (IP) networks, a feature that was previously solely reserved to technologies like ATM and SONET/SDH. Further notable extensions include advanced resilience schemes, which we will focus on in Section 5.2.

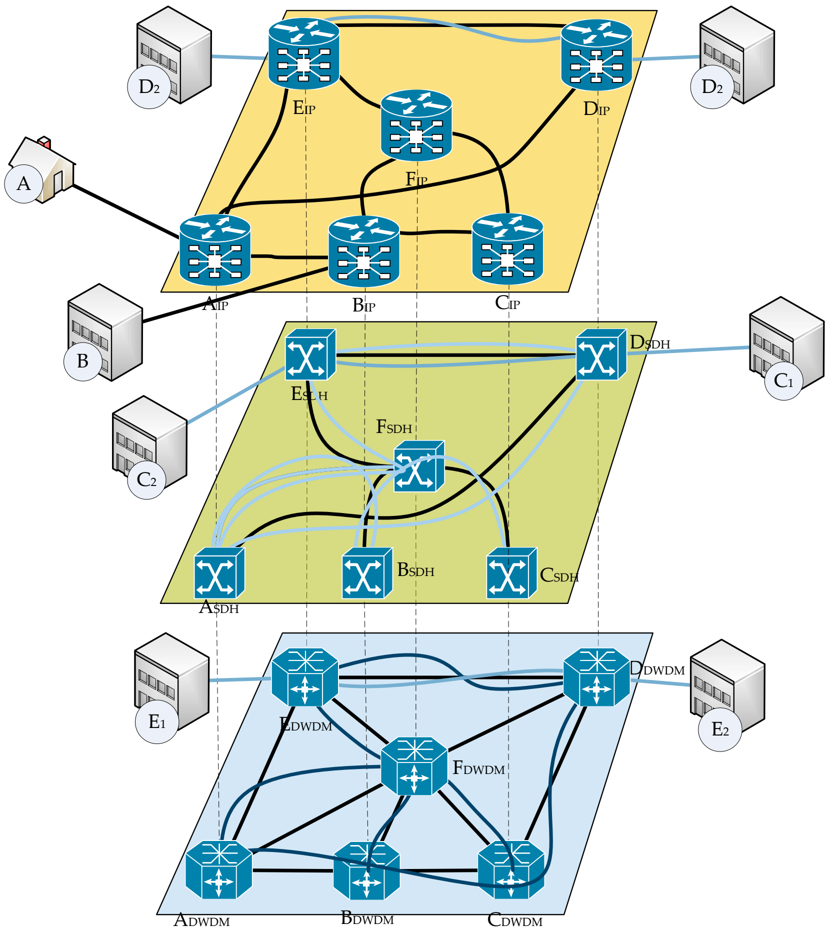


Figure 2.4: Example Backbone

In Figure 2.4, we show how five different services are transported through the same network:

- Customer A could serve as an example for the tech-savy home-user who is connected to the Internet via a cutting-edge Hybrid Wireless-Optical Broadband-Access Network (WOBAN), i. e. a wireless multi-hop connection to an optical network.
- Customer B on the other side is a webhoster. His Internet-connection needs to be considerably faster than the typical home-user interface, because he wants to serve many customers at the same time, which is why he rents an $OC-12$ connection to the Internet. Apart from being faster (and more expensive), this customer still has an IP-only service.
- Customer C_1 and C_2 are two locations of customer C which are connected via a

leased OC-12 connection. Despite the fact, that IP-packets also flow over this link, the customer takes care of all layers above the mere OC-12 connection himself.
- Customers D_1 and D_2 run a private IP-service (also known as a *Virtual Private Network*) among themselves, which has been realized as a MPLS-tunnel.
- Customer E rented a private wavelength between his locations E_1 and E_2.

Basic Backbone Network Architecture Model

In the technology overview given above, we were more elaborate on DWDM and MPLS than on SONET/SDH. The reason for this is quite straightforward: Among providers there is a desire to reduce the complexity of their network architectures, by which they hope to lower both, the operational and hardware costs of their total network. Since a major part of the traffic of an ISP is IP-traffic, it would be difficult for example to get entirely rid of IP and just use SONET/SDH. The necessary border routers, translating IP-destinations to SONET/SDH circuits, would be quite complex and since a 'retranslation' to IPv4 would be necessary at the end-nodes anyways, the cost-savings would probably be non-existent. On the other hand, as layed out in the previous section, MPLS offers a lot of the amenities of circuit-switched networks and is readily integrated in todays IP backbone routers. Carrier Grade Ethernet is expected to offer similar features as MPLS, with the added bonus, that a connection between customer Ethernet networks might be realized on the native layer. However it remains to be seen whether this portion becomes significant enough to justify the additional hardware, which is necessary since switching capabilities for Carrier Grade Ethernet have not yet appeared in IP routers to the best of the author's knowledge. On the other hand IP-over-DWDM has been a research topic for more than nine years [GDW00] and coloured DWDM interfaces for IP routers exist. Hence, we consider the IP/MPLS-over-DWDM model to be of larger interest than for example SDH-over-DWDM. From a planning point of view, Carrier Grade Ethernet and MPLS are similar enough to be treated via identical models, which we will present in 3.3.3. Our layering will therefore be reduced to two network layers: A DWDM network and an IP/MPLS or Carrier Grade Ethernet layer on top as shown in Figure 2.5 We

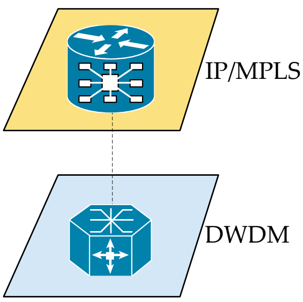


Figure 2.5: Backbone Architecture used in this thesis

shall conclude with the remark, that we do not expect SONET/SDH to become extinct in the near future. Some services are simply too dependent on peculiar features of SONET/SDH (the precise timing comes to mind) which are not offered by other technologies and those customers might generate sufficient revenue to justify a separated SONET/SDH network on the same DWDM infrastructure.

2.1.3 Horizontal Layers

As we can easily discover, our view of the network is yet incomplete: Despite the fact, that customer A and B in Figure 2.4 have vastly different requirements regarding their connection (and pay very different prices) and hence use very different technologies, they appear to be very similar in the above picture. This unification is achieved by upstream *metro-* and *access-networks*, which can be seen as separate entities hierarchically connected to the backbone.

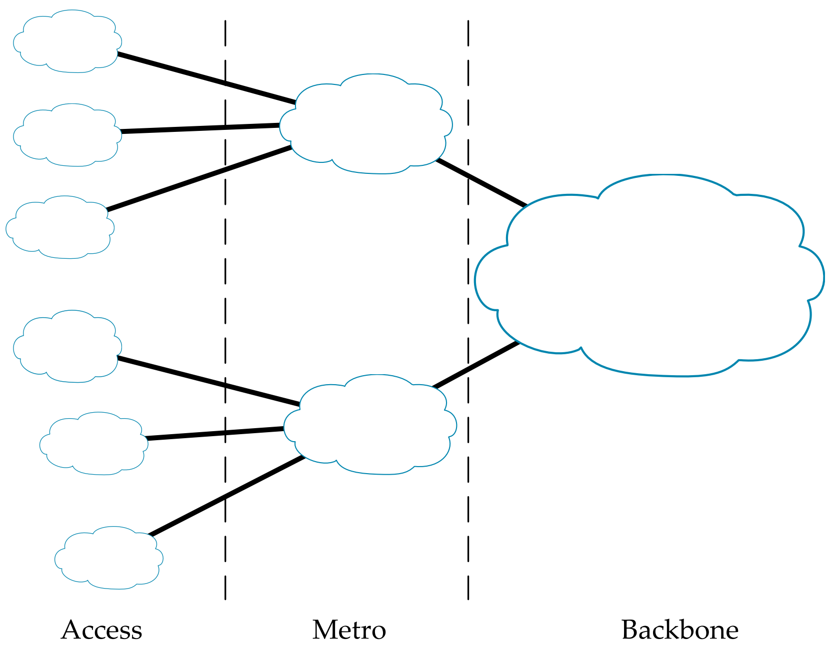


Figure 2.6: Horizontal layering

This modularity greatly simplifies the introduction of new technologies in each of the different clouds in Figure 2.6, which sketch the *horizontal layering*. It is important to understand that the vertical layering may (and will) be different in every horizontal layer: In the previous section, we have illustrated how the vertical layering may look like in the core network. In order to highlight the difference to metro- and access networks, let us reconsider customers A and B from Figure 2.4. A typical WOBAN connection might look like depicted in Figure 2.7. We can see that customer A is connected via two wireless hops to a basestation, which is connected via a Passive Optical Network (PON) to an SDH-based metro network. This metro ring might be connected via an OC-48 connection to the backbone node. The wireless portion is a so-called *wireless-mesh network*. In contrast to todays Wireless LANs (WLANs) which

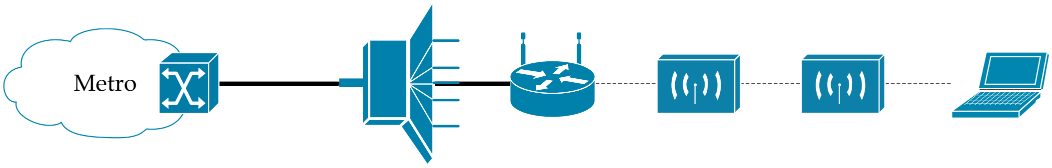


Figure 2.7: WOBAN connection from the end-user to the metro network

are usually based on IEEE 802.11, where every end user is directly connected to a base station, mesh networks allow multi-hop connections. Standardization and integration into 802.11 is still on its way within the task group IEEE 802.11s. Despite its experimental status in the standardization process, there is already considerable commercial interest, leading to mesh-networks realized in higher network layers as research vehicles (like the MIT RoofNet [roo] project, which provided the data basis for the study in Section 6.1) or commercially available solutions, as for example developed by Meraki Inc [mer]. A combination of PONs and a wireless mesh network, wich is called a *WOBAN* has further fueled industrial interest and has been shown to be able to serve as a cost-effective access network [SDM07]. We will examine the link availability of the wireless portion of such a WOBAN in Section 6.1 and develop a methodology to assess and plan wireless mesh networks with beamforming antennas which are a promising way to improve the connectivity in wireless mesh networks.

Going back to the above scenario, customer B on the other hand would completely bypass the access network and connect directly to the gateway to the metro network.

In our example, we have assumed that the metro network is completely SDH-based. Unfortunately this would mean, that even if customer A and B were located on the same metro-network, the connection would go all the way to the first backbone node and then back to the same metro network. We will examine how important it is to consider the fact that most metro networks are connected to two backbone nodes for resilience reasons during the backbone planning in Section 5.3.

2.2 Network Assessment

Now that we have introduced the technological background of network planning, we have to define the goal of our planning. In a nutshell, let us assume that we are given two *network configurations* which meet the same requirements. How can we differentiate between these two solutions and how can we tell which solution is "better"? Building upon this basic question, what are goals that we should pursue besides meeting a set of hard requirements? Throughout this thesis, we will use a combination of hard requirements (which we have to fulfil) or *constraints* and an optimality criterion. Furthermore, we will only consider *feasible* problems, i. e. problems with a solution where all constraints are met.

2.2.1 Feasibility Criteria

Demand

In many of the subsequent planning problems, we will require that a network is capable of handling a certain *demand*. In our case a demand is always interpreted as a request for a given bandwidth from a source to a destination node. Since we can easily order all source-destination relations in a matrix (the column determining the source, the row determining the destination and the entry determining the bandwidth), we will also speak of a *demand matrix*. How these demands are calculated in return will be discussed in the following section.

Resilience

Since Chapter 4 and 5 examine backbone networks, we will require some degree of failure resilience, in our case that a demand can be routed through the network even with a single (Chapter 4 and 5.2) or dual link failures (Section 5.3).

In anticipation of the various mathematical models following in the remainder of this thesis, we want to highlight once more that these two constraints are directly dictated by us. In contrast to these immediate constraints, we will formulate many constraints that arise from technologies, paradigms we impose and so forth, which we deem representative or useful for the scenario we want to examine.

2.2.2 Optimality Criteria

Bandwidth Efficiency

This most abstract criterion, is building upon the notion, that transmission bandwidth usually costs money. Consequently, if we have two feasible solutions, we will prefer the one which consumes less bandwidth. In Section 5.2, we will compare several protection schemes to the unprotected case, which is sometimes also called the *resilience overhead*. With this ratio, we can easily compare the amount of extra bandwidth required to use a given resilience scheme.

Capital Expenditure

One of the main points of criticism towards the previous criterion is that it models bandwidth costs strictly linear. This however might not be a good representation of reality: As our experience tells us, equipment with a medium performance usually offers a good price/performance ratio, whereas very high-end equipment is comparably expensive. Furthermore, we can usually buy equipment only in discrete units (i. e. buying half a router or half a line-card is usually not possible). Hence it seems a lot more promising to directly incorporate equipment prices (which account to the

Capital Expenditures (CAPEX))into our model. Setting aside arising complexity considerations from this, we face a new problem: How do we determine these prices? While this question might seem rather trivial, equipment providers are not very informative regarding their equipment prices. Furthermore vendor- or buyer-dependent rebate systems, make these prices hardly comparable among equipment vendors. To this end, the telecommunication community has defined so-called *cost-models*, which claim to be vendor-neutral[7]. It is important to keep in mind however, that multiple vendors might want to "push" some hardware components, by deliberately assuming lower prices or achieve the opposite by pricing other components higher. From this perspective it is important, to keep the cost model as flexible as possible to be able to cope with different settings.

Operational Expenditure

In recent years, ISPs generally observed that the operational expenditures to run a given network can easily be in the same region as the capital expenditures. A substantial amount of research was started to examine the cost structures arising from Operational Expenditures (OPEX), however up to now, no cost models suitable for network optimization have surfaced – owing to the fact that we (researchers and providers) are just at the very beginning of understanding how their operational expenditures relate to the networks they operate.

Connectivity

In Chapter 6, we will focus on connectivity in wireless networks. While connectivity itself is a feasability criterion (either a network is connected or not), we define a softer criterion, namely the *path probability* to allow further differentiation.

2.3 Network Planning

Having sketched out, what we are going to plan and with which goals, we can now elaborate on the planning itself.

So what do we understand by *network planning*? Previous work, such as Gruber's PhD thesis [Gru07, p. 5] often understood network planning as a task to design a network meeting certain, very specific requirements (such as Quality of Service) and optimality criteria (availability, cost). To be more verbose, Gruber defines at the beginning of his thesis that

> Optimal resilient network planning is the task to design a robust network that enables the transportation of data with strict Quality of Service requirements, lowest possible cost, and highest possible availability.

[7] which is usually achieved by embracing more than one equipment-provider

While we do find much truth in his definition, which for example matches some aspects of this thesis perfectly (e. g. Section 4.1), we do have two points of criticism:

- First, his definition and subsequent, more detailed descriptions resulting in a *planning cycle* are heavily geared towards fixed networks.
- Second, we find much of his definition too specific, to be a general description of fixed-network planning. For example, his own theoretical work on a lower bound on the capacity requirements of resilience does contain a lot of planning, but it does not fit easily into this model which is very much geared towards planning from an ISP perspective.

[EKW08] provides a comprehensive overview of practical fixed-network planning and highlights the differences that appear once specific time-frames are taken into account. A division into long-term (strategic), mid-term (tactical) and short-term planning shows which problems can or have to be considered in which time-frames: At the moment it is simply unrealistic for network providers to renew their fibre infrastructure every two years, hence the design of such an infrastructure has to be valid for longer time-frames than software configuration settings which can (at least in theory) be changed within minutes.

Since this thesis covers a mixture of different network planning problems from both fixed and wireless networks, arising from both practical and theoretical interest, we need a more general view on network planning. We define

> Network planning is the process of designing a network towards a desired set of properties.

We want to discuss a number of consequences of this generalization:

- First of all, since we do not make any assertions of the planning steps itself, we do not need to separately consider the effects of jointly optimizing some of these (for example in Gruber's model separate) steps separately. In our model, we will simply consider the scope of a planning step as a *modelling decision*.
- Second, we want to highlight the "predictive" or "estimative" character of some of the planning problems we consider: The technologies in Section 4.2 or 6.2 do not yet exist, we rather try to predict their performance, which in return might result in a technology pursued for commercialization. Hence following a complete "planning cycle" is impossible, as there is neither market nor users to refine our planning.
- Last, we do not require any human interaction or decision making during our planning. While we probably are just at the very beginning of this line of research, proposals for routing systems that try to find a useful network configuration on their own (such as Gruber's *Self Regulating Traffic Distribution* [Gru07, pp. 65]) exist. In our general definition these systems can be seen as an integrated network planning system.

The point we are trying to make within this section is by no means, that previous definitions of network planning were wrong, we just consider them to be too specific. There are just more network planning problems than meet the eye.

During the remainder of this thesis, we will always use a three-step approach for a network planning problem:

Figure 2.8: Planning Steps

1. Our first step consists of defining a *network model*. This model describes (usually in a non-mathematical way) the problem we are trying to solve, necessary assumptions on the technical side and required input data such as network topologies, demands, etc.
2. The second step consists of a translation of the network model into mathematical formulation usually geared towards *mathematical programming*. Together with a definition of our objective(s), we will call the result of this step the *optimization model*.
3. In the last step, we will specify and use algorithms to solve the optimization model adequately. In the majority of cases later on in this study we will employ problem-specific MIP methods.

In the remainder of this section, we will present and discuss specific modelling assumptions used in our planning problems later on in this thesis, especially in Chapters 4 and 5.

2.3.1 Fibre Topologies

For the backbone networks considered in Chapters 4 and 5 the research community has defined so-called *reference networks* which represent fibre networks resembling the infrastructure of ISP. The three topologies which we will use repeatedly have been defined in joint research projects of ISPs, universities and hardware vendors such as NOBEL [NOB04] and EIBONE [EIB05] and are therefore deemed to be as realistic as possible without using real network data. All networks are available from the SNDLIB [OPTW07]-website [snd09].

The use of commonly used reference networks is supported by a few practical reasons: First of all, no privacy issues regarding the publication of results as for example with the first topology used in Section 4.1, where concerns about the publication of the underlying fibre topology existed, will arise if we only use publicly available data. Furthermore results will be at least to some extent comparable – very much depending on the demand and cost models used.

2.3.2 Demand Models

An important portion of the modelling process is the user demands which we summarize in the *demand model*. Substantial research exists on modelling and predicting both phone and data traffic, since realistic traffic measurements are usually considered to be highly confidential due to competition among providers and regulatory bodies, especially for several kinds of services at the same time. All demands in the considered reference networks have been generated according to a model by Dwivedi and Wagner [DW00] with the respective parametrization performed by T-Systems. The output of this model is plain and simple the "average revenue-bearing traffic". Considerable debate exists on how these traffic predictions (keeping in mind that we usally want to plan a future network) have to be incorporated into a network planning process.

We consider this to be a very important modelling decision in real-life instances where a network provider decides on how to dimension his network. The more simpler choices are rules of thumb, which have been used successfully in the past (i. e. the average load on an IP link should only be 50%) to more complex traffic models differentiating between different kinds of traffic with different requirements (throughput, delay, etc) as for example discussed by Riedl [Rie03] or traffic definitions that completely deviate from the traditional demand matrix structure such as the *hose model* for Virtual Private Networks (VPNs) introduced by Duffield, Goyal and Greenberg in [DGG+99].

While this modelling decision is important for the actual dimensioning of a real network, we consider it to be of minor importance for the results of this thesis: All our planning methods work as long as we can represent the traffic model by a demand matrix and capacity constraints by link- or path-capacities. To this end we use the demands from the reference networks directly as it was intendended within the EIBONE project and is done by similar work, for example by Orlowski [Orl09].

2.3.3 Cost Models

As we have already argued in 2.2, it might be quite desireable to integrate CAPEX right into the planning process to adequately model the differences between technological alternatives.

While numerous cost models for different technologies existed at that point no efforts had been made towards a concise and unified description of such models – quite contrary to fibre topologies for which a number of description languages such as GML and XML-dialects exist and are actively in use [snd09]. Quite obviously this lack enormously increased the amount of work necessary to switch between cost-models, which might be very useful for all possible parties (vendors, service providers, software providers) involved. To this end an effort was started within the working group "Referenznetze" of the EIBONE project to cumulate existing cost models and at least start the definition of a common language. During this process vast differences between different cost models became obvious, leading to further insight into cost-modelling. Within this effort we learned, that we were not only trying to define a cost-model (which can be thought of as an equipment price-list in its most simple case), but more or less had to define a formal specification of the multi-layer planning process we were going to take, since the cost model influences a huge portion of the resulting optimization model[8]. Having learned this lesson, we will resort to comparably simple cost models for this thesis, since our goal is to find more fundamental behaviour of network technologies, which is harder to condense from more complex models.

2.3.4 Multi-Layer Models

With the exception of the our heuristic algorithms, we restrict ourselves to very simple two- or three-layer problems which are far from the complexity generated by a vertical-layer hierarchy as presented in the first section of this chapter. We chose this for two main reasons: Traffic models for the available reference networks do not differentiate between traffic arriving at different layers, however in a study [EAB09] resulting from a joint project between Nokia Siemens Networks and Swisscom, Engel, Autenrieth and Bischoff showed that the network load has significant influence on the cost-effectiveness of the layer structure. Furthermore, the majority of our planning problems arise from fundamental questions regarding the cost-effectiveness of certain technologies or methodologies. We do not believe that we would gain additional insight if we complicated the underlying structure further more. Another argument exists against (a yet hypothetical) planning process which could plan and optimize all layers in one step: It might be close to impossible for ISPs to realize and support this configuration, because the organizational changes required to build configurations whith *heavy* interdependencies will be considerable since very little software assistance is available for this multi-layer approach.

2.4 Summary

In the beginning of this chapter, we showed that some key priniciples in the early design stages of the original Internet made long-term evolutionary technological improvements of backbone communication networks possible. The result of this development are today's complex network architectures, which have to be handled by ISPs as cost-effective as possible. In order to be able to meet customer requirements, numerous planning steps are necessary. While these planning steps are challenging enough as they are, it is more than likely that the number of planning problems will rise in the future, since network aspects have to be considered during the design of new technologies.

[8] The result of this efforts are actively used by atesio, an optimization spin-off from ZIB.

3 Optimization

"I hate definitions."[1]

In this chapter, we will present the mathematical foundation of this thesis. In the first section of this chapter, we will offer an introduction to *Linear Optimization*, discussing both continuous and integer programs and corresponding respective solution methods. In the last section, we will show after a brief overview of relevant graph theoretical issues, how we can formulate basic network planning problems as linear programs.

3.1 Linear Programming

Systems of linear inequalities have been known and studied since the beginning of the 19th century – Fourier developed the first known algorithm to solve linear arithmetic constraints 1824 [Fou27]. Unknown in the west, the Soviet mathematician Kantorovich worked on planning problems in a centrally controlled economy, which resulted in linear programming formulations and, even more important, an algorithm to solve these linear programs in 1939[2].

The jump-start for linear programming however were Dantzig's works regarding military planning problems in the aftermath of WW II. The *Simplex Method* invented in 1947 is still the foundation stone of most available solvers.

[1] Benjamin Disraeli, *Vivian Grey*

[2] The first English translation appeared 21 years later [Kan60]. Late recognition was awarded to Kantorovich with the Nobel Prize in Economics which he shared with Tjalling Koopmans in 1975 [LKS91].

3.1.1 The Standard Form of Linear Programs

A Linear Program (LP) is an optimization problem consisting of a linear *cost* or *objective function*[3] and linear *constraints* (equalities and inequalities).

$$\tilde{\mathbf{a}}_i^T \tilde{\mathbf{x}} \geq b_i, \quad i \in \mathcal{M}_1, \tag{3.1a}$$
$$\mathbf{a}_i^T \tilde{\mathbf{x}} \leq b_i, \quad i \in \mathcal{M}_2, \tag{3.1b}$$
$$\mathbf{a}_i^T \tilde{\mathbf{x}} = b_i, \quad i \in \mathcal{M}_3, \tag{3.1c}$$
$$\tilde{x}_j \geq 0, \quad j \in \mathcal{N}_1, \tag{3.1d}$$
$$\tilde{x}_j \leq 0, \quad j \in \mathcal{N}_2., \tag{3.1e}$$

with the variables $\tilde{x}_j \in \mathbb{R}, j \in \mathcal{N}_1 \cup \mathcal{N}_2$. Any solution within these variables is called a *feasible solution*, however a feasible solution is not necessarily optimal regarding the cost function

$$\min \mathbf{c}^T \tilde{\mathbf{x}}, \tag{3.1f}$$

where $\mathbf{c}$ is called the *cost vector*. Maximizing $\mathbf{c}^T \tilde{\mathbf{x}}$ is obviously equivalent to minimizing $-\mathbf{c}^T \tilde{\mathbf{x}}$. By setting $\mathbf{a}_i = -\tilde{\mathbf{a}}_i, i \in \mathcal{M}_1$ we can reformulate (3.1a) to take the form of (3.1b). Furthermore, we can replace the equalitites (3.1c) by two inequalities $\mathbf{a}_i^T \tilde{\mathbf{x}} \leq b_i$ and $\mathbf{a}_i^T \tilde{\mathbf{x}} \geq b_i, i \in \mathcal{M}_3$, and finally consider (3.1d) and (3.1e) as special cases of (3.1b). Replacing unrestricted variables with the difference of two positive variables $\tilde{x}_f^{(1)} - \tilde{x}_f^{(2)}$, so that $\tilde{\mathbf{x}} \geq 0$ and placing all $\mathbf{a}_i^T$ as rows in a matrix $\mathbf{A}$ gains the so-called *inequality form* of an LP

$$\min \mathbf{c}^T \tilde{\mathbf{x}} \qquad \text{subject to} \tag{3.2a}$$
$$\mathbf{A}\tilde{\mathbf{x}} \leq \mathbf{b} \tag{3.2b}$$
$$\tilde{\mathbf{x}} \geq 0. \tag{3.2c}$$

Now we can replace every inequality $\mathbf{a}_i^T \tilde{\mathbf{x}} \leq \mathbf{b}$ with an equality using an additional *slack variable* s_i leading to constraints

$$\mathbf{a}_i^T \tilde{\mathbf{x}} + s_i = \mathbf{b}, \quad s_i \geq 0. \tag{3.3}$$

Joining s and $\tilde{\mathbf{x}}$ to $\mathbf{x}$, the program

$$\min \mathbf{c}^T \mathbf{x} = z \qquad \text{subject to} \tag{3.4a}$$
$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.4b}$$
$$\mathbf{x} \geq 0 \tag{3.4c}$$

is called to be in the *standard form*, which is more convenient for solution algorithms. From a geometric point of view $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$ is a polyhedron.

[3]In general, the cost function has to be piecewise linear and convex as shown in [BT97, pp. 15].

3.1.2 Duality

The notion of *duality* was introduced early in the theory of LPs by John von Neumann in conversations with George Dantzig in October 1947 "and appears implicitly in a working paper [von47] he wrote a few weeks later" [Dan63, p. 123]. In the context of duality, we will call (3.4) the *primal problem*. Relaxing the original constraint (3.4b) by placing a penalty $\mathbf{p}$ on the difference between $\mathbf{Ax}$ and $\mathbf{b}$ transforms the problem into

$$\min \mathbf{c}^T\mathbf{x} + \mathbf{p}^T(\mathbf{b} - \mathbf{Ax}) \qquad \text{subject to} \tag{3.5a}$$
$$\mathbf{x} \geq \mathbf{0}. \tag{3.5b}$$

Obviously, the relaxed problem allows more options in creating a feasible solution, while every feasible solution of the original problem is a feasible solution of the relaxed problem, but not vice versa. Hence, we can quite intuitively deduce that the value of the optimum for the relaxed problem cannot be larger than for the optimal solution $\mathbf{x}^*$ of the original problem, i. e.

$$\min_{\mathbf{x}\geq\mathbf{0}} \left[\mathbf{c}^T\mathbf{x} + \mathbf{p}^T(\mathbf{b} - \mathbf{Ax})\right] \leq \mathbf{c}^T\mathbf{x}^*. \tag{3.6}$$

In other words, the left hand side of the inequality provides a lower bound on the optimum of our original problem. The minimization problem can be written as

$$\min_{\mathbf{x}\geq\mathbf{0}} \left[\mathbf{c}^T\mathbf{x} + \mathbf{p}^T(\mathbf{b} - \mathbf{Ax})\right] = \min_{\mathbf{x}\geq\mathbf{0}} \left[\mathbf{c}^T\mathbf{x} + \mathbf{p}^T\mathbf{b} - \mathbf{p}^T\mathbf{Ax}\right] \tag{3.7}$$

and since $\mathbf{p}^T\mathbf{b}$ is not depending on $\mathbf{x}$

$$= \mathbf{p}^T\mathbf{b} + \min_{\mathbf{x}\geq\mathbf{0}} \left[\left(\mathbf{c}^T - \mathbf{p}^T\mathbf{A}\right)\mathbf{x}\right]. \tag{3.8}$$

Clearly, we can distinguish two cases

$$\min_{\mathbf{x}\geq 0} \left(\mathbf{c}^T - \mathbf{p}^T\mathbf{A}\right)\mathbf{x} = \begin{cases} 0, & \text{for } \mathbf{c}^T \geq \mathbf{p}^T\mathbf{A}, \\ -\infty, & \text{otherwise.} \end{cases} \tag{3.9}$$

Recalling Equation (3.6), we can formulate an optimization problem of finding the strongest lower bound on our primal problem. Specifically, we want to maximize the value of (3.8) by adjusting the *price vector* $\mathbf{p}$. Consequently, we only have to consider those cases of (3.9), where our minimum value is *not* $-\infty$, gaining the following LP

$$\max \mathbf{p}^T\mathbf{b} \qquad \text{subject to} \tag{3.10a}$$
$$\mathbf{p}^T\mathbf{A} \leq \mathbf{c}^T, \tag{3.10b}$$

which we call the *dual problem* with *dual variables* $\mathbf{p}$. Following the same steps for constraints of the form $\mathbf{Ax} \geq \mathbf{b}$ reveals the following scheme for construction of the dual system [BT97, p. 142]:

$$
\begin{array}{rlrl}
\tilde{\mathbf{a}}_i^T\mathbf{x} \geq b_i, & i \in M_1, & p_i \geq 0, & i \in M_1,\\
\mathbf{a}_i^T\mathbf{x} \leq b_i, & i \in M_2, & p_i \leq 0, & i \in M_2,\\
\mathbf{a}_i^T\mathbf{x} = b_i, & i \in M_3, & p_i \text{ unbounded }, & i \in M_3,\\
x_j \geq 0, & j \in N_1, & \mathbf{p}^T\mathbf{A}_j \leq c_j, & j \in N_1\\
x_j \leq 0, & j \in N_2, & \mathbf{p}^T\mathbf{A}_j \geq c_j, & j \in N_2\\
\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x}, & & \max_{\mathbf{p}} \mathbf{p}^T\mathbf{b}. &
\end{array}
$$

The central result of the duality theory is the *duality theorem*, which today is usually split up[4] in a weak and a strong form[5].

Theorem 3.1 (Weak Duality) *If* $\mathbf{x}$ *is a feasible solution to the primal problem and* $\mathbf{p}$ *is a feasible solution to the dual problem, then*

$$\mathbf{p}^T\mathbf{b} \leq \mathbf{c}^T\mathbf{x}.$$

Theorem 3.2 (Strong Duality) *If a linear programming problem has an optimal solution, so does its dual, and the respective optimal costs are equal.*

3.1.3 Solving Linear Programs

The Simplex Algorithm

More than 50 years after its invention by G. B. Dantzig the Simplex Algorithm is still the de-facto standard for solving LPs. Consequently, a plethora of introductory texts on this method exist, for example Dantzig's classic book on Linear Programming [Dan63]. More recent texts such as Chvátal's introduction in [Chv83] (which dispenses the use of Simplex tableaus) and [BT97] offer additional insight into implementation issues. Since we will not implement our own LP-solver, we will resort to a brief presentation of the basic Simplex algorithm.

Initialization We start with the standard form, shown in equations (3.4) and will restrict ourselves for the sake of simplicity to $\mathbf{b} \geq 0$. With this restriction[6], we can

[4] One of the reasons for this split might lie in the fact, that only weak duality holds for Integer Programs.

[5] Gale, Kuhn and Tucker formulate an explicit duality theorem and prove it with Farkas' Lemma [Far02] in [GKT51].

[6] Otherwise we have to solve another LP first, namely the so-called *auxilliary problem*

$$
\begin{aligned}
\min \mathbf{1}^T\mathbf{s}\\
\mathbf{A}\mathbf{x} + \mathbf{s} = \mathbf{b}\\
\mathbf{x} \geq 0\\
\mathbf{s} \geq 0
\end{aligned}
$$

easily construct a feasible solution by setting $\tilde{\mathbf{x}} = \mathbf{0}, \mathbf{x} \in R^m$ and adjusting the slack variables consequently to

$$\mathbf{s} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}} \tag{3.11a}$$
$$z = \mathbf{c}^T\tilde{\mathbf{x}}. \tag{3.11b}$$

Following [Str72] and [Chv83], we call this representation a *feasible dictionary*, which in our first feasible solution is of course equivalent to

$$\mathbf{s}' = \mathbf{b} \tag{3.12a}$$
$$z' = 0. \tag{3.12b}$$

Iteration The basic principle of an iteration in the Simplex Algorithm is "successive improvements" [Chv83, p. 14]. In the dictionary resulting from the previous step (either initialization or iteration), we now search for a *non-basic*, i. e. zero-valued variable x_j, which would improve the current solution. In order to evaluate, whether a variable would improve the solution, we calculate the *reduced cost* $\overline{c}_j$ of a variable x_j. Given a solution with an associated basis matrix $\mathbf{B}$ and the corresponding cost vector $\mathbf{c}_B$ for the basis variable, we can calculate the reduced cost of x_j via

$$\overline{c}_j = c_j - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{A}_j. \tag{3.13}$$

It is important to keep in mind that this criterion does not provide a unique *pivot column*, because there might be more than one variable with negative reduced cost. A number of strategies to select the pivot column exist, the more obvious being:

- Largest Cost Reduction [Dan63]
  This approach selects the variable with the largest impact to enter the solution basis. While this may seem to be the most natural pivoting method (although the resulting *cost change* is different from the reduced cost), it might be quite inefficient for large LPs, due to the number of operations involved in the matrix inversion and multiplications.
- Smallest Subscript [Bla77]
  As the name suggests, the first non-basic variable x_j with negative reduced costs will be used. Consequently no further calculations are necessary, once a suitable variable has been found.

For an in-depth survey of pivoting rules, we refer to [TZ93].
Now that we have found the *entering variable*, we search for the *leaving variable* by first computing $\mathbf{u} = \mathbf{B}^{-1}\mathbf{A}_j$, where $\mathbf{A}_j$ is the column of $\mathbf{A}$ corresponding to the entering variable. If no element of $\mathbf{u}$ is positive, the problem is *unbounded*, i. e. the optimal cost is $-\infty$. If some component u_k of $\mathbf{u}$ however is positive, we form a new basis where we replace x_k with x_j, i. e. we replace $\mathbf{A}_k$ with $\mathbf{A}_j$. Analogue to finding the pivot row,

which can be initialized similarly to our restricted problem.

we again have a degree of freedom selecting the leaving variable, and thus a number of possible strategies, however the runtime penalties in finding

$$\xi' = \min_{i=1 \ldots m | u_i > 0} \frac{x_{B(i)}}{u_i} \tag{3.14}$$

is less than for finding the pivot column with minimum reduced cost. In a geometrical interpretation, the constraints of our LP define a polytope and the Simplex Algorithm "walks" from one extreme point to another while improving the cost of the solution with every step.

Theorem 3.3 (Optimality) *A basic feasible solution* $\mathbf{x}$ *with an associated basis matrix* $\mathbf{B}$ *and a vector* $\overline{\mathbf{c}}$ *with the corresponding reduced costs is* optimal *if* $\overline{\mathbf{c}} \geq \mathbf{0}$[7].

Termination Using Theorem 3.3, we will stop the algorithm, when $\overline{\mathbf{c}} \geq \mathbf{0}$. If our solution basis still contains artificial variables, [BT97, p. 112] provides a method to "drive them out of the basis". As shown by Marshall and Suurballe in [MS69], it is also possible, that the simplex algorithm *cycles*, which is avoided by some pivoting rules (for example the smallest subscript rule).

Performance of the Simplex Algorithm

It is important to keep in mind, that our explanation of the Simplex algorithm did not consider any implementation issues. Especially the construction and the following inversion of the new basis matrix can be sped up considerably by using elementary theorems from linear algebra [BT97, p. 95 ff], which leads to the *Revised Simplex Method*[8].

Performance analysis of the simplex method has received a lot of attention because of an interesting discrepancy of its *worst-case* performance and its performance in real life. Beginning with a famous example by Klee and Minty [KM72], for most deterministic pivot rules a scenario has been found where an exponential number of steps is necessary to find an optimal solution.

A related problem is the *Hirsch conjecture* [Dan63, p. 168]

> In a convex region in $n - m$ dimensional space defined by n half-planes, is m an upper bound for the minimum-length chain of adjacent vertices joining two given vertices?

It is worth noting however, that a proof of the Hirsch conjecture would not necessarily include an appropriate pivoting rule.

[7] For a proof, we refer to [Dan63, p. 95] and [BT97, p. 86].

[8] Which has been introduced by Dantzig in [Dan63, p. 210] as "the simplex method using multipliers".

Despite the bad worst-case behavior of the algorithm, Dantzig already reports that the algorithm terminiates in "usually less than $\frac{3m}{2}$ steps" [Dan63, p. 160]. Later research started by Borgwardt [Bor77, Bor87] and Smale [Sma83] tries to find various measures for the average performance under probabilistic assumptions of the optimization problem in question. Recent work by Spielman and Teng [ST04] tries to fill the remaining gap by introducing a new kind of complexity analysis named *smoothed analysis of algorithms* and shows that the simplex algorithms has polynomial *smoothed complexity*.

Alternative Algorithms

In 1979, Khachian could prove [Kha79] that an algorithm developed in the Soviet literature by Shor [Sho72] and Yudin and Nemirovskii [YN77] has polynomial time complexity[9]. However, these so-called *ellipsoid methods*, originating from the methods of centers of gravity by Levin [Lev65], did not lead to efficient algorithms for real-life applications, i.e. although their worst-case behavior is favorable, the simplex algorithm performs sufficiently well on *average*.

Nevertheless, the proof that LPs could be solved also theoretically efficently, initiated further research which lead to *interior point methods*. As the name suggests, interior point methods try to find an optimal solution while "moving within the feasible set" [BT97, p. 394], in contrast to the Simplex Algorithm which walks along the boundary of the feasible set. While these algorithms perform rather well, it is non-trivial to "warmstart" these algorithms [Mit96, KRR91], which is necessary for Integer Programs (see Section 3.2), hence they are not the default choice in widely used general-purpose LP-solvers such as CPLEX. An overview of a number of interior point methods can be found in [BT97, p. 393ff].

Column Generation

In the following chapters, it will become obvious, that the sheer size of LPs arising from realistic problems is a challenge in itself. The number of steps in the Simplex method to solve LPs is both depending on the number of variables and constraints. *Delayed Column Generation* or simply *Column Generation* is a technique, where we will exploit the fact, that in many large-scale optimization problems, a huge number of variables exist, however very few of them will be part of the optimal solution, i. e. non-zero. Obviously, we do not have to include variables in our LP, which will not be in the solution base, in order to gain the optimal solution. Determination of the "necessary" variables beforehand, on the other hand is not possible either. Column generation uses an iterative approach of adding improving variables to a feasible solution until no further improvements are possible.

[9] In an interesting side-note, the popular press misinterpreted this theoretical result "in a rather ludicrous way" by declaring it to be the "Soviet Answer to the 'Travelling Salesman'" and "retreated from their earlier excesses by invoking cold-war suspicions" [Tod02].

The first important step towards column generation is the revised simplex method, which needs only the current basis and the variable entering the LP to construct the new solution base, i. e. we can ommit all variables not part of our solution. Secondly, column generation relies on an efficient method to calculate the reduced costs $\bar{c}_j$ corresponding to a variable x_j which is not part of the current solution base. Finding a variable with negative reduced costs or with the most negative reduced costs is commonly referred to as the *pricing problem*. As we will show in the following chapters, sometimes it is possible to find variables with negative reduced costs using problem dependent algorithms which exploit special structures of the underlying problem.

Interestingly enough, the first algorithm related to column generation appeared in a paper on the maximum multi-commodity flow problem [FF58] which in turn inspired Dantzing and Wolfe to their famous decomposition principle [DW60] in which column generation appeared implicitly. The first explicit application of column generation was in cutting-stock problems [GG63a, GG63b].

Summarizing a column generation algorithm starts with a feasible solution of the original problem, but just with a subset of all existing variables – the so-called Restricted Master Problem (RMP). We then try to find variables x_j with negative reduced costs and add them after resolving our problem including the new variables. Once we cannot find, any further improving variables, we will have gained the optimal solution.

Cutting Planes

As we have seen in Section 3.1.2, we introduce a dual variable x for every constraint in the primal and vice versa. Consequently, the introduction of a new variable in one iteration in column generation in the primal system generates a corresponding constraint in the dual system. Thus, searching for improving variables with negative reduced cost $\bar{c}_j$ is equivalent to finding violated constraints (which are not part of the RMP) in the dual – a fact we will make use of in a number of planning problems in the following chapters.

3.1.4 Computational Complexity

Considering the fact, that interior point methods with polynomial complexity exist, an additional section on computational complexity may be superflous at a first glance. At a closer look however, polynomial complexity for the solution methods are related to a given LP-formulation How the LP-formulation itself behaves is not included in this measure. On top of that, we will later see that formulations with a huge number of variables can be solved quite efficiently provided that we find an efficient method to implicitly consider these variables in our solution base, i. e. use column generation.

In a ground-breaking[10] article by Grötschel, Lovász and Schrijver [GLS81, GLS88], the authors show that although an LP may have an exponential size regarding the original

[10]The authors received the John von Neumann Theory Prize in 2006 for this work.

problem, this does not automatically imply that it cannot be solved efficiently. Their *separation theorem* states that as long as *separation* can be performed in polynomial time, the whole problem can be solved in polynomial time. A proof in the other direction is more difficult and cannot be shown in general: Assume that we have shown a pricing problem to be $\mathcal{NP}$-hard for a non-compact[11] formulation. Usually, this proof will not show that no compact formulation or another extended formulation with polynomial pricing can be found. For an overview of complexity issues of network optimization problems relevant to this thesis, we refer to a survey by Orlowski and Pióro [OP08] and an article by Tomaszewski, Pióro, and Żotkiewicz [TPŻ09].

3.2 (Mixed) Integer Programming

Up to now all our variables x were continuous, i. e. elements of $\mathbb{R}$. In many problems in real life however, the goods represented by a variable cannot take fractional values. For example, one can only use a complete wavelength on a fiber (and not use half of the wavelength and spend only money for half of the equipment necessary) or one can hire only one person and not half a person, etc.
In other words, we would like some variables to take only discrete values. This turns out to be a mathematically challenging problem – in general IPs are $\mathcal{NP}$-hard. Perhaps the most intuitive solution would be to simply round unwanted fractional values, however this method is not guaranteed to gain optimal or even feasible solutions. The first research towards an algorithms solving IPs was motivated by exactly this problem in the US Navy, where planners “complained about LP solutions prescribing fractional numbers of, e. g. aircraft carriers” [Joh05]. This resulted in Gomory’s cutting algorithm [Gom58].

3.2.1 Standard Form

For the sake of simplicity, we will start with pure IPs of the form

$$\min \mathbf{c}^T\mathbf{x} = z \qquad \text{subject to} \tag{3.15a}$$
$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.15b}$$
$$\mathbf{x} \geq 0 \tag{3.15c}$$
$$\mathbf{x} \in \mathbb{N}^m. \tag{3.15d}$$

where all variables x_j can only take discrete values and we will call the corresponding program without integer constraints, i. e. $\mathbf{x} \in \mathbb{R}^m$, the *linear relaxation*. As it will be quite obvious how to use the following solution methods in programs consisting of both, discrete and contiuous variables, so-called MIPs, we will not mention them explicitly. Since the relaxation allows for more solution possibilities (however includes all integer solutions), we can deduce that the value of the optimum for the relaxation

[11]Growing exponentially with respect to the original problem.

is not larger than for the IP. Hence, if we solve the relaxation and obtain an integer solution we will have found the optimum already.

3.2.2 Solving MIPs

Cutting Plane Algorithms

The general idea of a cutting plane algorithm for IP is similar to the one for LPs: We try to find constraints, which are fulfilled by every integer solution, but not by our current optimum. The first systematic and finitely terminating algorithm was the previously mentioned cutting plane algorithm by Gomory [Gom58], which in practice however "has not been particularly successful" [BT97, p. 482] as a stand-alone algorithm, but has proven to be very useful in Branch-and-Bound algorithms.

Branch-and-Bound

Branch-and-Bound or *divide and conquer* (a more common name in computer-science) was first proposed by Land and Doig to solve IPs [LD60] and gained considerable attention after Little et al. applied it successfully to the travelling salesman problem in [LMSK63].

The basic idea of branch-and-bound is to recursively split up the solution space of all feasible integer solutions into smaller and smaller subsets which are easier to solve. This process can be visualized as a tree, where every node represents a subset of the solution space. While this strategy alone would lead to total enumeration of the solution space (which is only a viable approach for rather small problems), we are able to reduce the number of subsets we have to evaluate in real-life problems considerably by *bounding*. For example, the value of a known solution for the problem can be used as a bound: If we are sure, that all nodes below a certain node will have worse solutions than the node we are currently examining, we will not need to evaluate the nodes below our current node.

In the context of IPs, the linear relaxation is a way to obtain such bounds: If the solution of a linear relaxation in a given node is worse than a known integer solution, we know that any integer solution obtained below this node will be worse as well, i. e. we do not have to search the tree below this node. This branch-and-bound approach with linear relaxtions is the base algorithm of most commercially available (M)IP-solvers to the author's best knowledge.

Initialization In the first step, we will solve the linear relaxation of our IP. Since all branching starts from this point, this stage is the *root node* of our *branch-and-bound tree*. As pointed out above, the value of this optimum is a guaranteed lower bound for our IP. The difference between the lower bound and the best known integer solution is called the *(optimization) gap*.

Branching *Branching* is the process of dividing the solution space. A number of branching schemes have been proposed in the literature, however for the sake of simplicity we will present only the most widely used and generally applicable method – *variable dichotomy* introduced by Dakin [Dak65] – and present further problem-specific branching schemes (which we will need for branch-and-price) where they are necessary[12].

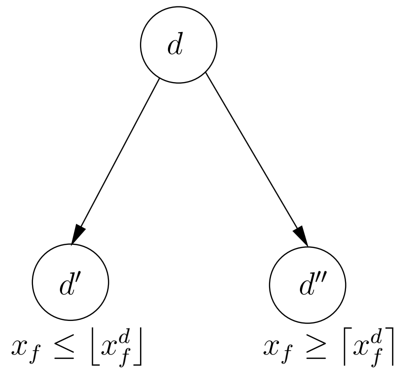


Figure 3.1: Creating a Branch

In the current node d of the branch-and-bound tree as depicted in Figure 3.1, we select a variable x_f with a fractional value x_f^d in the solution of the linear relaxation and create one node d' with a new constraint $x_f \leq \lfloor x_f^d \rfloor$ and another node d'' with $x_f \geq \lceil x_f^d \rceil$ respectively, below d. Each of these constraints renders our current solution $\mathbf{x}_f^d$ infeasible in the relaxation by adding a single constraint in every node, which means that we can reoptimize the problem by means of the dual simplex algorithm (warm-) starting from $\mathbf{x}_f^d$ [BT97, p. 487]. Quite obviously, there will be cases where more than one variable has a fractional value and we need an additional criterion for variable selection. A number of rules are commonly used, but the "right" choice depends very much on the goals the user is aiming at (a fast integer solution, tight bounds, etc.) and can greatly influence the performance of the branch and bound algorithm. Nemhauser and Wolsey [NW88] and Wolsey [Wol98] discuss some of the more obvious choices.

Node Selection After adding the new nodes to the branch-and-bound tree, we need to decide which of the nodes to examine next. All possible strategies lie somewhere between a *depth-first* and a *breadth-first* search. While a depth-first search is likely to gain a feasible integer solution quickly, a breadth-first search may provide better lower bounds on the problem. Consequently, it may be worthwile to vary the search strategies during the solution process [Wol98]: Since a feasible solution will gain an upper bound, this is usually the first priority. After a feasible integer solution has been found, it is quite useful to switch to a *best-node first* strategy, i. e. we examine the node with the best available bounds first. Once again, the "best" search method is very much dependent on our aims.

[12] A computational study of various branching schemes can be found in [LS99].

Node Pruning As pointed out in the beginning, the strength of branch-and-bound algorithms lies in the fact, that we *avoid* to enumerate the whole solution space. After solving the linear relaxation in a node d, we might find several reasons to *prune* d and thus consequently all of its child nodes [NW88, p. 356]:

- The LP relaxation is infeasible. Since further branching can only further constrain the solution space, we can be sure not to find a feasible solution further down the branch-and-bound tree. Hence we can prune d.
- The LP relaxation gains an integer solution, which provides a lower bound for the branching decisions taken up to node d. Again, further branching cannot yield a better solution, thus all child nodes can be pruned.
- The value of the LP relaxation in d is larger than an upper bound we found earlier in the tree. As discussed previously, we will not be able to improve the LP relaxation by further restrictions, consequently we can prune d and all of its child nodes.

Termination The algorithm will terminate, if no nodes are left to visit, or the optimization gap is zero.

It is important to notice, that the performance of our branch-and-bound algorithm is heavily influenced by the availability of strong bounds. Due to this observation, a lot of effort has been put forth to find good bounds. Many MIP frameworks provide local search heuristics, which will try to improve solutions found in a given node and thereby strengthen the upper bound of the problem. Dual algorithms generate valid lower bounds which can also greatly reduce the solution space. Nevertheless, some factors can still severely impact the speed with which we can solve problems to optimality. Besides the sheer size of a problem, many formulations we will investigate later-on are highly *symmetric*, i. e. a large number of solutions with equal cost exist. Although research has started to use these symmetries (e. g. [Mar03, KPP06]), they are not yet incorporated in commodity solvers.

Branch-and-Cut

A well-known technique to improve the bounds in our branch-and-bound tree is to integrate cutting-planes. Just as in the LP-case, a number of problem-specific (i. e. exploiting aspects of the underlying structure) and generally applicable methods are known and used. For a general introduction to branch-and-cut algorithms, we refer to [PR91] and to [KOR$^+$06] for an overview of network-related cuts.

Branch-and-Price

Branch-and-Price which is also known as *IP Column Generation* is a similar approach as column generation just applied to integer variables. Combining column generation with branch-and-bound however is quite challenging, as was already discovered by

Appelgren [App69] in 1969, which explains why the first branch-and-price algorithm known to the author appeared 22 years later [NP91]. Recalling the basic principle of column generation as described in 3.1.3 and keeping in mind that branching corresponds to adding constraints to our problem, reveals the reason: Every branching step changes our pricing problem! Consider for example a variable, that has been branched out, e. g. its value was fixed to zero. If the pricing method cannot take this into account, it might add another variable, which is identical to the one just branched out. A good introduction and early overview of branch-and-price is [Van94].

3.3 Network Optimization

In this section, we will build the foundation of all network optimization problems we will encounter in the following three chapters. Building upon graph-theory, we introduce the notational conventions and some basic formulations which we will heavily use in the remainder of this thesis.

3.3.1 Graph Theory

Terminology

A graph $\mathcal{G}$ consists of a set of *nodes* or *vertices* $\mathbb{V}$ and a set of relations $\mathbb{E}$ between these nodes. In case these relations are unidirectional, for example from node v_1 to node v_2, we call such a relation an *arc*. If they are bidirectional, we will call this relation an *edge*. In case a graph consists solely of directed arcs, we call the graph *directed* (or shorter a *Digraph*) and in the opposite case *undirected*.

A *path* p is a chain of edges or arcs

$$p = \{e_1, \ldots, e_n\} . \tag{3.16}$$

Obviously, in case of arcs all arcs have to be directed in the same direction. *Source* or *origin* and *target* or *destination* of a path are self-explanatory terms for the respective end-nodes of a path.

We will call a graph *connected*, *iff* a path exists for every node-tuple $\{v_1, v_2\} \in \mathbb{V}$. In Section 6.2, we will use the *path probability* Π as a softer criterion to measure the connectivity of a graph:

$$\Pi = \frac{\text{number of connected node pairs}}{\text{number of nodepairs}}. \tag{3.17}$$

A considerable amount of literature cited in the same section makes heavy use of random and random geometric graphs and related theorems. In a *random graph* following the standard Erdős-Rényi model [ER59], an edge connecting two nodes v_1 and v_2 exists independent of other edges or nodes with a probability p. In a *random geometric* *graph* the nodes are uniformly distributed on a bouned region and two nodes v_1 and v_2 will be connected (i. e. an edge exists between them) *iff* they are not further apart than a given threshold.

Graph Algorithms

Shortest Path Algorithms The shortest path problem and its solution by Dijkstra [Dij59] with polynomial complexity can be considered to be one of the classic graph-theoretical problems. Briefly stated, given a graph $\mathcal{G}(\mathbb{V}, \mathbb{E})$ and associated positive edge-weights $w(e)$, we look for the shortest path from a given source node v_s to a target node v_t in respect to the edge-weights.

Naturally Dijkstra's result has been improved in the last 40 years. Classic works in this area are [Joh77, AMOT90] and the most efficient algorithm known to the author is presented in [Pet04], with a complexity of $O\left(|\mathbb{E}| \cdot |\mathbb{V}| + |\mathbb{V}|^2 \log\log |\mathbb{V}|\right)$. Meyer examines the average-case complexity of Shortest-Path Algorithms in [Mey03] and introduces a label-setting algorithm running in linear time on the average with high probability.

Numerous specializations imposing different constraints on the edge-weights exist. More important for our field of research however is a generalization allowing negative edge-weights (as long as no negative cycles exist) dubbed the *modified Dijkstra's algorithm* presented in [Bha99, pp. 29].

k-Disjoint Shortest Paths A related problem to the shortest path problem is the problem of finding shortest node- or edge-disjoint[13] path tuples. That is, path tuples which do not use the same edges or nodes and are again the shortest in respect to the edge-weights. Depending on the number k of paths in this path tuple, we speak of *k-disjointness*.

The two most common approaches are Suurballes's algorithm [ST84] and Bhandari's algorithm [Bha94, Bha99]. The GRAPH library [Lay09] provides an implementation of the latter. While none of the algorithms can claim to be more efficient [Bha99, p. 92], Bhandari's algorithm seems to be easier to implement and can reuse existing algorithms (most notably the modified Dijkstra's algorithm).

3.3.2 Basic Formulations

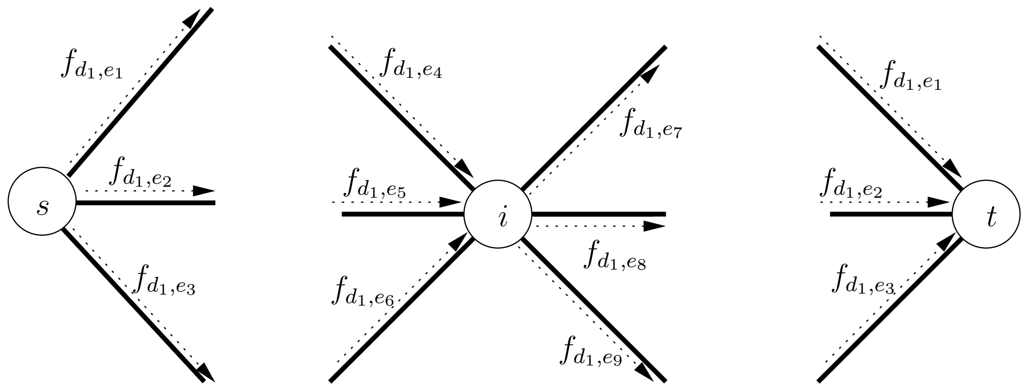
In this section, we will present two basic formulations for the so-called *Multi-Commodity Flow (MCF)*-problem. In its simplest form the problem can be described as follows: We want to send a set of *commodities* or *demands* $d \in \mathbb{D}$ through a network. We can interpret such a demand as a request for transmission capacity D_d from a source v_s to a destination node v_t. In our cases, the links in our network are capacitated, bearing the consequence, that the problem in general cannot be solved via shortest-path algorithms used consecutively for demand after demand.

[13] Obviously a node disjoint path tuple is always edge disjoint.

While this problem may seem natural for communication networks, it is by no means exclusively used in this setting. As a matter of fact, they originated in *transportation problems* which were among the first problems to be solved with LPs.

Flow Approach

The underlying principle of this approach is the law of *flow conservation*, well-known among electrical engineers through Kirchhoff's Current Law. For a *commodity* or *demand* d_1 for which we want to send an amount of traffic D_{d_1} from a source s to a sink t, we define a set of flow variables $f_{d_1,e}$ for every arc e in the network. These flow variables will represent the amount of the commodity going over edge e. Applying the rule of flow conservation, we can differentiate between the three cases depicted in Figure 3.2.



(a) Source (b) Intermediate (c) Target

Figure 3.2: Flow Approach

- For all arcs $e \in \mathbb{E}_s$ starting at the source node s as illustrated in Figure 3.2a, the sum of the corresponding flow variables has to be equal to the demand D_{d_1}

$$\sum_{e \in \mathbb{E}_s} f_{d_1,e} = D_{d_1}. \tag{3.18a}$$

- For an intermediate node i (e. g. neither source nor sink) no flow must terminate or originate at the node. Thus the sum of all flow variables of arcs $\mathbb{E}_i$ starting at i (which we count positively) or arcs $\mathbb{E}^i$ ending at i (which we count negatively) has to be zero.

$$\sum_{e \in \mathbb{E}_i} f_{d_1,e} - \sum_{e \in \mathbb{E}^i} f_{d_1,e} = 0 \tag{3.18b}$$

- Finally, for the target node t, the sum of all flow variables for incoming arcs $\mathbb{E}^t$ as shown in Figure 3.2c, which we count negatively for the sake of consistency, has to be $-D_{d_1}$

$$-\sum_{e \in \mathbb{E}^t} f_{d_1,e} = -D_{d_1}. \tag{3.18c}$$

According to [PM04, p. 110], the number of constraints and variables generated is proportional to $|\mathbb{V}|^3$. A modified version, which we will apply in Section 6.2, exists with complexity proportional to $|\mathbb{V}|^2$. Furthermore, it is worth mentioning that the structure of the resulting polyhedron can be as important as the mere number of variables and constraints. As will be shown later, the flow approach can show both behaviours clear advantages in the total runnning time of an optimization as well as being clearly slower than a path approach with column generation.

Path Approach

While the flow approach seems to be a very natural way of formulating network optimization problems, it has one disadvantage besides the above mentioned performance issues: It is very hard to impose limitations on the paths which demands are allowed to take. Gruber showed in [Gru07] how many of the required restrictions can be formulated in flow approaches, but the performance hits due to the increased number of necessary (integer) variables are even more severe. Thus the path approach

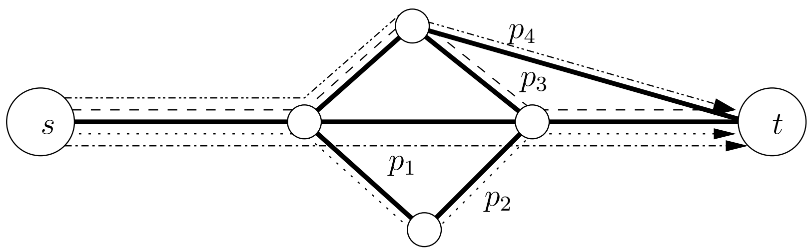


Figure 3.3: Path Approach

follows a completely different philosophy: Instead of viewing the flows on a per-node basis, we create flow variables for every possible path, along which traffic might travel. Hence, for a commodity from s to t as shown in Figure 3.3 and a set $\mathbb{P}_{d_1}$ of four possible paths $p_1, \ldots, p_4$, we create four flow variables $f_{d_1,p_1}, \ldots, f_{d_1,p_4}$ representing the traffic flowing along the four paths. Consequently the sum of the traffic flows via these paths $p \in \mathbb{P}_{d_1}$ has to be at least as large as the demand D_{d_1}

$$\sum_{p \in \mathbb{P}_{d_1}} f_{d_1,p} = D_{d_1}. \tag{3.19}$$

Quite obviously, the set of eligible paths ($\mathbb{P}_{d_1}$ in our case) has to be precomputed before the actual optimization can take place. While this might seem tedious in the beginning, it offers great flexibility as we can easily influence the composition of this path-set, e. g. restrict path-lengths, leave some paths out, etc. Pióro et al. show in [PM04, p. 110], that the number of constraints and variables are proportional to $|\mathbb{V}|^2$.

Summarizing, both approaches have their strength and weaknesses and it is application dependent, which approach leads to the desired results. Thus, we will use

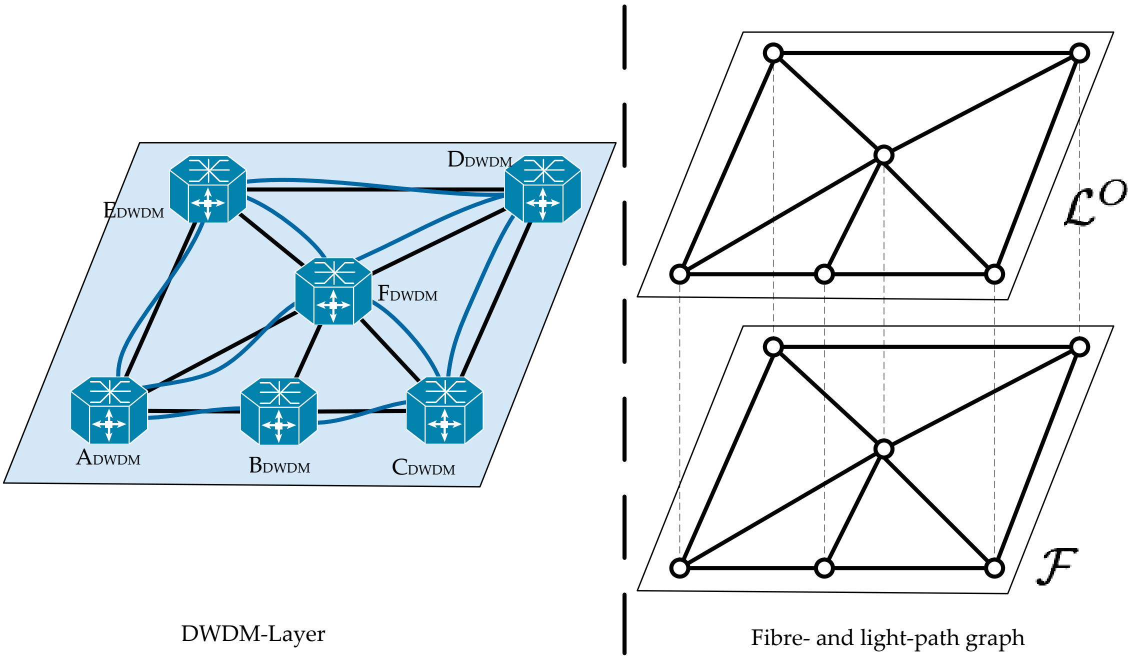


Figure 3.4: DWDM-layer and network model

both approaches in the following chapters and in some multi-layer problems even a combination of them.

3.3.3 Basic Network Models

Having defined basic terminology, we will now develop our basic network model for the basic IP/MPLS over DWDM network architecture presented in Section 2.1.2.

Opaque Optical Networks

As explained in Section 2.1.2, a direct optical link can only be established between two neighbouring nodes. Consequently the possible topology of our optical network follows the topology of our fibre network strictly. Our fibre network from now on will be modelled as an undirected graph $\mathcal{F}(\mathbb{V}, \mathbb{F})$, hence every fibre corresponds to exactly one *fibre edge* $e \in \mathbb{F}$ and every node location to one node $v \in \mathbb{V}$.

On top of this fibre graph, we construct a (in the opaque case trivial) light-path graph $\mathcal{L}^O(\mathbb{V}, \mathbb{L}^O)$, which is exactly identical to our fibre graph, because every possible "light-path" has to follow exactly one edge in the fibre graph.

In order to use an edge in $\mathcal{L}^O$, we have to install a transponder t from the set of eligible transponders $\mathbb{T}_l$ for l. Every transponder t offers a well-defined amount of transportation capacity or *bandwidth* C_t to an upper layer. Now assume that we know that we need the bandwidth u_l on edge $l \in \mathbb{L}^O$. If $n_{l,t}$ is the number of transponder-pairs t

installed at the end nodes of link l, then obviously the following constraint must be fulfilled:

$$\sum_{t \in \mathbb{T}_l} n_{l,t} \cdot C_t \geq u_l, \qquad \forall l \in \mathbb{L}^O. \tag{3.20}$$

Every one of the transponder-pairs consumes exactly one wavelength in our DWDM-system which is usually limited to a given number of W_e wavelengths per fibre e. As a consequence, we can only install W_e tranponders on the corresponding link l in an opaque network, which we denote by

$$\sum_{\substack{l \in \mathbb{L}^O: \\ e \in l}} \sum_{t \in \mathbb{T}_l} n_{l,t} \leq W_e \qquad \forall e \in \mathbb{F}. \tag{3.21}$$

Although this formulation may seem overly complicated for an opaque network, an extension to transparent and translucent networks is relatively simple and straightforward, as we will demonstrate in the next section.

Since transponders do cost a considerable amount of money (in simple optical nodes, they can be the main cost-driver), we are usually interested in minimizing the costs caused by these transponders. As their prices are not directly proportional to the bandwidth they offer, we have to consider the actual transponder costs, which we denote by K_t and not just the mere number of transponders. Nevertheless, we can easily calculate the total costs K caused by the transponders via

$$K = \sum_{l \in \mathbb{L}^O} \sum_{t \in \mathbb{T}_l} n_{l,t} \cdot K_t. \tag{3.22}$$

Transparent and Translucent Optical Networks

In a transparent or translucent network, the light-paths become true paths, i. e. are no longer restricted to a single hop. Obviously, the fibre graph $\mathcal{F}$ can remain unchanged, while the light-path graph gains importance. Every edge $l \in \mathbb{L}$ now corresponds to a series of fibres, namely the way the light-path transparently takes. In a mathematically rigorous manner, we would need to define a separate set containing the fibre edges of l. We will be sloppy in this case and use l for both, the edge in the light-path graph and the path on $\mathcal{F}$, which shortens the amount of notations we need below our summation symbols considerably. With this notation, no changes in the transponder and light-path constraints and cost equation are necessary, we just have to change the set of light-paths from $\mathbb{L}^O$ to $\mathbb{L}$:

$$\sum_{t \in \mathbb{T}_l} n_{l,t} \cdot C_t \geq u_l, \qquad \forall l \in \mathbb{L} \tag{3.23}$$

$$\sum_{\substack{l \in \mathbb{L}: \\ e \in l}} \sum_{t \in \mathbb{T}_l} n_{l,t} \leq W_e \qquad \forall e \in \mathbb{F} \tag{3.24}$$

$$K = \sum_{l \in \mathbb{L}} \sum_{t \in \mathbb{T}_l} n_{l,t} \cdot K_t \tag{3.25}$$

The only difference lies in the set of light-paths, which contains only one hop "paths" in the opaque case $\mathcal{L}^O\left(\mathbb{V}, \mathbb{L}^O\right)$ and all possible[14] paths in the transparent and translucent case $\mathcal{L}\left(\mathbb{V}, \mathbb{L}\right)$.

3.4 Summary

In this chapter, we have presented an overview of the used mathematical methodologies with one considerable limitation: We solely focussed on optimal methods. As we will demonstrate in the first section of the following chapter, *heuristic methods* have one common weakness, namely the lack of information about the solution quality. More verbosely, with an optimal method we will always get a solution gap, which tells us how far we are away from a theorically possible optimum. With a heuristic method we do not gather any comparable information and blind trust into one heuristic method or another was shown to be questionable at best by Wolpert and Macready in [WM97].

[14] Physical and political limitations to simplify network operation are quite likely.

4 Capacity Planning of IP/DWDM Transport Networks

"We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil. Yet we should not pass up our opportunities in that critical 3%."[1]

Building on the general architecture presented in 2.1.1 we will focus on planning problems in the *optical domain* in this chapter. In the first section, we will present a heuristic planning method, which is capable of finding good solutions for very complex multi-layer plannning problems within minutes. Subsequently,we will examine the influence of an upcoming transmission technology (optical OFDM) on optical backbones and compare the predicted costs to conventional optical backbones.

4.1 Fast Heuristic Planning

As we will show in the following section, modern DWDM nodes offer considerably more flexibility than the relatively static picture we drew in Section 2.1.1 suggested. In a network equipped with these modern nodes, it will be possible to set up light-paths in a matter of minutes, which is – compared to the current timeframe of a few weeks – quite an improvement. Hand in hand with this technological improvement, a new business model for ISPs becomes viable: *bandwidth on demand*. In this setting, a customer can rent relatively fast connections (in the most extreme case a complete light-path, which would mean 40 Gbit/s with current transmission systems and up to 100 Gbit/s in the near future [1]) for short periods of time and on very short notice. Quite obviously, this will largely increase the dynamics of both traffic in the backbone network and, given quickly reconfigurable network nodes, the backone itself. As a consequence, faster planning methods are necessary, since rapidly changing demands might make it impossible to wait for days for optimization results. In a nutshell we have to be able to design a cost-effective optical network configuration within a time frame of a few minutes. Naturally, this is not possible without sacrifices, which in our case is the loss of information about the solution quality. While LP-based methods

[1] Donald Knuth, "Structured Programming with go to Statements", *ACM Computing Surveys*, vol. 6, no. 4, December 1974

will always report a solution gap, heuristic methods cannot provide this piece of information, which is why we will benchmark our heuristic in a variety of settings with known optimal solutions.

The evolution of the heuristic took place in two big steps: At first, we considered a scenario which was used in earlier work by Scheffel et al. in [SPG+06]. By using the same setting, we could use the provably optimal solutions of their approach as a solid reference and thus get reliable information on the solution quality. In the next step, the multi-layer capabilities of the developed framework were fully exploited and we extended the heuristic to a network architecture where no optimal bounds were known.

We will follow this chronological sequence and introduce the first scenario and explain the basic algorithm in this scenario for the sake of clarity and demonstrate the solution quality capabilities by comparing it to known optimal solutions. Subsequently, we will present the second, far more complex scenario and devise the necessary extensions of the heuristic.

4.1.1 DWDM-Grooming

Scenario

While the planning of a DWDM link is quite complex on the physical layer[2], the pure network planning issues are well-understood and can be successfully treated as capacitated multi-commodity flow problems with LP-methods [PM04], as long as we restrict our light-path topology to an either *transparent* or *opaque* network. An opaque network consists only of point-to-point connections, i. e. within a given fibre topology our light-paths traverse only via single links, but never over multiple hops. Hence, the light-path topology is equivalent to the fiber topology and traffic flows will go through O/E/O-conversion at every intermediate node. The opposite extreme is a transparent network, where a traffic flow, will *never* be converged to the electrical domain until it reaches its destination. Consequently, seen from the light-path perspective all nodes are reachable via a single-hop. Both approaches have strengths and weaknesses:

- The number of transponders within the transparent case is a lot smaller than for the opaque case, as O/E/O-conversion only takes place at the source and sink nodes for a given demand.
- The amount of traffic, which can be sent through a transparent network can be dramatically smaller: Every light-path has to be routed on the same wavelength along its path (*wavelength continuity constraint*) and every wavelength on a single fibre can only be used one single time (*distinct wavelength constraint*). Finding

[2] We refer to [Fis07] for an in-depth survey of the physical effects as well as simulation and planning methods.

a routing configuration together with an optimal *wavelength assignment* (the so-called Routing and Wavelength Assignment (RWA) problem) is a very demanding and challenging task[3], especially due to its highly symmetric nature[4].

- While the total cost of a transparent solution (if a feasible solution can be found) may seem to be lower at a first glance, it depends very much on the link length and the required transponders which have to be used. Furthermore, it might be necessary (as no feasible solution might exist otherwise) or more cost-effective to allowt for some demands 3R regeneration at least, because *wavelength conversion* can be performed with no additional cost. Quite obviously the resulting planning problem of regenerator allocation will be even more complex especially if protection considerations come into play[5].

However, there is no reason to restrict the design of the light-path topology to one of these extremes. Many modern add-drop multiplexers support transparent light-paths, i. e. not all wavelengths on a fiber have to be dropped. These *translucent networks* offer a viable compromise, yet again increase the complexity of our planning problem, since it is far from obvious where light-paths should start and end. While the first nodes which allowed this configuration were little more than optical counterparts to a patch-panel, modern node architectures allow remote reconfiguration of add, drop and transparent channels more or less on the fly.
The act of multiplexing several client signals into one light-path is called *grooming*. Obviously transparent networks offer only *end-to-end grooming*, i. e. either a traffic flow is taking exactly the same route and thus can be groomed or not. Translucent networks additionally offer *intermediate grooming*, where a traffic flow can use a sequence of ligh-paths. Figure 4.1 shows a number of options within an example with $2.5\,$Gbit/s and $10\,$Gbit/s light-paths. As shown, client $2.5\,$Gbit/s signals from node s to t have four basic possibilities:

- Just as in a transparent network it can travel using its own exclusive wavelength (signal d_1) or
- groomed within an end-to-end light-path (signal d_2).
- In the opaque case of signal d_3, full O/E/O-conversion is performed at the intermediate node.
- Furthermore signal d_4 is groomed in a $10\,$Gbit/s light-path from node s to i and gets an individual light-path from node i to t. Quite obviously, the other way round would be possible also.

Grooming traffic however is far from being trivial: Regular transponders offer only one client interface, so grooming has to take place in one of the upper layers (which could be for example SDH/SONET, Carrier Grade Ethernet, IP/MPLS). Since espe-

[3]Zymolka presents in [Zym07] the state of the art optimization approaches for this problem.

[4]Given the stated requirements, it is obvioulsy quite irrelevant how the wavelengths are assiged as long as the assignment meets the constraints, i. e. wether light-path A uses wavelength 1 and light-path B wavelength two or vice versa, does not make a difference. Thus, if we can use N_λ wavelengths, $N_\lambda!$ identical solutions will exist.

[5]Schupke, Scheffel and Grover [SSG03] present an optimization method for networks with p-cycle protection and Scheffel [Sch05] presents an approach for path-based protection.

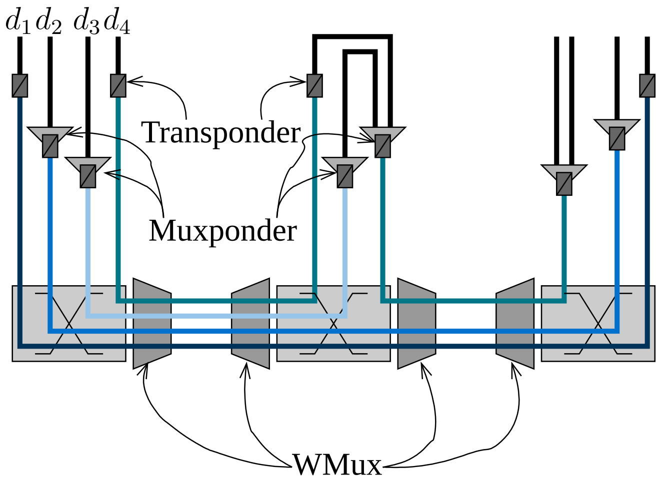


Figure 4.1: Grooming options in a translucent DWDM network

cially prices for IP linecards are rather high and considering the fact that larger routers are extraordinarily expensive as for example apparent from [HGMS08], the price for intermediate grooming can be quite high as well. In order to reduce this overhead, a new device named *muxponder* was introduced. A muxponder is a combination of an electrical multiplexer and a transponder and is capable of multiplexing lower datarate signals, namely

- four 2.5 Gbit/s signals into one 10 Gbit/s or
- four 10 Gbit/s signals into one 40 Gbit/s,

thereby reducing the necessary equipment for grooming. Our task will now be to design a cost-effective configuration of the light-paths in the network using the element previously described.

Scheffel et al. present near-optimal results[6] in [SPG+06] for this setting. In order to allow for a comparison of both approaches, we use the same cost model as in the work cited above, which was created within the NOBEL project [NOB04]:

It is important to note however, that the framework created to accomodate the heuristic algorithm described below was designed to be capable of dealing with complex multi-layer structures from the very beginning.

Algorithm

Our optimization algorithm is divided into two stages. During the first stage, we try to find a (reasonably) good initial solution. Subsequently, we will improve this initial solution in the second stage. This separation offers great flexibility: Suppose we only want to check whether a given scenario (demand matrix, fibre topology and

[6] The only limitations imposed on the solution were a limit on the number of paths eligible for transparent routes and the premature termination after six hours of running time.

Equipment	Relative Costs
2.5 Gbit/s transponder	50
10 Gbit/s transponder	160
$4 \cdot 2.5$ Gbit/s to 10 Gbit/s muxponder	185
$4 \cdot 10$ Gbit/s to 40 Gbit/s muxponder	840
WDM Mux base unit	480
WDM Mux 10 channel module	105
Optical Protection Switch	42

Table 4.1: Cost Model

usable equipment) is feasible. Under these circumstances, we can terminate after stage one. Furthermore, we can control the running-time of the second stage, with a low risk of not finding a feasible solution at all. Thus, depending on the purpose of our simulation we can give the heuristic everything from seconds to days to improve a given solution.

Related Work Since a number of problems share the basic structure of our problem, a number of relevant planning methods exists. However none of them could deliver the solution times or the flexibility required for the above scenario. Historically many grooming strategies specifically for SONET/SDH-based ring networks have been developed (an overview of which is presented in [ZZM03b]), but the unique nature of ring networks, especially when taking protection into account, makes an adoption far from trivial. An important contribution to the evaluation of the solution of our qualitiy is a method developed by Scheffel et al. in [SPG$^+$06], where they formulate a MIP in a DWDM grooming scenario, which we can directly use as a reference solution. However their solution times are in the range of six hours[7]. Further optimal approaches, which were either limited on the network size were studied in [ZM02] or solved a slightly different problem (namely RWA$^+$) in [LYTL04]. Heuristic WDM topology design (before multi-granular networks) was pioneered by Mukherjee et al. in [MBRM96]. A first comparison between heuristic and optimal algorithms was presented by Ramaswami and Sivarajan in [RS96]. Höller and Voß develop a heuristic for SDH routing in [HV06] with which they solve very large-scale problems (more than 100 nodes), but again the required solution time is too large for our scenario. Interestingly enough, their use of a Greedy Randomized Adaptive Search Procedure (GRASP)-like heuristic [FR89] seems to bring substantial improvements, which is why we have chosen to follow a similar philosophy in our improvement heuristic. Doumith and Gagnaire [DG06] studied several heuristic approaches for a model related to an earlier work by Zhu and Mukherjee in [ZZM03a], which could be adopted to muxponders and report running times of an hour and more, without

[7] A closer evaluation of their results furthermore shows, that the MIP-solver still finds substantial improvements in the solution instead of merely proving optimality. Therefore, simply restricting the solution time is not an option.

stating an optimal or lower bound of their solutions.

Data Representation Although the scenario described above consists of three granularities, we will restrict our explanation of the algorithm to two layers, namely 2.5 Gbit/s and 10 Gbit/s.

As presented in Figure 4.2, we use six graphs in total to store our multi-layer structure:

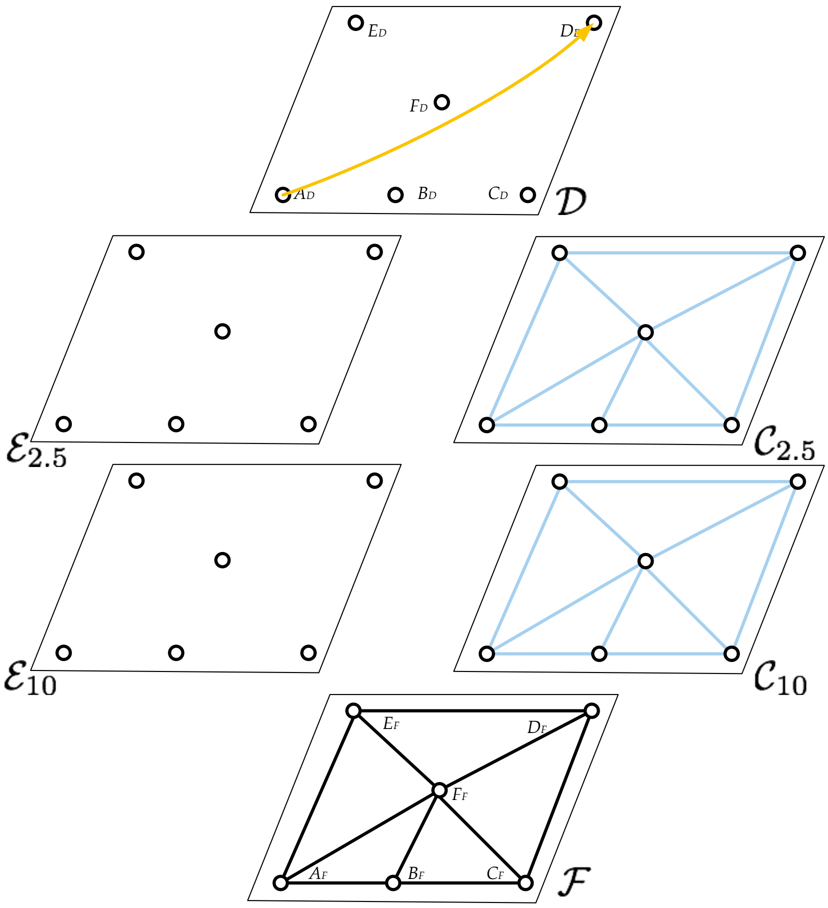


Figure 4.2: Graph representation of the DWDM multi-layer structure.

- All demands are represented in the *demand graph* $\mathcal{D}\left(\mathbb{V}_{\mathcal{D}}, \mathbb{D}\right)$ as arcs between source and targets of their respective demands. The arcs carry an attribute specifying the traffic volume of the demand and further properties such as required protection, etc.
- The underlying fibre topology (e. g. the set of eligible fibres) is stored in the *fibre graph* $\mathcal{F}\left(\mathbb{V}_{\mathcal{F}}, \mathbb{F}\right)$ at the bottom of Figure 4.2.
- The DWDM multi-layer hierarchy consisting of a 2.5 Gbit/s layer $\mathbb{L}_{2.5}$ and a 10 Gbit/s layer $\mathbb{L}_{10}$ is doubled in two sets of graphs, the *equipment* ($\mathcal{E}_{2.5}\left(\mathbb{V}_{\mathcal{E}_{2.5}}, \mathbb{E}_{\mathcal{E}_{2.5}}\right)$, $\mathcal{E}_{10}\left(\mathbb{E}_{\mathbb{V}_{\mathcal{E}_{10}}, \mathcal{E}_{10}}, \right)$)

and *candidate graphs* ($\mathcal{C}_{2.5}\left(\mathbb{V}_{\mathcal{C}_{2.5}}, \mathbb{E}_{\mathcal{C}_{2.5}}\right)$, $\mathcal{C}_{10}\left(\mathbb{V}_{\mathcal{C}_{10}}, \mathbb{E}_{\mathcal{C}_{10}}\right)$). The equipment graph contains all provisioned light-paths, while the candidate graph is a representation of the unused capacity in the network. Furthermore, the candidate graphs contain so-called *pseudo edges*, (light blue in our illustrations) which are corresponding edges to fibre edges, which have unused wavelengths, i. e. fibres which could be used by additional light-paths.

It is worth noting, that the nodes carry relations to nodes in other layers as well. For example every demand node has at least a corresponding fibre node and a correspoding node in the eligible candidate and equipment graphs. Consider for example a demand for a 2.5 Gbit/s connection from node A_D to D_D. Two basic possibilities exist to meet this request. Either the request could be realized directly as a 2.5 Gbit/s light-path (via a "regular" transponder) or as a 2.5 Gbit/s channel multiplexed into a 10 Gbit/s light-path, which would leave three 2.5 Gbit/s channels free for further use. Suppose we decide to route the demand via node C within a 10 Gbit/s light-path as shown in Figure 4.3. The demand arc has a reference on a 2.5 Gbit/s channel which

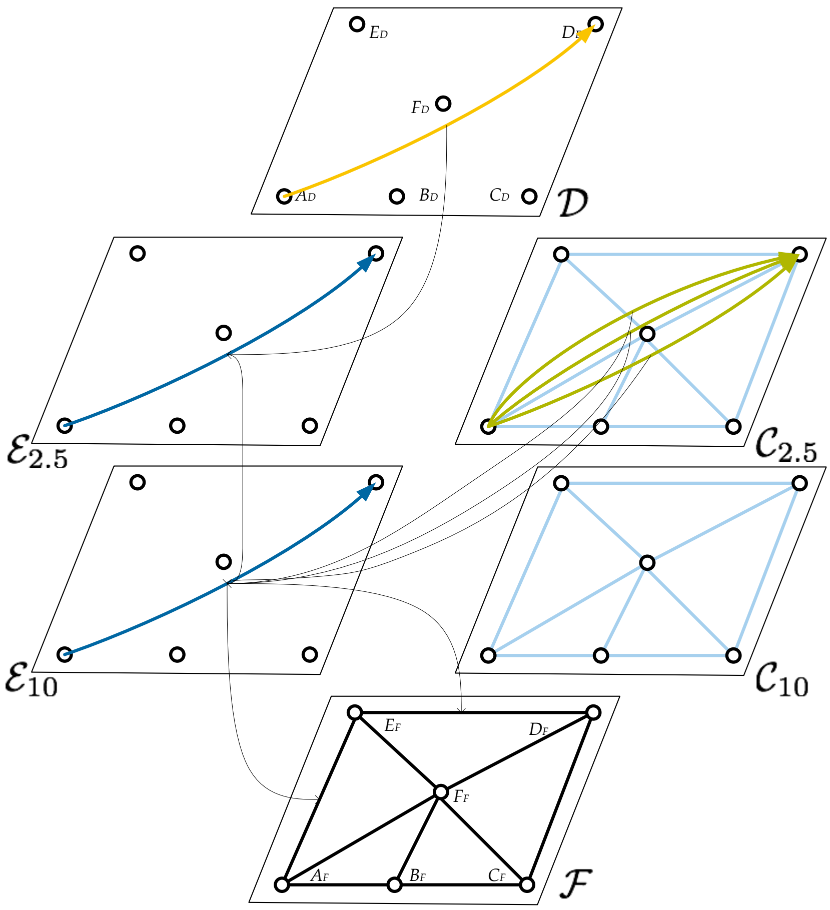


Figure 4.3: 2.5 Gbit/s demand from A_D to D_D routed via a 10 Gbit/s light-path.

is represented as an arc in the 2.5 Gbit/s equipment graph from node $A_{2.5}$ to to $E_{2.5}$. In return, the arc in the equipment graph has a reference to the used 10 Gbit/s light-path in the 10 Gbit/s layer below. The 10 Gbit/s light-path uses two fibres ($A_F - E_F$, $E_F - D_F$). Furthermore, three 2.5 Gbit/s channels, represented as three arcs in the 2.5 Gbit/s layer in the candidate hierarchy are available for further use. We chose this representation for several reasons: First and foremost, the usage of graph hierarchies is an intuitive representation of the technical reality, which greatly simplifies error-checking and further debugging. Secondly, by employing GRAPH, a library developed by Layonics [Lay09], a spin-off of the Institute of Communication Networks at TU München, we can efficiently implement these structures (considering both, implementation time and efficiency of the code itself) and easily make use of well-implemented graph algorithms.

Initial Solution The outline of the algorithm is presented in Algorithm 1. After ordering the set of all demands $\mathbb{D}$ according to the length of the corresponding shortest path (determined via Dijkstra's algorithm [Dij59]) in the fibre graph $\mathcal{F}$ with link lengths L (ll. 1), we construct the set of graphs (ll. 6) for routing and storing routing decisions as presented in the previous section. Every node of the fibre graph $\mathcal{F}$ has an associated node in every one of the equipment and candidate graphs.

Furthermore, every fibre edge (i. e. a fibre, we can use in our plannning) $e_{\mathcal{F}}$ has a corresponding pseudo edge in the candidate graphs, indicating that this very fibre can be used to establish light-paths. If two or more parallel fibres exist between a pair of nodes, we will have to consider the costs of amplifiers for every fibre separately, as we have to place amplifiers per fibre. Quite obviously, it is undesirable to use more fibres than absolutely necesseary due to the resulting amplifier costs. Hence, in order to avoid the use of parallel fibres, we increase the link weight for every additional parallel fibre by a factor of 10.

Since we usually do not have fully meshed fibre topologies, it is quite possible that a path in the candidate graph contains series of pseudo edges. Obivously, many possibilities to realize such paths exist. In our scenario with the given cost structure, it is usually beneficial to use as few transponders as possible, i. e. use end-to-end grooming. Hence in Algorithm 4, we search for the longest possible sub-paths and recursively install the necessary equipment along the fibre path (ll. 4, 5). The result of this operation, is a candidate graph with a completely empty edge from the source to the sink of the pseudo path. The remaining step in l. 7 is to "move" those edges necessary to fulfil demand d to the equipment graphs and remove them from the candidate graphs. Quite obviously, the last part in particular can be rather non-trivial. Consider for example a 10 Gbit/s light-path: This connection translates to one arc in layer $\mathbb{L}_{10}$ and four arcs in layer $\mathbb{L}_{2.5}$. If we want to use one of the 2.5 Gbit/s connections, we have to remove one arc from $\mathbb{L}_{2.5}$ (because it is occupied) and remove the arc from the $\mathbb{L}_{10}$, since we will not be able to use the 10 Gbit/s connection, once one of the 2.5 Gbit/s connections is occupied. As we will show in the next scenario, far more complex situations will have to be resolved, once more layers of different kinds come into play. The basic algorithm as shown in Algorithm 5 however remains the same. Given the upper and lower layers $\mathbb{L}_u$, $\mathbb{L}_l$, we send notifications to the upper and lower layers

Algorithm 1: Generate Initial Solution

Data: $\mathcal{F}$, $\mathcal{D}$, L

Result: $\mathcal{E}_{2.5}$, $\mathcal{E}_{10}$, $\mathcal{C}_{2.5}$, $\mathcal{C}_{10}$

begin Order demands by length
- **forall** $d \in \mathbb{D}$ **do**
  - $l\,[d] \longleftarrow$ ShortestPath(d, $\mathcal{F}$, L);
- Sort($\mathbb{D}, 1$);

end

begin Create initial graph set
- **forall** $n_{\mathcal{F}} \in \mathbb{V}_{\mathcal{F}}$ **do**
  - CreateCorrespondingNode($n_{\mathcal{F}}, \mathcal{C}_{2.5}$);
  - CreateCorrespondingNode($n_{\mathcal{F}}, \mathcal{C}_{10}$);
  - CreateCorrespondingNode($n_{\mathcal{F}}, \mathcal{E}_{2.5}$);
  - CreateCorrespondingNode($n_{\mathcal{F}}, \mathcal{E}_{10}$);
- **forall** $e_{\mathcal{F}} \in \mathbb{F}$ **do**
  - CreateCorrespondingPseudoEdge($e_{\mathcal{F}}, \mathcal{C}_{2.5}$);
  - CreateCorrespondingPseudoEdge($e_{\mathcal{F}}, \mathcal{C}_{10}$);

end

begin Create initial weight set
- **forall** $e_{\mathcal{C}_{2.5}} \in \mathbb{V}_{\mathcal{C}_{2.5}}$ **do**
  - **if** $|\{\tilde{e}_{\mathcal{C}_{2.5}} \in \mathbb{V}_{\mathcal{C}_{2.5}} : S(e_{\mathcal{C}_{2.5}}) = S(\tilde{e}_{\mathcal{C}_{2.5}}) \wedge T(e_{\mathcal{C}_{2.5}}) = T(\tilde{e}_{\mathcal{C}_{2.5}})\}| = 1$ **then**
    - $w\,[\mathcal{C}_{2.5}]\,[e_{\mathcal{C}_{2.5}}] \longleftarrow 10$;
  - **else**
    - NoOfParallelArcs$[S(e_{\mathcal{C}_{2.5}}), T(e_{\mathcal{C}_{2.5}})]$++;
    - $w\,[\mathcal{C}_{2.5}]\,[e_{\mathcal{C}_{2.5}}] \longleftarrow$ NoOfParallelArcs$[S(e_{\mathcal{C}_{2.5}}), T(e_{\mathcal{C}_{2.5}})] * 10$;
- **forall** $e_{\mathcal{C}_{10}} \in \mathbb{V}_{\mathcal{C}_{10}}$ **do**
  - **if** $|\{\tilde{e}_{\mathcal{C}_{10}} \in \mathbb{V}_{\mathcal{C}_{10}} : S(e_{\mathcal{C}_{10}}) = S(\tilde{e}_{\mathcal{C}_{10}}) \wedge T(e_{\mathcal{C}_{10}}) = T(\tilde{e}_{\mathcal{C}_{10}})\}| = 1$ **then**
    - $w\,[\mathcal{C}_{10}]\,[e_{\mathcal{C}_{10}}] \longleftarrow 10$;
  - **else**
    - NoOfParallelArcs$[S(e_{\mathcal{C}_{10}}), T(e_{\mathcal{C}_{10}})]$++;
    - $w\,[\mathcal{C}_{10}]\,[e_{\mathcal{C}_{10}}] \longleftarrow$ NoOfParallelArcs$[S(e_{\mathcal{C}_{10}}), T(e_{\mathcal{C}_{10}})] * 10$;

end

begin Route and protect demands
- **forall** $d \in \mathbb{D}$ **do**
  - Route Demand(d);
  - **if** *Requires1+1Protection(d)* **then**
    - 1+1 Protect Demand(d);

end

Algorithm 2: Route Demand

Data: $\mathcal{F}$, $\mathcal{D}$, $\mathcal{C}_{2.5}$, $\mathcal{C}_{10}$, $\mathcal{E}_{2.5}$, $\mathcal{E}_{10}$, w, d
Result: $\mathcal{E}_{2.5}$, $\mathcal{E}_{10}$, $\mathcal{C}_{2.5}$, $\mathcal{C}_{10}$

if *Layer(* d *)* = $\mathcal{C}_{2.5}$ **then**
  ShortestPath(d, $\mathcal{C}_{2.5}$, $w\left[\mathcal{C}_{2.5}\right], p_d$);
  ProvisionPath(p_d, d);
else if *Layer(* d *)* = $\mathcal{C}_{10}$ **then**
  ShortestPath(d, $\mathcal{C}_{10}$, $w\left[\mathcal{C}_{10}\right], p_d$);
  ProvisionPath(p_d, d);

Algorithm 3: 1+1 Protect Demand

Data: $\mathcal{F}$, $\mathcal{D}$, $\mathcal{C}_{2.5}$, $\mathcal{C}_{10}$, $\mathcal{E}_{2.5}$, $\mathcal{E}_{10}$, w, d
Result: $\mathcal{E}_{2.5}$, $\mathcal{E}_{10}$, $\mathcal{C}_{2.5}$, $\mathcal{C}_{10}$

if *Layer(* d *)* = $\mathcal{C}_{2.5}$ **then**
  forall $f \in$ *Route(* d *)* **do**
    forall $e \in f$ **do**
      HideEdge($\mathcal{C}_{2.5}$, e);
  ShortestPath(d, $\mathcal{C}_{2.5}$, $w\left[\mathcal{C}_{2.5}\right], p_d$);
  UnHideAllEdges($\mathcal{C}_{2.5}$);
  if *Length(* p_d *) = 0* **then**
    Unroute(d);
    forall $e \in \mathbb{E}_{\mathcal{C}_{2.5}}$ **do**
      if *!IsPseudoEdge(* e *)* **then**
        HideEdge($\mathcal{C}_{2.5}$, e);
    KDisjointShortestPaths(d, $\mathcal{C}_{2.5}$, $w\left[\mathcal{C}_{2.5}\right], 2, pS_d$);
    ProvisionPath($pS_d[1]$, d);
    $p_d \leftarrow pS_d[2]$;
    UnHideAllEdges($\mathcal{C}_{2.5}$);
  ProvisionPath(p_d, d);
else if *Layer(* d *)* = $\mathcal{C}_{10}$ **then**
  ...

Algorithm 4: ProvisionPath

Data: p, d, $\mathcal{C}_{2.5}$, $\mathcal{C}_{10}$, $\mathcal{E}_{2.5}$, $\mathcal{E}_{10}$, $\mathcal{D}$
Result: $\mathcal{C}_{2.5}$, $\mathcal{C}_{10}$, $\mathcal{E}_{2.5}$, $\mathcal{E}_{10}$, $\mathcal{D}$
begin Provision Paths
    if $e \in p$ *is pseudo edge* **then**
        subPaths $\longleftarrow$ FindPseudoSubPaths(p);
    forall *pseudoPath* $\in$ *subPaths* **do**
        realPath $\longleftarrow$ InstallPath(pseudoPath);
        Replace(pseudoPath, p, realPath);
    RealizePath(p);
end

that an arc in the path is occupied and thus not useable any more. After sending out these notifications, we can be sure that the necessary equipment was provisioned in the layers below, however, it might still be necessary to provision equipment on the current layer via `RealizeEdge`.

Algorithm 5: RealizePath

Data: p, $\mathcal{C}$, $\mathcal{E}$, $\mathbb{L}_u$, $\mathbb{L}_l$
Result: route, $\mathcal{C}$, $\mathcal{E}$
forall $e \in p$ **do**
    UsedEdgeInLowerLayer($\mathbb{L}_u$, e);
    UsedEdgeInUpperLayer($\mathbb{L}_l$, e);
    if *CorrespondingEdgeExists(* e, $\mathcal{E}$ *)* **then**
        route $\overset{+}{\longleftarrow}$ CorrespondingEdge(e, $\mathcal{E}$);
    else
        route $\overset{+}{\longleftarrow}$ RealizeEdge(e);
    RemoveEdge($\mathcal{C}$, e);

As shown in the previous section, there are many cases, where more than one possibility to satisfy a given demand exists (i. e. a 2.5 Gbit/s light-path vs. a 2.5 Gbit/s connection in 10 Gbit/s light-path). If we have to bring new equipment into the network, we can thus put more equipment than necessary ($4 \cdot 2.5$ Gbit/s) in the expectation of using the remaining capacities for other demands. This decision is made in `InstallPath`, which decides randomly which equipment to use. In the given cost-model, it is cost-effective, to use one $4 \cdot 2.5$ Gbit/s to 10 Gbit/s muxponder instead of four 2.5 Gbit/s transponders, but this is not true for the 10 Gbit/s to 40 Gbit/s case. Consequently, a probability for overprovisioning of 90% in the first and 5% in the second case lead to good results. Especially with the solution polishing described in the next paragraph, the safe choice is to put more equipment than necessary in the network.

For demands requiring protection, we route the demand as previously described and subsequently hide arcs being affected by the relevant failure states. For single fibre cuts, this is equivalent to edges using the same fibres, however it is easy to predefine more complex failure states as for example Shared Risk Link Groups (SRLGs) considered in **[KMSE09]**. Afterwards, we route the demand a second time. Should this behavior not lead to two disjoint paths, we unroute the demand and search the two disjoint paths in one step via Bhandari'sK Disjoint Shortest Path Algorithm [Bha99, p. 175] directly on the pseudo edges[8] as shown in Algorithm 3. Quite obviously, this algorithm will also fail if a necessary fibre is fully occupied, however we never observed this situation in our scenarios.

Solution Polishing Both the previously described heuristic to find an inital solution and the improvement heuristic in the next section, construct a feasible solution. Since both algorithms will install more equipment than is probably necessary, it is quite likely, that some of the equipment is not fully used in the final solution (e. g. only 2 2.5 Gbit/s of a 10 Gbit/s light-path). Our polishing function searches for exactly those cases and replaces them where possible (i. e. when enough wavelengths are still unoccupied) with more cost-effective solutions.

Improvement Heuristic The previous steps (initial solution and polishing) construct solutions for our reference scenario in less than one second, which are roughly 25% away from the optimum in [SPG+06], leaving a little less than 30 seconds from the considered time horizon to improve this solution. Our underlying strategy for this improvement heuristc is to reroute demands thus freeing equipment, which in return is not necessary anymore and can therefore be removed from the solution. In order to achieve this, we do not rely on deterministic algorithms, such as the previously presented polishing routine for example, but on a combination of a greedy search algorithm[9] and randomized searches in the spirit of GRASP [FR89, RR03].

Initialization Starting from a feasible solution (either our initial solution or the solution from the previous iteration of the improvement heuristic), we increase the costs for installing new equipment by raising the link weights of the pseudo edges to 200.

Iteration Step We pick a random set of demands and "unroute" them, that is we revoke all routing decisions which will usually cause some equipment edges to become candidate edges again. We will follow these changes as deeply as possible through the graph hierarchy. Considering our previous example of a 2.5 Gbit/s demand route via a 2.5 Gbit/s channel of a 10 Gbit/s light-path, we would obviously "move" the

[8] An even better solution could be achieved by incorporating disjointness information from the fibre layer as shown in [Bha99, p. 117]. Close examination of our algorithm however revealed, that this function was used in very rare cases only, which is why we do not expect the performance gain to justify the effort necessary for the non-trivial implementation.

[9] A greedy search algorithm always decides for the largest improvement possible. If no improvement can be found the algorithm terminates [CLR90, p. 179].

2.5 Gbit/s channel from the equipment to the candidate graph. Imagine the underlying 10 Gbit/s light-path was unused otherwise (i. e. all four 2.5 Gbit/s channels are now empty), then we can also move the 10 Gbit/s light-path to the candidate graph and thus effectively remove the equipment necessary for this light-path. If however the light-path was used otherwise, we would randomly set the associated link-weight of the 2.5 Gbit/s edge to a value between 40 and 80 (evenly distributed) in the candidate graph. After unrouting the demands, we will sequentially route them just as for the initial solution. After routing all selected demands, we calculate the costs of the new solution. We decide according to a *Simulated Annealing*[10] scheme whether we will keep the solution and try to improve it further or fall-back to a better, previous solution. Hence, given costs c_{new} for the newly acquired solution and costs c_{old} of the previous, better solution, we will accept the new solution with a probability of

$$p = e^{\frac{c_{\text{old}} - c_{\text{new}}}{T}}, \tag{4.1}$$

with T being the current *temperature*, which we decrease via

$$T = T_{\text{old}} \cdot a \tag{4.2}$$

with a cooling factor of a.

Termination A number of sensible choices for a termination criterion exist for simulated annealing. In the given scenario however, we are interested in quick solutions within a predetermined time-limit, thus we simply terminate once this time-limit has been reached. It is important however to chose the cooling factor a and the start temperature carefully: When T reaches zero, our simulated annealing routine will become a simple greedy search. On the other hand, when T is chosen too large (and a too small), our algorithm will transform into a random search, since almost any solution will be accepted. Hence, adequate parameter choices will be discussed in the following result section.

Preliminary results prepared for **[KA07]** showed that this improvement heuristic was only able to gain improvements of roughly 5% compared to the initial solution. A closer look at our network architecture reveals why: In order to gain savings, a full light-path has to be freed. To make this happen all demands using a light-path have to be selected by our random selection at the beginning of an iteration step. This probability is rather low, when the complete scenario consists of a couple of hundred demands. We could achieve further improvements by modifying the iteration step in the following way:

[10] *Simulated Annealing* is a meta-heuristic modelled after the cooling process of melted metals and was proposed independently in [KCDGV83, Čer85]. The underlying observation being that atoms of annealing metals will go through energetically unfavourable positions on the way to a perfect crystalline structure. Applied to search heuristics, this means that we will temporarily accept worse solutions on the way to an optimum. The probability for accepting these worse solutions is higher in the beginning (when the metal is still hot) and gradually decreases (i. e. cools down). For practical aspects, [HJJ03] is a comprehensive overview.

Iteration Step (modified) We randomly select one fibre, unroute all demands using this fibre and route them again as described in the original iteration step.

Results

As explained in the introduction, this optimization algorithm was devised as a heuristic counterpart to the work of Scheffel et al. [SPG+06], hence their results are the obvious benchmark. In [SPG+06], the authors were interested in determining the cost impact of muxponders in an optical backbone network, which they studied in three scenarios:

- All network elements allowed,
- no 2.5 Gbit/s transponders allowed,
- no transponders at all allowed.

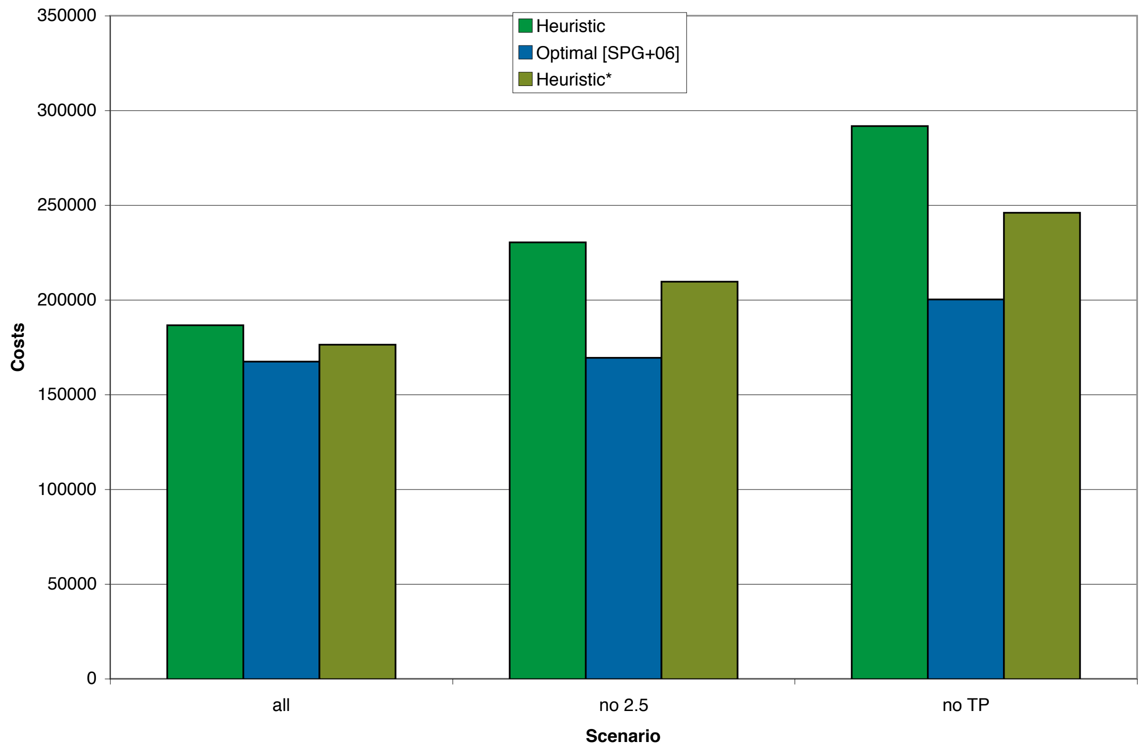


Figure 4.4: Heuristic vs Optimal Results

In Figure 4.4, we compare the average costs from ten runs of the heuristic to the optimal results of [SPG+06]. The heuristic was allowed to run for 20s and the gained results derived less than 6% from the average. We used a start temperature of $T_{\text{start}} = 100,000$ and a cooling factor of $a = 0.0001$, which is sufficent for 10^8 iterations before reaching 0. The exact value of the start temperature is rather uncritical, as long as it is in a similar order of magnitude as the costs of the solution. We leave the parameter settings (link weights, start temperature, cooling factor) constant for all scenarios.

Within $20s$, the modified heuristic (marked as "Heuristic*"in Figure 4.4) achieves a solution costing only 7.5% more than the optimal solution from the literature. A closer investigation of the log-files of the simulations performed in [SPG+06] reveals, that their MIP formulation takes roughly $90s$ to construct a feasible solution at all and about two hours to match the quality of the heuristic solution. Solution qualities will get worse if restrictions on transponder use are imposed, which is not surprising, due to the fact that the settings for the link weights were specifically chosen for the other scenario — decreasing the costs for longer detours (i. e. increasing the cost for pseudo edges) will improve these results. On an interesting side note, we discovered that the sequence in which the demands are route routed is more important in the restricted scenarios: When we changed from "longest demand first" to "shortest demand first", solution costs increased by more than 20% on average.

4.1.2 Multi-Layer Networks

As we already stated in the introduction, flexibility was one of the most prominent design criteria in the development of this heuristic and the previous scenario was merely considered to be a test case. An extension of this original scenario should thus be easily treatable using our framework. As a proof of concept, we used a multi-layer planning problem, which – to the best of the author's knowledge – has not been solved to optimality for realistic problem sizes.

Scenario

For many network providers, a considerable amount of customer traffic does not originate on the optical layer since many customers do not need the data rate of a full 2.5 Gbit/s light-path and therefor rent connections with considerably less capacity as IP or Ethernet connections. Due to the necessity of grooming these connections it is important not to restrict our planning to the optical layer. While it is perfectly feasible to perform traffic engineering via link-weight optimization of the used Interior Gateway Protocol (IGP) (e. g. Open Shortest Path First (OSPF) or Intermediate System - Intermediate System (IS-IS)), as pioneered by Fortz and Thorup in [FT00], many service providers are either reluctant to change recommended vendor settings or require a finer traffic engineering. One of the predominant technologies to achieve this goal is MPLS [RVC01, RTF+01], whose traffic engineering capabilities more resemble those of ATM[11]. MPLS with traffic engineering extensions allows for freely configurable paths with guaranteed bandwidth[12] and, depending on the vendor, an immediate reaction to hardware alarms thus matching recovery times of SONET/SDH[13]. Due to the re-

[11] Which of course does not come as a surprise considering the evolution of MPLS as a "smart" way to control ATM-switches, which was pioneerd by Toshiba as a "Cell Switching Router" [Dav00] in 1995 and Ipsilon [NEH+96b] in 1996.

[12] Which are set-up via Resource ReSerVation Protocol (RSVP) [BZB+97].

[13] An example for such a proprietary extension would be Cisco's Fast ReRoute (FRR) which has found its way into IETF standardization as [PSA05].

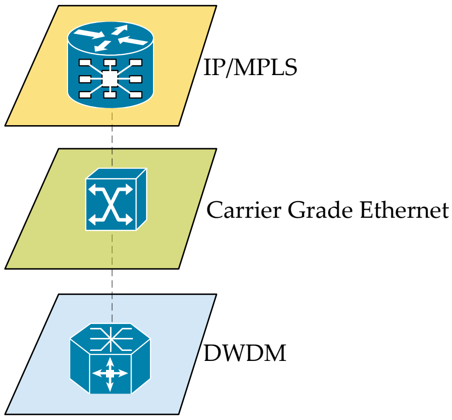


Figure 4.5: Multi-Layer Network Architecture

duced lookup complexity, hardware load on the routers can decrease dramatically[14]. Upcoming Carrier Grade Ethernet Standard(s) offer the same functionality and thus can be treated in a similar fashion by our planning algorithm.

An important limitation within IP is that one traffic flow cannot be arbitrarily splitted: Most routers distribute traffic via simple round robin algorithms and thus can only evenly distribute traffic. While so-called "coloured interfaces" (i. e. DWDM interfaces) for IP routers exist, service providers may opt for an integration involving one additional layer between IP and DWDM. In the past the choice for this layer was usually SONET/SDH[15], however Carrier-Grade Ethernet might offer an interesting compromise between cost-effectiveness (due to relatively "simple" hardware) and a considerable amount of native traffic: Since business customers start to move towards IP telephony, demand for POTS will remain constant at best. Company networks however use Ethernet as the predominant Local Area Network (LAN) technology, which in turn might move traffic from the IP layer currently used to connect company networks to a more "natural" Ethernet layer. The same is true for classic leased-line services used for the same purpose. Furthermore, Ethernet switches allow cost-effective traffic grooming and additionally offer a practicable way to deploy *shared protection* for services not requiring a dedicated protection scheme, such as 1+1 protection. The resulting network architecture then looks as depicted in Figure 4.5. It is obvious, that our degrees of freedom have increased dramatically. To make performance considerations even more critical, in order to make use of the finer granularity in the switching capabilities, we also have to consider finer granular demands, which in turn means, that we should be able to consider larger, more accurate networks like the 50 node Germany50 network.

[14] Admittedly, this can also be achieved via vendor-specific implementations of IP-lookups: Cisco Express Forwarding (CEF) [CEF] is a prominent example for this.

[15] An early overview of IP over WDM integration can be found in [GDW00].

Data Representation

The representation of our network architecture in the graph structure can be performed quite obviously using the same principles as in the first case. Since such a transformation is possible, there is no immediate reason, why the algorithms used in the previous scenario should not be applicable. Closer inspection however, reveals the need of three extensions:

1. First, it is now possible to partially occupy edges: For example if we use 5 Gbit/s on a 10 Gbit/s Ethernet link, we can now use the remaining capacity for other demands, hence using an edge no longer implies that we can remove the edge from the set of candidates. Unless the capacity is exhausted, it will stay in the respective candidate graphs, however with an attribute for the remaining capacity.
2. In our previous scenario, we had a unique relation between the different layers. Again, this is no longer true anymore. We could for example realize 1 GE link in a 10 GE link, which in return uses a 10 Gbit/s light-path. However, we could also use a 8×1 GE DWDM card. Consequently, several possiblities to use a link in upper layers exist, which have to be dealt with explicitly, since using such a link automatically rules all but one possibility out. We store these XOR relations as additional attributes within our framework.
3. Shared protection poses an interesting challenge: Not only do we have to know how the links in our network are occupied, but we also have to know, under which failure situations they have to carry which load (and thus might or might not be eligible for shared protection). Quite obviously, this could be achieved by modifying our "remaining capacity" attribute, however then all existing routines for finding working or 1+1 protection paths would have to be modified and would take a considerable performance hit (because all non-suitable paths would have to be hidden before routing every single demand). For these reasons, we introduce a third graph hierarchy (named *sharing graphs* $\mathcal{S}$), which are solely used for finding shared protection paths, while our regular candidate graphs remain unchanged. The sharing graphs work like our previous candidate graphs, with the exception of the above mentioned extension of failure state specific capacity information.

Algorithm

As stated before, the outline of our algorithms can remain unchanged, however we need to extend our routing algorithms in order to cope with the new protection scheme and the fractional capacities. Finding working and dedicated protection paths on IP and the Ethernet Layer requires only one additional step: Before running Dijkstra's Algorithm, we hide all edges which have insufficient capacities. For demands originating on one of the DWDM-Layers, no changes are needed.

For 1+1 protection, the required changes are similar. Given the shared graph $\mathcal{S}$, shared path protection can be dealt with analogously: The only difference is that we have to

Algorithm 6: Route Demand (extended)

Data: $\mathcal{F}$, $\mathcal{D}$, $\mathcal{C}_{2.5}$, $\mathcal{C}_{10}$, $\mathcal{E}_{2.5}$, $\mathcal{E}_{10}$, w, d
Result: $\mathcal{E}_{2.5}$, $\mathcal{E}_{10}$, $\mathcal{C}_{2.5}$, $\mathcal{C}_{10}$
if *Layer*(d) = $\mathcal{C}_{2.5}$ **then**
  ...
else if *Layer*(d) = $\mathcal{C}_{IP}$ **then**
  forall $e \in \mathbb{E}_{\mathcal{C}_{IP}}$ **do**
    if *RemainingCapacity*(e) $< D_d$ **then**
      HideEdge(e);
  ShortestPath(d, $\mathcal{C}_{\text{IP}}$, $w\,[\mathcal{C}_{\text{IP}}]$, p_d);
  UnHideAllEdges($\mathcal{C}_{\text{IP}}$);
  ProvisionPath(p_d, d);

consider the necessary failure states given the routing of a working path `route`(d) and have to hide edges that cannot provide sufficent bandwidth. Obviously, we have to modify our `ProvisionPath` routine as well, in order to deal with the fractional demands and to update the shared graphs.

Evaluation of the Heuristic

To the best of the author's knowledge, no solvable MIP models and corresponding algorithms equivalent to the (theoretical) capabilities of the heuristic exist. Only in very recent works by Orlowski [Orl09], MIP-models and -algorithms became available which could provide optimal results for a simplified network architecture (no sharing, limited set of light-paths) considered in our heuristic. A scientifically meaningful evaluation thus is more or less impossible – without proven results we simply lack a reference point.

Nevertheless, the heuristic has found considerable use within a collaboration partner (namely Nokia Siemens Networks) in the Efficient Integrated Backbone (EIBONE) project [EIB05], where it served as a reference for other heuristic methods. Despite its general approach and proven flexibility, other heuristics were not able to gain consistently better results in comparable running times (one minute for the Germany 50 scenario). In the remainder of this thesis however, we will focus on optimal methods, which provide a quality guarantee for a solution.

4.2 Adaptive DWDM

The previous section dealt with the central problem of speeding up backbone planning to accomodate more dynamics within the backbone network which has become possible with recent network elements. However progress is not restricted to the

Algorithm 7: RealizePath (extended)

Data: $p, \mathcal{C}, \mathcal{E}, \mathbb{L}_u, \mathbb{L}_l, D_d$
Result: route, $\mathcal{C}$, $\mathcal{E}$

```
forall e ∈ p do
    UsedEdgeInLowerLayer( 𝕃_u, e );
    UsedEdgeInUpperLayer( 𝕃_l, e );
    if CorrespondingEdgeExists( e, ℰ ) then
        route +← CorrespondingEdge( e, ℰ );
    else
        route +← RealizeEdge( e );
    if RemainingCapacity( e ) > D_d then
        RemainingCapacity( e ) −← D_d;
        RemainingCapacity( CorrespondingEdge( e, 𝒮 ) ) −← D_d;
    else
        RemoveEdge( 𝒞, e );
        if RemainingCapacity( CorrespondingEdge( e, 𝒮 ) then
            RemoveEdge( 𝒮, CorrespondingEdge( e, 𝒮 ) );
```

switching elements of optical networks, but moreover transmission technologies are continuously improving. While faster connections (which of course are always a research goal) can be easily handled by existing algorithms (just imagine adding a 100 Gbit/s layer to the heuristic), some technological improvements cause more fundamental changes in the network characteristics. As explained earlier, current DWDM technology multiplexes a number of frequency separated signals onto a common fibre. Each of these signals uses one single carrier frequency. Due to physical side effects, a frequency band is reserved around these carriers (resulting in the ITU-T G.692 [ITU98] frequency grid [Kra02, p. 17]). A transponder for such a DWDM system can usually be characterized by two parameters: a data rate (i. e. 10 Gbit/s) and a Maximum Transmission Distance (MTD), such as 1200 km. Should the light-path be shorter than the MTD, no advantages can be gained for the user. Due to progress in Digital Signal Processing where reconfigurable units become increasingly common, this simple characterization might not be sufficient for future, more *adaptive* transponders. In this section, we will give a brief introduction to two possible future technologies for transponders in next generation transport networks and will assess their performance in future networks by comparing them to existing technologies.

4.2.1 Technological Foundations

Optical OFDM

Orthogonal Frequency Division Multiplex (OFDM) is a modulation technique enjoying great popularity in both wired and wireless scenarios[16] Its basic approach of spreading the signal on a number of orthogonal sub-carriers, allows for easy adaption to vastly differing channel conditions. This distinctive strength is of great importance in the fading environments of high data rate radio transmissions as well as in the plain old copper-lines of service providers. While fibre characteristics may seem to be quite perfect in comparison, their non-linear effects become considerably worse with higher data rates and higher transmission power (albeit that is at higher rates than the previous two examples). As the name already suggests OFDM employs not only a number of conventional carriers such as Frequency Divisioned Multiplex (FDM) (which of course is identical to Wavelength Division Multiplex (WDM)), but these sub-carriers are also *orthogonal*. Informally speaking, this means that the Neighbor Channel interference among the different sub-carriers is zero, i. e. they do not interfere with each other. Translating this to a more mathematical description, we demand that for two signals around the sub-carrier frequencies f_1 and f_2 with a symbol duration T,

$$\int_0^T \sin\left(2\pi f_1 t + \phi_1\right) \sin\left(2\pi f_2 t + \phi_2\right) \mathrm{d}t = 0 \tag{4.3}$$

for arbitrary phase offsets ϕ_1 and ϕ_2 [PS02]. We can easily see that the above condition will hold, if the sub-carriers are chosen so that

$$f_1 - f_2 = \frac{n}{T} \tag{4.4}$$

with n being an integer value larger than 0. Hence for N sub-carriers, we choose the sub-carrier $x_n(t)$ such that

$$x_n(t) = \sin\left(2\pi f_n t\right) \quad f_n = \frac{n}{T}, \quad n \in \{0, 1, \ldots, N-1\} \tag{4.5}$$

Quite obviously, the symbol rate r_n of a single subchannel for a system with N sub-carriers can be reduced to $r_n = \frac{r}{N}$ with r being the symbol rate of the comparable single carrier system. The immediate benefit of spreading our signal to N sub-carriers is that we can use a different bit per symbol rate on each of the sub-carriers. This allows us to deal efficiently with different subchannel conditions, since we can easily adapt to the actual Signal-to-Noise Ratio (SNR) of the channel. While the basic prinicple of OFDM has been known for quite a while[17], the availability of cost-efficient digital signal processing and particularly Fast Fourier Transformation (FFT) however made

[16] The most popular probably being Asymmetric Digital Subscriber Line (ADSL) and Digital Video Broadcasting – Satellite (DVB-S).

[17] Doelz et al. described a multi-carrier transmission system named *Kineplex* following a proposal of W.H. Wirkler from 1949 as early as 1957 in [DHM57]. Two other classic works on OFDM are [Cha66] and [Sal67].

it a viable approach for commodity technologies such as ADSL and DVB-S. A simple OFDM system built on FFT is presented in [PS02, p. 557ff].

OFDM for optical systems (OOFDM) was firstly introduced for multi-mode fibres in [DPI01] and for free-space links in [GPJR$^+$05]. Lowery et al. presented in [LDA07] a design for single mode fibres[18], which are used in backbone networks. Yang et al. went one step further and integrated individual power allocation to the different sub-carriers [YSM08] using a well-established concept for wireless links [KRJ00]. With this addition, it is possible to adjust the bitrate quasi-continuously according to the actual Optical Signal-to-Noise Ratio (OSNR). Again, this adaptive characteristic is enabled by the availability of software-controlled DSPs — with conventional fixed hardware designs, the implementation of flexible signal-modulation would be very hard. It is important to note, that these additions come automatically with the required signal processing structure required for OOFDM. This signal processing portion however is the main source for the complexity of future coherent polarization multiplexed systems, may they be single- or multi-carrier. Thus, the cost difference of an actual transponder for 100 Gbit/s transmission (i. e. following [FDvdB$^+$08] or [JMST09]) can be largely attributed to differences in the computational complexity. Studies by Spinnler et al. [SHK08] and Poggiolini et al. [PCCF09] show that both approaches are equal in their complexity or even attribute a small advantage to OFDM. Hence we will assume that we can build OFDM and fixed-rate transponders with the same performance characteristics at the identical costs.

Adaptive DWDM

As stated before, the advantage of OFDM lies within the fact, that we adjust modulation format and power assignment on a per-sub-carrier basis. It is however also at least theoretically possible to adjust the modulation format of the single-carrier DWDM transponder according to the actual OSNRas well. We call this hypothetical technology *Adaptive DWDM*. The biggest disadvantage compared to OOFDM is that we can adjust the bit per symbol rate only discretely[19]. We will use the following case-studies to estimate the influence of this restriction.

4.2.2 Network Model

In the previous section, we could groom traffic in fixed Optical Transport Unit (OTU)-granularities using muxponders. Adaptive DWDM and OFDM transponders do not allow this, as it is quite unlikely that all links connected to a DWDM node have the same length (and hence the same transmission capacity). Due to this, we will first examine a purely transparent network (which per definition does not allow any

[18] Further important works on theoretical and practical physical aspects of OOFDM are [SYMT07] and [JMST09].

[19] One might argue that the “steps” of OOFDM are discrete as well. Compared to Adaptive DWDM however, a much finer granularity can be achieved.

grooming within the optical network), an opaque network (where we have to perform O/E/O-conversion at every node) and finally a translucent network configuration, where we will leave the decision where to perform O/E/O-conversion up to the optimization and thereby create the possibility to groom traffic. In a fixed-rate DWDM network the reason of performing O/E/O-conversion is two-fold: First, we can groom traffic, which creates the possibility of using transponders with a better price-performance ratio. Second, it might be cheaper to use two transponder pairs with a shorter reach, than one high-quality transponder pair. Adaptive DWDM and OFDM add a third dimension: Performing O/E/O-conversion at an intermediate node increases the data rate (even when using the same transponders as before) *and* allows for traffic grooming.

For grooming in the opaque and translucent case, we need to consider an additional (electrial) layer on top of our optical network simultaneously in our optimization. In order to keep our model practically manageable and to isolate the effects of adaptive DWDM and OFDM, we will not impose any limitations on the routing in this layer, i. e. a demand can be split among an arbitrary number of differently sized paths[20].

Optical Link Model

In a conventional fixed-rate DWDM network, the characteristics of a link are straightforward: Assuming a transponder t offers a data rate of C_t and has a reach or MTD of L_t, a link (may it be a point-to-point connection or a transparent light-path) l with a length of L_l can be established *iff* $L_t \leq L_l$ and will offer a data rate of C_t. Every one of these transponders costs K_t which has to be considered for the total network costs. As stated before, a hypothetical OFDM transponder would be capable of adjusting its data rate according to the OSNR on the link. The OSNR in return is depending on a multitude of different factors, whereas the length of the link has predominant influence. Using a fixed-rate transponder t offering a data rate of C_t at Quadrature Phase-Shift Keying (QPSK), we can calculate the data rate offered by an equivalent OFDM transponder using Shannon's Law [Sha49] via

$$C_{t,l}^{\mathrm{OFDM}} = \frac{C_t}{2}\left(1 + \log_2\left(\frac{2L_t}{L_l}\right)\right), \quad L_l \leq 2L_t. \tag{4.6}$$

Halfing the link length L_l improves our SNR by 3 dB, which in return allows us to use one additional bit per symbol. Furthermore, we have the ability to step down to Binary Phase-Shift Keying (BPSK) with only one bit per symbol for $L_t < L_l \leq 2L_t$.

[20] This can be achieved via Carrier Grade Ethernet or MPLS with traffic engineering extensions. These technologies allow freely configurable paths, however unequally splitting traffic is beyond the capabilities of most routers/switches. Nevertheless this problem can be circumvented by setting up paths with the capacity of the greatest common divisor. Consider for example a demand of 10 Gbit/s which has to be split among a 6 Gbit/s and a 2 Gbit/s connection. Setting up three 2 Gbit/s paths on the 6 Gbit/s connection and two 2 Gbit/s connections allows the router to equally distribute traffic among 2 Gbit/s paths.

A single carrier DWDM transponder with an adaptive modulation format would follow the discretized version of the above function, namely

$$C^{\mathrm{A}}_{t,l} = \frac{C_t}{2}\left(1 + \log_2 \left\lfloor \frac{2L_t}{L_l} \right\rfloor\right), \quad L_l \leq 2L_t. \tag{4.7}$$

Quite obviously, this data rate can grow considerably for short links and high-quality transponders. Having huge data-rates on a single transponder card, requires very capably electronics on the customer side (i. e. the grey signal), which become increasingly complex. Reviewing existing lab prototypes in the literature and the form factor of current cards, we put a limit of $C^{\max} = 160\,$Gbit/s on the resulting data rate.

Cost Model

As depicted in the description of OOFDM, we assume that adaptive transponders can be built at the same costs as their fixed-rate counterparts. While this statement was only validated for high-end transponders, we will use equivalent transponder costs even for low-end (e. g. $2.5\,$Gbit/s) transponders in the current demand scenarios. Since both systems are not yet commercially available, we will assume that traffic growth will lead to a demand/transponder-capacity ratio comparable to today. Nevertheless, the gained margins are sufficiently large so that the results remain uncritical to considerable variations in the used transponder costs. We take transponder data and costs from [HGMS08][21] and network topologies and associated demands from [OPTW07]. We compare the resulting network costs of conventional fixed-rate DWDM, adaptive DWDM and OOFDM DWDM transponders with both protected and unprotected traffic.

4.2.3 Optimization Model

For performance reasons, we found it helpful to use slightly different formulations for the transparent and the opaque/translucent case, but the basic principle is similar in all of them.

Transparent Network

To one extreme of optical networks, we have the transparent network, where every demand is satisfied via a transparent light-path from the source to the target of the demand. Similarly to the opaque case, we model the fibre topology as an undirected graph $\mathcal{F}(\mathbb{V}, \mathbb{F})$ with link lengths of L_e. On top of this fibre graph $\mathcal{F}$, we create a light-path graph $\mathcal{L}^T\left(\mathbb{V}, \mathbb{L}^T\right)$, with $\mathbb{L}^T$ obviously being the set of available transparent light-paths.

[21] which to the best of the author's knowledge is the most recent vendor independent cost model

For every demand $d \in \mathbb{D}$, we denote the set of eligible light-paths by $\mathbb{L}_d^T$ and their respective path-lengts as $L_l \in \mathbb{R}_0^+$. With path-flow variables $f_{d,l}$, we can formulate our demand requirements as

$$\sum_{l \in \mathbb{L}_d^T} f_{d,l} \geq D_d \quad \forall d \in \mathbb{D}. \tag{4.8}$$

Again we have to install enough transponders, which we ensure via

$$\sum_{t \in \mathbb{T}} n_{l,t} \cdot C_{t,l} \geq f_{d,l} \quad \forall d \in \mathbb{D}, \forall l \in \mathbb{L}_d^T. \tag{4.9}$$

In case there are more there multiple demands with identical source and destination nodes, this formulation allows end-to-end grooming. The wavelength constraint remains unchanged in its structure, however since we need to find a suitable wavelength assignment after this planning step, it is adviseable to reduce the number of wavelengths, which we allow to be occupied in this step by adjusting W_e

$$\sum_{\substack{l \in \mathbb{L}^T: \\ e \in l}} \sum_{t \in \mathbb{T}} n_{l,t} \leq W_e \quad \forall e \in \mathbb{F}. \tag{4.10}$$

Our objective remains to satisfy all demands at minimum cost, i. e.

$$\min \sum_{l \in \mathbb{L}^T} \sum_{n \in \mathbb{T}} K_t \cdot n_{l,t}. \tag{4.11}$$

We can require Path Diversity (PD) by doubling the demand D_d in the demand constraint (4.8) and ensure that the amount of traffic for d flowing over an edge e will not exceed D_d:

$$\sum_{\substack{l \in \mathbb{L}_d^T \\ e \in l}} f_{d,l} \leq D_d \quad \forall d \in \mathbb{D}, \forall e \in \mathbb{F}. \tag{4.12}$$

Opaque and Translucent Network

In this section we will consider opaque and translucent networks as the compromise between opqaque and transparent networks. As stated before, in an opaque network O/E/O-conversion takes place at every node, no matter whether the flow is simply passed through, terminated or packed/groomed differently. In *translucent* or *hybrid* networks, the decision whether a demand uses one or more transparent or opaque paths or a combination thereof is left up to the planning process.

Hence, in the opaque case, the “light-path” graph $\mathcal{L}^O\left(\mathbb{V}, \mathbb{L}^O\right)$ is identical to the fibre topology. Whereas in our translucent network, a flow $f_{d,p}$ is not forced to use a light-path that begins and ends at the very same nodes as the demand, instead it can use an arbitrary sequence p of light-paths, which in the extreme case can be only single-hop light-paths (i. e. an opaque path). Thus, instead of restricting the eligible path-set for a demand d to the set of transparent light-paths $\mathbb{L}_d^T$ as in the previous case, we consider paths made up of light-paths $p \in \mathbb{P}_d$, where every p is consisting of arcs in the full light-path graph $\mathcal{L}(\mathbb{V}, \mathbb{L})$. With these new flow variables, the demand constraint (4.8) becomes

$$[\pi_d] \qquad \sum_{p \in \mathbb{P}_d} f_{d,p} + \delta_d \geq D_d \quad \forall d \in \mathbb{D} \tag{4.13}$$

with *dummy varibles* $\delta_d \in \mathbb{R}_0^+$ which we will need to ensure feasability during the column generation process (see below). The transponder constraint (4.10) changes to

$$[\sigma_l] \qquad \sum_{d \in \mathbb{D}} \sum_{\substack{p \in \mathbb{P}_d: \\ l \in p}} f_{d,p} \leq \sum_{t \in \mathbb{T}} n_{l,t} \cdot C_{t,l} \quad \forall l \in \mathbb{L}. \tag{4.14}$$

Similarly to the previous scenarios, we can require PD by doubling the demand and limiting the flow over one fibre edge

$$[\tau_{d,e}] \qquad \sum_{\substack{p \in \mathbb{P}_d \\ e \in l}} f_{d,l} \leq D_d \quad \forall d \in \mathbb{D}, \forall e \in \mathbb{F}. \tag{4.15}$$

In the objective function, we need to put a sufficiently large cost on the dummy variables in order to force the solver to find feasible solutions (i. e. with zero-valued dummies)

$$\min \sum_{l \in \mathbb{L}^T} \sum_{n \in \mathbb{T}} K_t \cdot n_{l,t} + \sum_{d \in \mathbb{D}} M \cdot \delta_d. \tag{4.16}$$

This seemingly small change has a huge impact on the simulation performance: For every path p and demand d, we need a flow variable $f_{d,p}$. The number of paths however grows exponentially with the network size. The key to a performance improvement of the solution process is a reduction of the number of variables. Unfortunately, a straightforward measure such as considering only paths up to a certain hop length however sensible and intuitive it might be, will invalidate all optimality guarantees, i. e. we cannot be sure any more that a given solution is 0.5%, 5% or 50% away from a theoretical optimum.

4.2.4 Optimization Algorithm

By using *column generation* as a problem specific enhancement of the original optimization method we can avoid this drawback entirely, whilst signifcantly reducing the solution time of our MIP. In our MIP formulation, we have two types of variables: continuous flow variables $f_{d,p}$ and discrete transponder variables $n_{l,t}$. It is possible to generate either of these types dynamically, however the programming effort required is considerably smaller for continuous variables[22], which is why we will employ column generation for $f_{d,p}$.

[22] The pure number of variables is also larger for $f_{d,p}$ (since the path set $\mathbb{P}$ contains all light-paths), however due to the branch-and-bound process it is not quite obvious which performance gain would be larger.

As already outlined in 3.1.3 the basic principle of column generation is to start with a small set of variables to generate a feasible solution (the so-called RMP) and then iteratively add variables which will improve the solution of the RMP. Since this pricing process has to be performed at every node of the branch-and-bound tree, computational efficiency is essential. We calculate the reduced costs of a variable by inspecting the corresponding[23] dual constraints. We associate the dual variables $\pi_d \in \mathbb{R}_0^+$ with constraint (4.13) and $\sigma_l \in \mathbb{R}_0^+$ with (4.14) respectively. The corresponding constraint for $f_{d,p}$ without protection can be deduced by applying the rules from 3.1.2. Visualizing the coefficient matrix of a path-flow variable reveals the structure depicted in Table 4.2 In this representation, every entry with $\oplus$ corresponds to an entry with $+1$

	$f_{d,p}$		
	$\oplus$		
π_d		$\oplus$	
			$\oplus$
	$\ominus$		$\ominus$
		$\ominus$	
σ_l	$\ominus$		
		$\ominus$	$\ominus$
		$\ominus$	$\ominus$

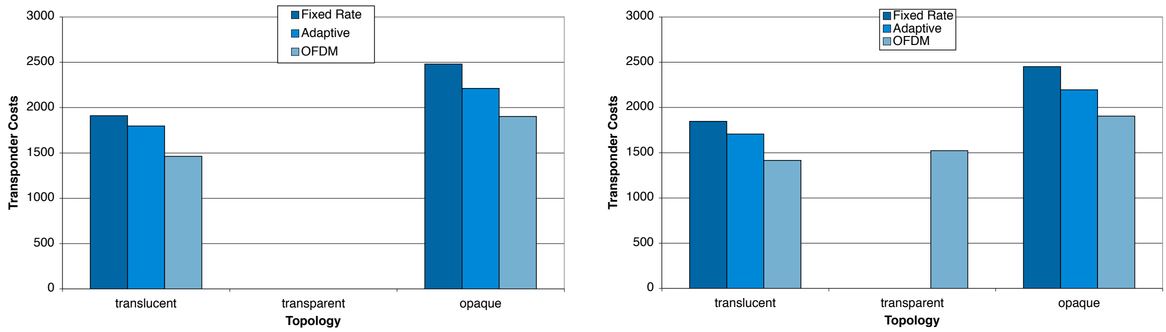
Table 4.2: Structure of the coefficient matix

in the coefficient matrix and $\ominus$ corresponds to an entry with -1, respectively. Every column represents one path flow variable $f_{d,p}$ and every row an inequality of the form of (4.13) in the first block and an inequality of the form of (4.14) in the second block respectively. Hence every demand d has a row in the matrix (first block) and similarly every link l (second block). As we can easily observe, every flow variable appears exactly one time in the first block (since it is unique per demand) and along all traversed links in the second block. Putting this into an equation and taking the inverse sign caused by the less-or-equal relation in Inequality (4.10) yields the following dual constraint for a path variable $f_{d,p}$:

$$\pi_d - \sum_{l \in p} \sigma_l \leq 0 \quad \forall d \in \mathbb{D}, \forall p \in \mathbb{P}. \tag{4.17}$$

Primal variables not yet in the solution base, but with negative reduced costs (i. e. variables that will improve the current solution), will violate their associated constraints in the dual. Hence our task is to identify paths p for which (4.17) is violated. As noted before, both π_d as well as σ_l are positive thus for (4.17) to be fulfilled, $\sum_{l \in p} \sigma_l$ has to be larger than π_d. Consequently a violated constraint can only be caused by a path p where $\sum_{l \in p} \sigma_l \leq \pi_d$. If we interpret the sum as the costs of p, the σ_l's represent the individual portions caused by the path elements. Finding the cheapest (or shortest)

[23] As shown in Section 3.1.2



(a) 40 Wavelengths

(b) 60 Wavelengths

Figure 4.6: Transponder Costs for the Nobel US Network

path in regard of the σ_l can be performed by a shortest path algorithm such as Dijkstra's Algorithm [Dij59] in polynomial time. In case the cost of this cheapest path for a demand d is smaller than π_d, we will add a new path-flow variable to the MIP. After we have performed the pricing for every demand, we reoptimize the LP[24]. Once we are unable to find variables with negative reduced costs (i. e. the costs of the shortest path are larger than π_d), we have solved the LP to optimality.

For demands requiring 1+1 protection, we associate the variable $\tau_{d,e}$ with 4.16 which changes the associated constraint for $f_{d,p}$ to

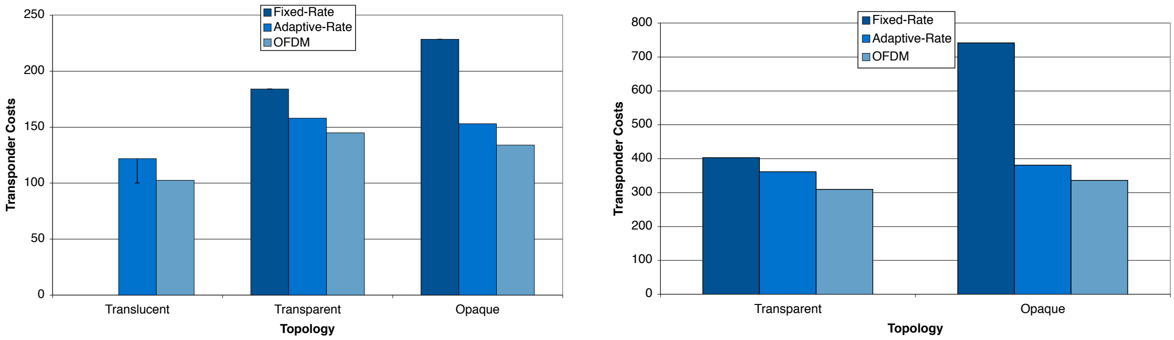
$$\pi_d - \sum_{l \in p} \left(\sigma_l + \sum_{e \in l} \tau_{d,e} \right) \leq 0 \quad \forall d \in \mathbb{D}, \forall p \in \mathbb{P}. \tag{4.18}$$

4.2.5 Case Study

In order to evaluate the possible gains from a network perspective, we implemented the above descibed MIPs using GRAPH [Lay09] for data-representation and graph algorithms, SCIP [Ach07][25] as branch-and-bound framework with ILOG CPLEX 9.13 as LP solver [CPL09]. Inspecting Figure 4.6a, we can easily see that OFDM compared to fixed-rate transponders offer a cost advantage between 20% and 30% with the adaptive single-carrier transponder between both extremes. In the 40 wavelength per-fibre case, a transparent solution is not possible with any of these technologies. With 60 wavelengths a transparent solution is possible however with OFDM transponders, which is quite close to the translucent solution. Otherwise the wavelength increase does not change the relative position of the transponder types.

[24]Keep in mind that although we are solving a MIP, in the nodes of the branch-and-bound tree, we are solving linear relaxations of the original problem augmented with additional constraints, making a previously found fractional solution infeasible.

[25]Pure CPLEX does not offer the possibility to add variables during the branch-and-bound process, i. e. column generation can only be performed in LPs only.



(a) 40 Wavelengths

(b) 80 Wavelengths and Path Diversity

Figure 4.7: Transponder Costs for the Germany17 Network

Requiring Path Diversity protection slightly increases the gap between OFDM and fixed-rate transponders. Due to the increased symmetries caused by identical link-capacities, we were not able to compute a solution for the fixed-rate translucent scenario. A transparent solution is not possible.

In the Germany 17 network the transponders behave slightly differently due to the lower network load. Similarly to the previous case OFDM, transponders offer a cost advantage of more than 21% in a transparent topology and in the opaque case of more than 41%. The limits of our model become more prominent in the larger network and we are not able to calculate a translucent solution for the fixed-rate transponders even without PD – this grows significantly worse with PD, where we do not come to solutions better than the opaque configuration. The opaque configuration itself however highlights the advantages of OFDM with a gap of more than 50% between fixed-rate costs and OFDM.

Compared to fixed-rate DWDM, the adaptive technologies offer a considerable advantage. In all cases the margin grew when considering protected traffic because the disjointness constraints force the use of longer links requiring more expensive transponders which could be avoided previously. In general, we can state, that the cost savings possible with adaptive transponders are large enough, that these transponders could be priced higher, while still being attractive for network operators.

Simulation Performance

While the transparent and opaque scenarios could be solved to optimality within minutes (the latter thanks to column generation), the translucent cases were considerably more difficult. It is important to emphasise that we are trying to solve a very hard optimization problem: Even cutting-edge algorithms as developed in Orlowski's Ph.D. thesis [Orl09] are only capable of solving these problems with a highly reduced path-set, i. e. they consider five light-paths per node-pair. With OOFDM however, this simplification is hard to justify since quite long paths are possible albeit at lower data-rates. Nevertheless cutting down the number of paths helps immensely: While we are capable of solving the Nobel US network within minutes to optimality with a full light-path set (due to the geographical size this set however is rather small), we cannot find a considerably better solution for the Germany 17 network (where more or less every transponder is capable of setting up light-paths all across the network). Cutting down the number of paths, helps finding solutions in the OOFDM case (albeit with a considerable gap left for improvement). The key to a performance increase would be the generation of light-path variables, which we will discuss in the summary of this chapter.

4.3 Summary

The contributions of this chapter are twofold:

1. In the first part, we presented a heuristic optimization method suitable for a variety of planning problems in today's multi-layer networks. In a comparison with known optimal solutions, we could demonstrate, that the heuristic strikes a good balance between speed and solution quality. Furthermore, the ability to control the time used to improve the first feasible solution, makes the method useable in a wide range of applications ranging from quick evaluations of what-if scenarios to (possibly online) planning systems for more dynamic scenarios in future optical transport networks. Using this inherent flexibility, the developed framework is easy to enhance. This was demonstrated by an extended version capable of planning Carrier Grade Ethernet over DWDM scenarios.
2. In the second part, we estimated the cost-efficiency of yet hypothetical adaptive transponder technologies. We formulated the planning problem of all three thinkable network architectures (opaque, transparent, hybrid) as a MIP. Based on complexity evaluations in the recent literature, we compared the resulting total transponder costs for various network scenarios and could show that adapative transponders would indeed offer large advantages compared to conventional fixed-rate transponders. As expected, finer granularities as a consequence of the use of OOFDM are able to further improve the cost efficiency.

Open Issues

It is important to notice that our planning methods are far from being perfect: Since we have no optimal bound for the multi-layer scenarion in the first part of this chapter, we cannot assess the solution quality of the heuristic. A comparison with results generated by state of the art multi-layer optimization schemes as presented in [Orl09] could close this gap at least for the smaller and simpler scenarios considered.

Due to its difficult structure, these methods will not improve the situation in the translucent cases of the second part of this chapter. Further research is necessary to evaluate whether and how it is possible to add variables for the light-paths and the necessary inequalities at the same time. A promising approach for this (at least as long as we do not consider protection) would be to introduce an intermediate layer of flow variables between our IP path-variables and light-path path variables. The coupling of both layers would then take place on a per node-pair flow-variable, thus allowing the generation of IP paths on top and light-paths from below. Due to the granular nature of the latter a full branch-and-price approach as presented in the following chapter will be necessary.

5 Advanced Resilience Strategies and Failure Localization

"Protection is not a principle, but an expedient."[1]

In the planning approaches of the previous section, we tried to find cost-efficient network configurations for demands which we protected via PD protection against single-link failures. Reinspecting the results of Sections 4.1 and 4.2 makes it apparent that despite cutting-edge transmission technologies, the capacity overhead for PD protection is quite immense – and PD is a *lower* bound for the more common 1+1 protection.

We will thus give a short overview over existing resilience methods and identify capacity sharing as the most promising approach to reduce the protection costs. Among a multitude of thinkable mechanisms SBPP is one of the most promising. While existing literature suggests that SBPP behaves quite favourable. Provable optimal results however could only be achieved for rather small networks. Thus, we will develop a branch-and-price approach allowing us to solve larger network instances to satisfying optimization gaps in the first part of this chapter.

Reducing the costs of resilience however is only following one group of customer requests. Since exceedingly critical services rely on transport networks, resilience even against less probable dual-link failures can be of considerable importance. It is however well-known, that protection against all dual-link failures is prohibitively expensive and not even possible for all demands (in very few of the known backbone topologies all nodes are three-connected). Assuming that a node in the metro- or aggregation structure below is connected to two distinct backbone nodes, which we call *dual-homing*, we have additional degrees of freedom to provide resilience in the core network. Subsequently we develop a MIP formulation to evalaute this approach and demonstrate in a case study, that exploiting this property can be a very cost effective method for both single- and dual-link failure protection.

In the last section of this chapter, we will focus on the localization of a failure. With transparent or translucent networks, the failure can natively only be localized to exist between the end-points of the failed light-path without additional measurements. In case multiple light-paths are affected, one solution would be a root failure analysis,

[1] Benjamin Disreali, *Speech on Agricultural Interests*

which however might fail for softer failures, which might only fail light-paths longer than a certain length. We opt for a different solution and propose to place measurement equipment at crucial points in the network for which we present an optimization scheme to achieve the "best" localization for a restricted budget.

5.1 Resilience Overview

5.1.1 Failure Causes

In public perception, today's communication networks are very reliable — a nationwide black-out of five hours due to a software problem in the home location register of one of the major mobile-phone providers in Germany is such a rarity, that it was *the* top news item for a few days. On the other hand, outages which seem not to be directly related to communication systems per se, such as the TV black-out during the football championship, highlight the dependence of our society on working means of communication.

When we try to analyze and improve network resilience, it is inevitable to find information about about the cause of network outages. One of the more reliable and rich sources was the Federal Communications Commission (FCC), however the aftermath of the terroristic attacks on 9/11 was used to increase the level of reporting required by providers whilst drastically decreasing the amount of publicly available information[2].

Available data (from previous years) lists fibre cuts as the most prominent cause for network outages (with a Mean Time Between Failures (MTBF) of 275 – 1000 years per kilometer). Power outages however are "an area of concern" [nsr08, p. 8ff.] of the NSRC and are even listed as a top-cause in a study of a regional provider as quoted in [Gru07]. In contrast, node failures seem to be relatively rare events, which can be easily attributed to the extensive redundancy options offered by hardware vendors (redundant power supplies, switching fabrics, backplanes, etc.) as demonstrated in [Ogg01, p. 119ff.][3]. The authors of related literature such as Grover [Gro04], Crawford [Cra92] and Verbrugge et al. [VCD+05] come to similar findings. Due to the prominence of link failures we will focus on cost-effective survivability of single-link failures in Section 5.2 and present a new resilience strategy for dual link-failures in Section 5.3 and finally improve failure localization in transparent and translucent networks in Section 5.4.

[2]Quite ironically, this makes it impossible for the providers to compete on the reliability of their respective networks [Sul06], which especially for cell-phone providers in the US still seems to be an issue. The updated reporting requirements are available at `http://www.fcc.gov/pshs/techtopics/techtopics15.html`. Obviously, bodies and organizations previously working on regulations and analysis of US telco networks such as the Network Reliability Steering Committee (NSRC) are equally crippled [nsr08, p. 4].

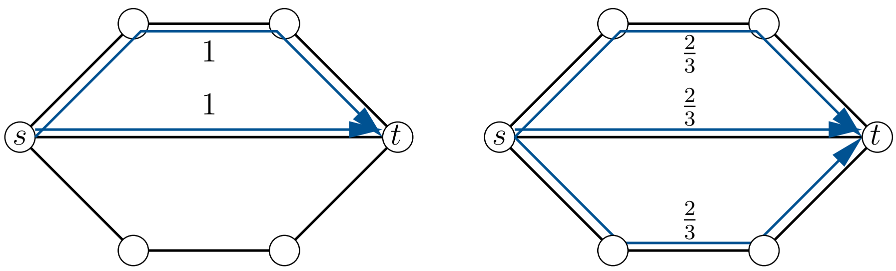
[3]The author estimates the downtime per year for a high-end router to be 24 minutes [Ogg01, p. 129], which we will consider to be an accurate value (the author works for the router vendor).

5.1.2 Resilience Schemes

As already noticed by Gruber in [Gru07] vocabulary related to resilience (which we use as the most general form for any kind of reaction against a failure following the IETF definition in [AMA+99, p. 19]) is *fuzzy* to say the least and very much dependent on the technical and/or historical background of an author, a mechanism, etc. Hence, Gruber introduces a classification framework which sorts all methods according to eight *building blocks* and further hierarchically sorted finer criteria. We will use this framework as an outline of our characterization of the following methods, which we will use repeatedly in this chapter.

1+1 Protection

1+1 protection can be considered to be one of the classic protection schemes for optical networks. In a DWDM system with 1+1 Automated Protection Switching (APS)[4] , we will transmit our signal via exactly two (at least) link-disjoint routes in parallel. At the receiver side, we can consequently decide which of the signals is "better" according to a previously defined criterion (such as the OSNR) and select one of the signals accordingly. Advantages and disadvantages are equally obvious: Once the two routes have been established (which happens during the initial setup of the connection), we need no further signaling even in case of a failure. Despite this simplicity, it is possible to construct systems which will not suffer from a single bit-error during a failover [Gro04, p. 123] and negligible fail-over times. On the other hand, we will need more than twice the capacity compared to simple shortest-path routing, as two link-disjoint routes are usually longer than twice the shortest-path (which would be the best case). Realizing similar functionality in higher layers was not (easily) possible



(a) 1+1 Protection (b) Path Diversity

Figure 5.1: 1+1 Protection vs Path Diversity (PD)

for quite a while: In pure IP for example, it is not possible to simply duplicate packets and send them via disjoint routes[5]. Only recent standardization efforts for Transport

[4]standardized in [ITU06b] – the first revision of the standard was already available in 2003

[5]Since an IP router at best can only determine the previous hop of the packet, it is not possible to guarantee that the routes taken by the two packet are disjoint.

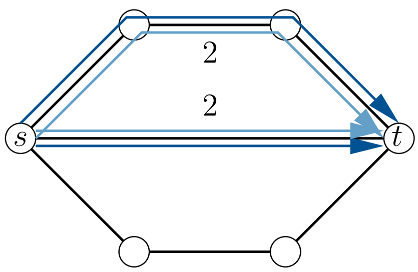
MPLS (T-MPLS)[6] and Ethernet[7] introduced this seemingly basic protection method in upper layers.

Path Diversity (PD)

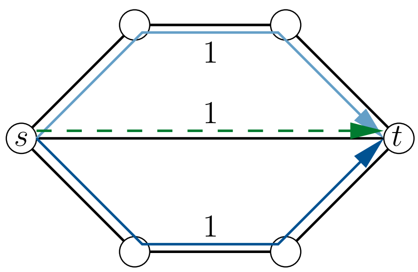
As Orlowski notes in [Orl09], an exact MIP-model for 1+1 protection is computationally quite demanding, since the restriction to exactly one working and one backup path requires integer variables, which in return considerably increases the complexity of our model. Due to this reason Dahl and Stoer introduced the concept of *PD* as a relaxation of 1+1 protection [DS98]. The basic idea is to limit the amount of traffic of one demand traversing one edge and thus ensuring that the remainder of the traffic survives in case of e. g. a single-link failure. Consequently, in order to protect a demand d with a required bandwidth of D_d, we require to route twice the demand, but forbid to route more than D_d on one edge. Inspecting Figure 5.1b illustrates why PD is a relaxation of 1+1 protection: Using three paths, each carrying $\frac{2D_d}{3}$ is a valid solution for PD, but not for 1+1 protection.
From a network engineering point of view, PD could be realized via TE-capable technologies such as MPLS or Carrier Grade Ethernet.

Shared Backup Path Protection (SBPP)

SBPP [Gro04, p. 141, pp. 411][8] attempts to reduce the amount of extra bandwidth while keeping many advantages of 1+1 protection. The basic principle is similar to *ATM Backup VP Protection* as presented in [KST94, KHT95, KT95]: For every commodity, we create a 1:1 dedicated protection path. Thus a backup path is setup before an error actually occurs, but no traffic is sent along the path until an error occurs. Following this concept, disjoint working paths can share backup capacity as depicted in Figure 5.2b.



(a) 1+1 Protection



(b) Shared Backup Path Protection

Figure 5.2: 1+1 Protection vs Shared Backup Path Protection (SBPP)

[6] MPLS with extensions defined in ITU-T Recommendation Y.1720 [ITU06c] is called *T-MPLS*.
[7] again standardized as ITU-T Recommendation G.8031/Y.1342 [ITU06a]
[8] Gruber calls this mechanism *Shared End-to-End Path Protection* and characterizes it in his classification framework in [Gru07, pp. 56].

In reality we can achieve such a setup with MPLS with traffic engineering extensions (MPLS-TE), where we establish one working and one backup path via RSVP [BZB+97]. With Ethernet, similar capabilities are available via [ITU06a]. In DWDM networks ITU-T recommendation G.873.1 [ITU06b] defines the necessary capabilities.

Global Restoration

Global restoration[9] is the most efficient resilience scheme possible, due to its highest degree of freedom: Just as SBPP it allows sharing of protection capacity, but more than one protection path per demand can be established. Traffic can be split arbitrarily and most importantly even paths not directly affected by an error can be reconfigured. As such this resilience scheme is not very realistic, although it *could* be achieved by a central entity receiving suitable failure reports and reconfiguring the network accordingly. Nevertheless, as an optimality bound for other resilience schemes it is quite valuable as already demonstrated in [Gru07, pp. 122], **[GK06]**, and [SK06].

5.1.3 Implications for This Work

Protection vs. Restoration

The difference beween protection and restoration lies largely within the time frame in which a resilience mechanism is established. For protection schemes this usually means "pre-planned, pre-configured and pre-established" while restoration mechanisms "are calculated, configured and established dynamically after the detection of a fault" [Gru07, p. 43]. Despite huge technical differences, protection and restoration mechanisms can have similar bandwidth requirements or may even be planned with identical methods. Recovery times[10] however may differ considerably, but are usually not within our planning schemes.

Multi-Layer Considerations

As layed out in 2.1.1, modern networks are in almost all cases multi-layer networks, thus failures can occur on different layers as well (e. g. a fibre-cut or the outage of a line-card in an IP-router). While it may seem natural to provide resilience at various layers as well, this is at least debatable from a bandwidth efficiency point of view. A lot of research has been conducted during the last decade on this matter (usually with backbone networks in mind), however similar challenges have been met

[9]Stidsen and Kjærulff refer to this as *Complete Rerouting Protection* in [SK06].

[10]Recovery times in itself have almost reached the status of a philosophical dispute — not so much the time which can or cannot be achieved by certain methods, but the times required by certain services which Grover describes quite tellingly as "the 50ms myth" in [Gro04, p. 111f.]. In contrast to the standard 50ms requirement, some service providers (most notably SPRINT) find even pure-IP restoration schemes viable and cost-effective [ICBD04].

since the very beginning of computer networks. A classic work by Saltzer, Reed and Clark [SRC84] aims at slightly different applications, however we consider the essence of their findings to still be suitable in our context:

> The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. [. . .] Low-level mechanisms to provide theses functions are justified only as performance enhancements.

In our context, providing resilience at a lower layer (e. g. for example 1+1 protection in the DWDM layer), cannot cope with errors on higher layers, thus we have more protection than necessary in our network ("redundant") because resilience against upper layer errors will require further capacity. Whether or not this is compensated for by the fact that resilience on lower layers *may* be cheaper (due to lower card-costs) is debatable. Completely banning resilience mechanisms in lower layers however is impossible: As shown in Section 2.1.1 customers demands can terminate at almost every layer and ask for resilience, due to the fact that they want to shift responsibility to the service provider and – blatantly speaking – as long as they are willing to pay, there is no reason for a service provider to deny this request. Operational difficulties as mentioned in Section 2.3 are another reason to employ several resilience schemes at the same time. In all chapters of this thesis, we provide resilience at the highest layer possible, although the heuristics presented in the previous section can be (and as a matter of fact was) extended to provide resilience in other layers as well. This paradigm has the positive side-effect of requiring the minimum amount of coordination between the layers: If a link goes down, the upper layers can notice this with their own methods, which might be sufficiently fast [ICBD04] especially with advanced failure detection methods as employed for example by Fast Rerouting [PSA05].

Multiple Failure Considerations

In the remainder of this chapter, we will focus on single and dual-link failures, quite simply because they are the most likely failures under the assumption that link failure probabilities are independent compared to other failure combinations. Upon this assumption Gruber deduces in [Gru07, p. 30] from generic estimations, that in a network of up to 50 nodes, demands have to be protected against single- and double-link failures as well as single node failures to achieve four nines of availability.

In reality this assumption is not always true: If for example a fibre-cut had occured and during the reparation of this failure a second fibre-cut would occur, these events can be considered to be independent. If however two links fail (possibly at the same time) because they use a common resource (as for example a bridge which breaks down) or the optical amplifiers along several links fail due to power outage, these events are *not* independent at all.

We can describe these related risks in *SRLGs*, which group links which *share* a resource that can fail (bridges, power-lines, ...). It is however quite a task to find these shared risks in backbone networks: While a provider should know the disjointness of the ducts used, it is quite a stretch to expect that he knows under which conditions which components will loose power, especially considering the fact that outages like the almost total blackout of the east-cost in August 2003 surprised experts as well.

An important notion of these SRLGs, is that they can easily grow to considerable sizes. While outages of such impact as the outage mentioned in the previous paragraph might be rare events, it is interesting for high-priority applications to find paths through the network which minimize the cumulative risk caused by these risk groups, since it might simply be impossible to find fully disjoint paths. In **[KMSE09]**, we develop two algorithms based on MIP-formulations and test them on two large national networks. As it was impossible to find realistic SRLG-assignments for these test-cases, we developed our own set based on earthquake probabilities for geologic regions. While it might not be realistic for many providers to consider the possibility of earthquakes in their resilience planning schemes, we could show that our algorithms were able to reduce the number of affected links in case of such an SRLG-failure significantly despite the size of the SRLGs, but ommit an in-depth presentation at this point for reasons of brevity.

5.2 Shared Protection Methods

As argued in Section 2.3, there is a strong interest from the service provider side to decrease the network costs and with that, the costs for resiliency without sacrificing quality. Additionally 1+1 protection might not be offered by every hardware platform on higher layers, while SBPP might very well be available[11] despite the perceived higher complexity of SBPP. However as Grover already discovered [Gro04, p. 477], planning SBPP is "a difficult to solve" optimization problem. Hence, we will present a promising advanced optimization algorithm and compare its computational effectiveness to readily available strategies.

5.2.1 Network Model

We assume our standard two layer network with an opaque optical network following Section 3.3.3 in the lower layer. On top of this layer, we consider an IP/MPLS or Carrier-Grade Ethernet layer, however we do not allow traffic bifurcation any more: Every demand has to be routed over exactly one working-/protection-path pair. The reason to forbid bifurcation lies within the fact that routers usually cannot split traffic arbitrarily. No stub release occurs, i. e. the capacity for the working path is always reserved and cannot be reused even if the working path is affected by a failure. We take

[11] Which for example was true for MPLS until the respective standardization effort [ITU06c]

only single-link failures into account thereby our set of network states $\mathbb{S}$ is identical to the set of fibres $\mathbb{F}$ (or edges in $\mathbb{L}^O$ for that matter) and the failure-free state s_0.

5.2.2 Optimization Model

In this section we will briefly review the classic formulation for SBPP-planning, highlight the problem for our optimization approach and consequently introduce our path-based formulation.

Classic Formulation

As usual, we model the fibre topology as an undirected graph $\mathcal{F}(\mathbb{V}, \mathbb{F})$ and the demands as digraph $\mathcal{D}(\mathbb{V}, \mathbb{D})$. We will consider protection against single-link failures, which we express by the set of network states $\mathbb{S}$ consisting of $\mathbb{F}$, plus an additional state s_0, indicating the failure-free network. For every commodity $d \in \mathbb{D}$ of size D_d, we have to select one path-pair $p \in \mathbb{P}_d$, whose working path p^w and backup path p^b are edge-disjoint on $\mathcal{F}$.

$$[\pi_d] \qquad \sum_{p \in \mathbb{P}_d} f_{d,p} = 1 \qquad d \in \mathbb{D} \tag{5.1}$$

On the edges e used by our path pair p, we have to reserve a capacity of D_d in all network states for arcs used by p^w and D_d on the edges used by p^b, but only for those network states affecting p^w. Summing this over all demands d, we gain the used capacity on l in network state s $u_{l,s}$

$$[\sigma_{l,s}] \qquad \sum_{d \in \mathbb{D}} \left(\sum_{\substack{p \in \mathbb{P}_d: \\ l \in p^w}} f_{d,p} + \sum_{\substack{p \in \mathbb{P}: \\ l \in p^b \wedge \\ s \cap p^w \neq \{\} \wedge \\ s \cap p^b = \{\}}} f_{d,p} \right) \cdot D_d \leq u_{l,s} \qquad \forall l \in \mathbb{L}^O, \forall s \in \mathbb{S}. \tag{5.2}$$

We will use a straightforward minimization of the used capacity in the network

$$\min \sum_{l \in \mathbb{L}^O} u_l \tag{5.3}$$

as discussed in 2.2.2 and our slightly more complex CAPEX-model considering the transponder costs as presented in Section 3.3.3, which we briefly reproduce for the sake of completeness:
We have to install a sufficient number of transponders on our light-paths (in this case obvioulsy only opaque links $\mathbb{L}^O$) to satisfy the worst-case capacity requirements

$$\sum_{t \in \mathbb{T}} n_{l,t} \cdot C_t \geq u_{l,s} \qquad \forall l \in \mathbb{L}^O, \tag{5.4a}$$

while limiting the number of wavelenghts to the number of wavelengths available in our system

$$\sum_{\substack{l \in \mathbb{L}^O: \\ e \in l}} \sum_{t \in \mathbb{T}_l} n_{l,t} \leq W_e \qquad \forall e \in \mathbb{F}. \tag{5.4b}$$

Consequently our goal is to minimize the costs generated by the transponders, namely

$$\min \sum_{l \in \mathbb{L}^O} \sum_{t \in \mathbb{T}_l} n_{l,t} \cdot K_t. \tag{5.4c}$$

Quite obviously, the number of disjoint path-pairs can be huge up to the point where they become not manageable by a MIP-solver any more. For every one of these path-pairs we need one binary variable. Similarly to related continuous problems (e. g. [GK06, SrPR+07, KS08]), we cannot simply reduce the path-set without sacrificing optimality. If we put our integrality constraint aside for the moment and just try to solve the linear relaxation of the problem (which would mean that we can split traffic among several path-pairs arbitrarily), one solution to improve scalability would be to use column generation. In order to determine variables which would improve a solution of an RMP (i. e. with a limited number of paths), we have to calculate the reduced costs of our variables $f_{d,p}$, which are (using the dual variables denoted in angular brackets in equations (5.1) and (5.2)).

$$\pi_d - \overbrace{\sum_{s \in \mathbb{S}} \sum_{l \in p^w} \sigma_{e,s}}^{\text{working path}} - \underbrace{\sum_{\substack{s \in \mathbb{S} \\ s \cap p^w \neq \{\}}} D_d \sum_{l \in p^b} \sigma_{l,s}}_{\text{backup path}} \leq 0. \tag{5.5}$$

Improving path-pairs would violate this constraint, i. e. the costs for the working path and the backup paths would be smaller than π_d and thus the sum larger than zero. While we can easily interpret this problem as a shortest cycle problem or a shortest path problem with resource constraints, the costs have one difficult property: The costs of the backup path p^b depend on the working paths, due to the fact, that the capacity of the backup path can be shared during network states not affecting p^w. This so-called *Quadratic Cost Disjoint Path Problem* has been shown to be $\mathcal{NP}$-hard in [SrPR+07, Spo08] by reduction to 3SAT [GJ79]. Unfortunately, this means we will only find "smart" brute force algorithms[12], such as the rather complicated labeling algorithm proposed in [Spo08]. While this algorithm proved to be quite successful as studied in [DZP+08], an efficient integration of a branching scheme would have required another complete reimplementation of the algorithm. Due to this fact, we opted for another way using a different formulation, transferring some of the complexity in the MIP.

[12] unless $\mathcal{NP} = \mathcal{P}$

Path-based Formulation

In order to achieve this, we separate working- and backup-path variables, i. e. instead of one single variable for a path-pair, we now have two variables, namely $f^w_{d,p}$ for the working-path and $f^b_{d,p,p'}$ for the backup-path. While the meaning of the indices of the former are quite obvious (demand d uses path p), the latter deserves a bit more explanation: This means, that the backup path variable $f^b_{d,p,p'}$ will use the path p' for protecting the demand d along working path p. Consequently our demand constraint changes to

$$[\pi_{d,s}] \qquad \sum_{\substack{p\in\mathbb{P}:\\ s\cap p=\{\}}} f^w_{d,p} + \sum_{\substack{p\in\mathbb{P}_d:\\ s\cap p\neq\{\}}} \sum_{\substack{p'\in\mathbb{P}'_{d,p}:\\ s\cap p=\{\}}} f^b_{d,p,p'} \geq 1 \qquad \forall d\in\mathbb{D}, \forall s\in\mathbb{S}, \tag{5.6}$$

i. e. , we have to ensure that we have either a working or a backup path in network state s. It is important to note, that we restrict the use of backup paths to states, where the associated working path p is affected by the error. In order to forbid the use of backup capacity only (which would result in a global restoration scheme), we force the use of at least one working path

$$[\tau^w_d] \qquad \sum_{p\in\mathbb{P}_d} f^w_{d,p} \geq 1 \qquad \forall d\in\mathbb{D}, \tag{5.7}$$

and restrict the number of backup paths used to at most one

$$\left[\tau^b_d\right] \qquad \sum_{p\in\mathbb{P}_d} \sum_{p'\in\mathbb{P}'_{d,p}} f^b_{d,p,p'} \leq 1 \qquad \forall d\in\mathbb{D}. \tag{5.8}$$

Similarly to our previous formulation, we can calculate the used capacity on an arc l in network state s as

$$[\tau_{e,s}] \qquad \sum_{d\in\mathbb{D}} \left(\sum_{\substack{p\in\mathbb{P}_d:\\ l\in p}} f^w_{d,p} + \sum_{\substack{p\in\mathbb{P}_d:\\ s\cap p\neq\{\}}} \sum_{\substack{p'\in\mathbb{P}'_{d,p}:\\ l\in p'\wedge\\ s\cap p'=\{\}}} f^b_{d,p,p'} \right) \cdot D_d \leq u_{l,s} \qquad \forall l\in\mathbb{L}^O, \forall s\in\mathbb{S}: l\cap s=\{\}. \tag{5.9}$$

Having calculated the used capacity, we can dimension the number of transponders similarly to the previous model:

$$\sum_{t\in\mathbb{T}} n_{l,t}\cdot C_t \geq u_{l,s} \qquad \forall l\in\mathbb{L}^O, \forall s\in\mathbb{S}: l\cap s=\{\} \tag{5.10a}$$

$$\sum_{\substack{l\in\mathbb{L}^O:\\ e\in l}} \sum_{t\in\mathbb{T}} n_{l,t} \leq W_e \qquad \forall e\in\mathbb{F}. \tag{5.10b}$$

$$\min \sum_{l\in\mathbb{L}^O} \sum_{t\in\mathbb{T}} n_{l,t}\cdot K_t \tag{5.10c}$$

Obviously our MIP formulation suffers from the same deficiencies as before: The number of variables is immense and thus solving the MIP via readily available solvers will take a considerable amount of time and memory.

5.2.3 Optimization Algorithm

Summarizing, we are left with the following choices to handle our planning problem:

1. Solve the problem with a standard MIP-solver in its original formulation. While this will lead to optimal results, this is not possible on current hardware, for realistic problem sizes and time-frames.
2. Solve the problem with a specialized heuristic procedure (such as the one presented in Section 4.1) while once again losing all solution guarantees.
3. Reduce the number of paths according to a reasonable scheme (e. g. path-length) and solve the problem with a standard MIP-solver. While this is certainly realistic, we can neither guarantee finding a feasible nor an optimal solution.
4. Solve the relaxed problem to optimality with column-generation (which we claim is efficiently possible with the second formulation) and use the set of variables generated as the variable base for a standard MIP-solver. Again, this method will probably lead to good results, but we can not guarantee finding an optimal or even a feasible solution as Anbil, Tanga, and Johnson report in [ATJ92] in crew-pairing problems.
5. Integrate column generation into the branch-and-bound process. This approach is called *branch-and-price* or *integer column generation*. As already mentioned in Section 3.2.2, the difficulty lies within the fact, that the branching process (which is equivalent to adding constraints in the nodes) changes our pricing problem.

In the following we will follow strategy (5) and develop a branch-and-price algorithm. In order to achieve this, we will first present a column generation algorithm and then – since standard variable branching is difficult to integrate in the column generation procedure – devise our own branching scheme.

Column Generation

For now, we will leave the integrality constraint aside and just consider the relaxed problem. In our algorithm we will start with an RMP, which has a feasible solution. While we could easily find such a solution with heuristic methods, we introduce dummy variables δ_d in our demand constraint, with which we ensure feasibility even within the branch-and-bound process where heuristic methods simply might fail. Hence the demand constraints change to

$$[\pi_{d,s}] \qquad \sum_{p \in \mathbb{P}^w_{d,p}} f^w_{d,p} + \sum_{\substack{p \in \mathbb{P}^w_d: \\ s \cap p \neq \{\}}} \sum_{\substack{p' \in \mathbb{P}'_d \\ s \cap p = \{\}}} f^b_{d,p,p'} + \delta_d \geq 1 \qquad \forall d \in \mathbb{D}, \forall s \in \mathbb{S}, \tag{5.11}$$

the working path requirement to

$$[\tau^w_d] \qquad \sum_{p \in \mathbb{P}_d} f^w_{d,p} + \delta_d \geq 1 \qquad \forall d \in \mathbb{D}, \tag{5.12}$$

and our objective function to

$$\min \sum_{l \in \mathbb{L}^O} \sum_{t \in \mathbb{T}} n_{l,t} \cdot K_t + \sum_{d \in \mathbb{D}} \delta_d \cdot K_\delta. \tag{5.13}$$

The costs K_δ for the use of the dummy variables have to be chosen sufficiently large, so that they will be avoided in the optimal solution if at all possible.

With these modifications, we can safely assume that we have a feasible solution and thus have access to the dual variables denoted in angular brackets. Visualizing the structure of the coefficient matrix in a similar fashion as in Section 4.2.3, we observe the structure depicted in Table 5.1. We can see that the working path flow variables

	$f^w_{d_1,p}$			$f^w_{d_2,p}$			$f^w_{d_3,p}$		
	$\oplus$		$\oplus$						
		$\oplus$	$\oplus$						
	$\oplus$								
				$\oplus$		$\oplus$			
$\pi_{d,s}$				$\oplus$	$\oplus$				
					$\oplus$	$\oplus$			
							$\oplus$	$\oplus$	
							$\oplus$		$\oplus$
								$\oplus$	
		$-D_d$			$-D_d$				$-D_d$
		$-D_d$			$-D_d$				$-D_d$
		$-D_d$			$-D_d$				$-D_d$
	$-D_d$					$-D_d$		$-D_d$	
$\sigma_{l,s}$	$-D_d$					$-D_d$		$-D_d$	
	$-D_d$					$-D_d$		$-D_d$	
		$-D_d$	$-D_d$	$-D_d$			$-D_d$		$-D_d$
		$-D_d$	$-D_d$	$-D_d$			$-D_d$		$-D_d$
		$-D_d$	$-D_d$	$-D_d$			$-D_d$		$-D_d$

Table 5.1: Structure of the coefficient matix for working paths $f^w_{d,p}$

appear in those demand constraints, where they can be of use (i. e. are unaffected by a failure) and appear along the used edges in all network states[13]. Hence the dual constraints for a working path variable $f^w_{d,p}$ can consequently be written as

$$\sum_{\substack{s \in \mathbb{S}: \\ s \cap p = \{\}}} \pi_{d,s} - D_d \sum_{s \in \mathbb{S}} \sum_{l \in p} \sigma_{l,s} + \tau^w_d \leq 0. \tag{5.14}$$

[13] This last point illustrates the difference between stub release and our scenario: With stub release the paths would be torn down in those states where they are affected by an error, thereby creating extra capacity for backup paths.

As stated in the beginning, $\mathbb{S}$ consists of all single-link failures and the failure-free state s_0, we can rewrite the above equation to

$$\underbrace{\sum_{s\in\mathbb{S}} \pi_{d,s} + \tau_d^w}_{①} - \overbrace{\sum_{l\in p}\left(D_d \sum_{s\in\mathbb{S}} \sigma_{l,s} + \pi_{d,l}\right)}^{②} \leq 0. \tag{5.15}$$

The first two terms in this inequality (labeled ①) remain constant for all paths of given demand. Since all variables are valued positively, ① and ② are larger or equal to zero. Inspecting ②, we can see that a violation of the inequality is possible, *iff* ② < ①, which would cause the inequality to become larger than zero. As already described in 3.1.3f, finding violations (or cuts) in the dual is equivalent to finding improving variables in the primal system. Hence, our task is to identify paths, which would violate their dual constraint. Since ① remains constant and positive for every demand d, we have to examine ②. If we interpret the individual terms of the outer sum as link weights, we can solve the task given above as a shortest path problem: If the total cost of the shortest path with link weights $\sum_{s\in\mathbb{S}} \sigma_{e,s} + \pi_{d,l}$ is smaller than ①, we have identified a path which would violate its dual constraint and hence will improve the solution. If the total costs of the shortest path are larger than ①, we can be sure that no path would violate the constraint, i. e. we have all necessary paths in the solution base.

We have a corresponding situation for the backup paths with the differences being that a backup path will only be used if the working path fails and occupies bandwidth only in those cases, where it is used. The dual constraint for our $f^b_{d,p,p'}$ is

$$\sum_{l\in p} \pi_{d,l} - D_d \sum_{l\in p} \sum_{l'\in p'} \sigma_{l',l} - \tau_d^b \leq 0. \tag{5.16}$$

Again, we can express our task of finding violations as a shortest path problem with link weights $\sum_{l\in p} \sigma_{l',l}$. Should the shortest path be more expensive than $\sum_{l\in p} \pi_{d,l} - \tau_d^b$, we have included all variables in our solution base for d.

When we reconsider our task at hand, namely to identify path pairs, it could happen that a working path not yet in the solution base may be favorable because of a cheap backup path. However with the algorithm as it is, we only check for backup paths of working paths which are already in the solution base. In order to resolve this dilemma let us take a look again at the dual constraints for the working and the backup paths, namely

$$\sum_{\substack{s\in\mathbb{S}:\\ s\cap p=\{\}}} \pi_{d,s} - D_d \sum_{s\in\mathbb{S}} \sum_{l\in p} \sigma_{l,s} + \tau_d^w \leq 0 \tag{5.14}$$

and

$$\sum_{l\in p} \pi_{d,l} - D_d \sum_{l\in p} \sum_{l'\in p'} \sigma_{l',l} - \tau_d^b \leq 0. \tag{5.16}$$

Inspecting (5.16), we can see that in the best possible case, when all $\sigma_{l,s} = 0$, the left-handed side of the constraint will take a value of $\sum_{l\in p} \pi_{d,l} - \tau_d^b$. This gives us a bound on how "costly" the working path can be if the path-pair should still "pay off". We can see that the working path gains something in those cases, where it can be used, i. e. $\sum_{\substack{s\in\mathbb{S}:\\ s\cap p=\{\}}} \pi_{d,s}$. In all other cases the backup path has to take over. Hence the total gain of a path-pair is $\sum_{s\in\mathbb{S}} \pi_{d,s}$. If we now calculate the costs of a working path p as $\sum_{l\in p}\sum_{s\in\mathbb{S}} \sigma_{l,s}$, then the costs have to be smaller than $\sum_{s\in\mathbb{S}} \pi_{d,s} - \tau_d^b$. In this case, the new working path only pays off with a "free" backup path p', i. e. $\sum_{l\in p}\sum_{l'\in p'} \sigma_{l',l} = 0$.

This has the consequence, that we have to check for those paths within this bound as well in our pricing algorithm. Unfortunately this cannot be performed by a simple shortest path algorithm, since the path might already be in the solution base. Due to this, we use a k-shortest path algorithm which searches for all paths up to a length of $\sum_{s\in\mathbb{S}} \pi_{d,s} - \tau_d^b$. We check whether the resulting paths are already in the solution base and if not add them. Putting all those three pricing schemes together, we arrive at our final pricing algorithm, Algorithm 8.

Complexity Considerations In contrast to the first pricing problem which was $\mathcal{NP}$-hard, the pricing problem of our current formulation can be solved efficiently via Dijkstra's algorithm [Dij59] or even better algorithms as discussed in Section 3.3.1. It is important to note however, that we are not solving the same problem, that is, while the outcome of the MIP will be similar, the pricing problems and the linear relaxations are *not* comparable. Comparing the costs for the working path in (5.5) and (5.15), we can see that we underestimate the costs for the working paths, i. e. we will add working paths, which cannot form path pairs with negative reduced costs. In other words, our strategy shifts complexity from pricing problem into the MIP, i. e. we generate more variables and the optimization process has to figure out the best couplings (which working path with which backup path), while this was already done in path-pairs of the classical formulation. For an overview of complexity issues in network optimization problems we refer to [OP08] for continuous problems and [TPŻ09] for discrete cases.

Branching Scheme

Up to now, we have only considered the linear relaxation of our original problem. Consequently usually some path-flow variables will have fractional values, i. e. the demand is split-up between several paths. In those cases, simple rounding methods might generate feasible integer solutions, however neither can be guaranteed and even if they are successful, the question of optimality or remaining solution gaps are still open as explained previously.

The standard approach to obtaining optimal integer solutions would be to branch on fractional variables as explained in Section 3.2.2. Using this methodology, we would branch on $f_{d,p}^w$, $f_{d,p,p'}^b$, and $n_{l,t}$ variables respectively. For example, if we have a variable $\tilde{f}_{d,p}^w$ which is fractional (i. e. $0 < \tilde{f}_{d,p}^w < 1$), we will generate two nodes in a branch-and-bound tree, in which we fix the variable to zero and one respectively and resolve the

Algorithm 8: Pricing Algorithm for SBPP (linear relaxation)

Data: $\mathcal{F}$, $\mathcal{D}$, π, τ, σ
Result: $\mathbb{P}$, $\mathbb{P}'$

```
forall d ∈ 𝔻 do
    begin
        Find improving backup paths
    end
    foundBP ⟵ FALSE;
    forall p ∈ ℙ_d do
        begin Precompute link-weights
            forall l' ∈ 𝕃^O do
                w_{l'} ⟵ D_d · ∑_{l∈p} σ_{l',l};
        end
        ShortestPath( d, 𝕃^O, w, p', c' );
        if c' < ∑_{l∈p} π_{d,l} − τ_d^b then
            AddBackupPathToPathSet( p', p, ℙ'_d );
            foundBP ⟵ TRUE;
    if !foundBP then
        begin Find improving working paths
            begin Precompute link-weights
                forall e ∈ 𝓕 do
                    w_l ⟵ D_d · ∑_{s∈𝕊} σ_{l,s} + π_{d,l};
            end
            ShortestPath( d, 𝕃^O, w, p, c );
            if c < ∑_{s∈𝕊} π_{d,s} + τ_d then
                AddWorkingPathToPathSet( p, ℙ_d );
                foundWP ⟵ TRUE;
            if !foundWP then
                begin Find improving working paths for pairs
                    forall e ∈ 𝓕 do
                        w_l ⟵ D_d · ∑_{s∈𝕊} σ_{l,s};
                    maxLength ⟵ ∑_{s∈𝕊} π_{d,s} + τ_d^w + τ_d^b;
                    kShortestPath( d, 𝕃^O, w, ℙ_Cand, c );
                    forall p ∈ ℙ_Cand do
                        if !InSolutionBase( d, p ) then
                            AddWorkingPathToPathSet( p, ℙ_d );
                end
        end
```

resulting LPs. While being conveniently available in MIP-solvers, like CPLEX [CPL09] and SCIP [Ach07], this approach suffers from two drawbacks:

- Even in a non-column-generation environment, branching on path variables is known to show very weak performance. More specifically as already reported by Ryan and Foster in [RF81], the decision not to use a given path (i. e. forcing $\tilde{f}^w_{d,p} = 0$) is a weak branching decision, because it can be 'circumvented' by using a slightly different path. Even more so, once we consider discrete edge capacities, where an edge could be filled up by another portion of another demand. On the other hand, The compliment (i. e. forcing $\tilde{f}^w_{d,p} = 1$) tends to induce large changes in the MIP, causing "an unbalanced growth of the branch and bound tree" [RF81, p. 279], which is "undesirable" [Wol98, p. 194].
- It is quite hard to take the branching decisions into account during the pricing problem: If we force $\tilde{f}^w_{d,p} = 0$, an unmodified pricer will simply find the same path (and with this the same variable) again. If we want to avoid this, we have to inspect all paths which would violate their dual constraints, which could be achieved by more sophisticated algorithms (as for example a modification of an algorithm proposed by Hershberger, Maxel and Suri [HMS03]), however implementation seems rather difficult.

We propose a method based on a modified form of the generalized Ryan-Foster branching scheme [RF81] introduced by Barnhart et al. in [BJN+98] and refined by Barnhart, Hane and Vance in [BHV00].

Generalized Ryan-Foster Branching Closely inspecting (5.7), we can easily see that the following proposition holds for the working path variables $f^w_{d,p}$:

> If there is one fractional f^w_{d,p_1}, i. e. $0 < f^w_{d,p_1} < 1$, then there has to be at least one other f^w_{d,p_2}, such that $0 < f^w_{d,p_2} < 1$.

Furthermore, we can be sure by the way we constructed our MIP, that $p_1 \neq p_2$ and that they are loop-free, thus there has to be at least one edge l_1 on path p_1, which is not part of path p_2 and one edge $l_2 \in p_2 : l_2 \notin p_1$. In order to make use of this property, we need to find the *divergence node* v' of p_1 and p_2, that is, the last common node of these two paths starting from the source of d. By definition, the outgoing flows of p_1 and p_2 will use different edges. In our branch-and-bound tree, we create two nodes n_1 and n_2, where we will forbid the use of either l_1 or l_2 for outgoing flows from v'. We realize this by converting l_1 and l_2 to arc-pairs and force the outgoing flows ($l_1^{v'}$ and $l_2^{v'}$) to zero:

$$\sum_{\substack{p \in \mathbb{P}_d: \\ l_1^{v'} \in p}} f^w_{d,p} \leq 0 \qquad \text{to } n_1, \tag{5.17a}$$

and

$$\sum_{\substack{p \in \mathbb{P}_d: \\ l_2^{v'} \in p}} f^w_{d,p} \leq 0 \qquad \text{to } n_2, \tag{5.17b}$$

respectively. It is especially worth noting that we "branch down" in both nodes, which will greatly simplify the integration in our pricing algorithm. It is quite obvious, how to extend this branching scheme to backup flow variables as well, which we show in the resulting branch-and-bound algorithm, Algorithm 9.

This branching scheme is correct, because we can always find a pair of fractional working or backup paths in a fractional solution and the number of branches is finite due to the finite number of fibre edges and demands.

Integrating the Branching Decisions in the Pricing Algorithm As stated before, we only "branch down", i. e. we fix the flow over some edges to zero for a given demand and thus render the current fractional solution infeasible. We can therefor easily interpret a branching decisions as forbidding the use of a certain edge (or rather edges deeper in the branch-and-bound tree) for a demand. Consequently, in order to take these branching decisions into account, we have to make sure that the pricing algorithm will not find paths, which use these "forbidden" arcs. This can be easily achieved either by setting the arc weight to a high number, or by hiding these arcs from the graph altogether (which is what we do).

During the construction of the branch-and-bound scheme, we have claimed, that branching down simplifies the integration in the pricing algorithm, which we now want to illustrate. Assume that we have found an edge l_1 which is part of path p_1, but not of p_2. Instead of branching down on two edges we could also create two nodes with the following two constraints:

$$\sum_{\substack{p\in\mathbb{P}_d:\\ l_1\in p}} f^w_{d,p} \leq 0 \tag{5.18a}$$

$$[\beta^w_{d,p}] \qquad \sum_{\substack{p\in\mathbb{P}_d:\\ l_1\in p}} f^w_{d,p} \geq 1. \tag{5.18b}$$

In the spirit of variable dichotomy. (5.18a) is identical to (5.17a) and can thus be treated equally in our pricing by hiding the relevant edges. (5.18b) however is much harder to integrate:

- Forcing the use of a given edge can be efficiently performed in case of one single edge (one shortest path from the source of the demand to the edge and one shortest path from the edge to the target of the demand), but not for multiple edges [BHV00, p. 321].
- Simply taking the associated dual variable into account is not free of problems either. The dual costs of a path p would have to be calculated as

$$\sum_{s\in\mathbb{S}} \pi_{d,s} + \tau^w_d - D_d \sum_{l\in p} \left(\sum_{s\in\mathbb{S}} \sigma_{l,s} + \pi_{d,l} + \beta^w_{d,l} \right) \leq 0, \tag{5.19}$$

  however with $\beta^w_{d,p} \leq 0$ (as following from (5.18b)), we cannot make sure that the link-weights $\sum_{s\in\mathbb{S}} \sigma_{l,s} + \pi_{d,l} + \beta^w_{d,l}$ are positive. While this alone could be handled by the Bellman-Ford algorithm [Bel58], [CLR90, p. 532] or a modified Dijkstra's algorithm [Bha94], we cannot guarantee that our problem is free of negative cycles[14]. Without this guarantee, the related shortest path problem (i. e. find the shortest path while visiting every edge at most once) becomes $\mathcal{NP}$-hard [Pet08, p. 29].

5.2.4 Case Study

Implementation

In order to evaluate SBPP and the efficiency of our optimization approach, we implemented the MIP as well as the branching and pricing algorithms in C++, SCIP Version 1.1 [Ach07] with CPLEX Version 9.13 [CPL09], and GRAPH Version 3.01 [Lay09]. The implementation is non-trivial as a lot of otherwise unnoticed details can influence both correctness and efficiency of column generation algorithms in general and branch-and-price algorithms in particular. In order to shorten the debugging phase of future research efforts in this area, we want to highlight the following pitfalls:

- In order to work correctly *all* bounds on the variables have to be exactly as in the description. Numerical issues can arise from this point, i. e. when a variable range is effectively limited.
- Dual variables can become slightly negative (within the numerical precision of the solver) despite being larger or equal to zero by definition, when they go towards zero or take very small positive values. Shortest path algorithms need to have the same precision in the pricing algorithm, otherwise paths can easily be "refound".
- We found it worth the effort to check for duplicate paths before adding them, since in the trailing-off phase, the dual variables will be at the border of the numerical precision of the used routines.
- Presolvers and separators can change the pricing problem. For this reason we deactivated both, presolvers and separators when running our algorithm.

Computational Efficiency

We used numerous reference networks from the SNDLib to find the strengths and weaknesses of our branch-and-bound approach. We depict in Table 5.2 the gaps after a time-limit of one hour of running time[15] of our branch-and-price approach and the

[14]Orlowski reports in [Orl09, p. 61f], that in a seemingly at least comparable pricing problem in the recent literature (work by Raghavan and Stanojević which we were unable to obtain), the authors did not experience any negative cycles, however they could neither offer an explanation nor a proof for this.

[15]Not taking the cosiderable precomputation time for the path-set in the brute-force approach into account.

standard "brute force" approach of generating all possible paths[16]. The networks were chosen to illustrate the limits of the algorithms involved.

Network	Branch-and-Price	Brute Force
Nobel US	6.07%	11.43%
Germany 17	2.63%	7.63%
COST 239	13.01%	—
Germany 50	10.63%	—
Norway	1.65%	—

Table 5.2: Remaining gaps after 1h (top) and 2h (bottom) of computation time.

In case of the Nobel US network, both algorithms managed to solve the MIP to relatively good gaps. Due to the sparse topology not too many path pairs exist and hence memory consumption of the brute-force algorithm is around 1 GByte of memory which is around five times more than the branch-and-price counterpart. The achieved gap is roughly half as large as the brute-force approach. Switching our attention to the Nobel Germany network, we observe a similar behaviour, with the difference between both algorithms becoming even larger: the achieved gap was roughly one third with a memory consumption which was around one tenth of the brute-force algorithm. In the COST 239 network, we observe an even larger difference: We are not even able to construct the MIP for the brute force-algorithm because 16 GByte of memory are not sufficient for this. While the resulting bound is far from being spectacularly good, it is a clear improvement over having no bound at all.

Interestingly enough, our branch-and-price algorithm scales even further to the 27 node Norway and the Germany 50 network, where we can calculate a solution with 1% and 10% gaps respectively within 2h. Again, we were unable to just create the full model within 16 GByte of main memory.

Evaluation of SBPP

Apart from the pure usefulness of having a planning algorithm for SBPP, we want to compare the bandwidth efficieny of SBPP to PD, which is among the most bandwidth efficient realistic protection methods *without* sharing (and as argued in the resilience overview certainly more bandwidth efficent than 1+1 protection). We depict in Figure 5.3 the ratio of the capacity required with the respective protection scheme and the necessary capacity without any protection at all.

We can easily see, that PD requires at least 248% of the bandwidth for the unprotected case on top in the best case (Norway). SBPP in contrast requires 172% in the worst case (Nobel US). PD thus requires at least 1.48 times the capacity of SBPP. Quite unsurprisingly, a higher node degree helps to reduce the bandwidth requirements of

[16] All presolver and separators were activated in this case.

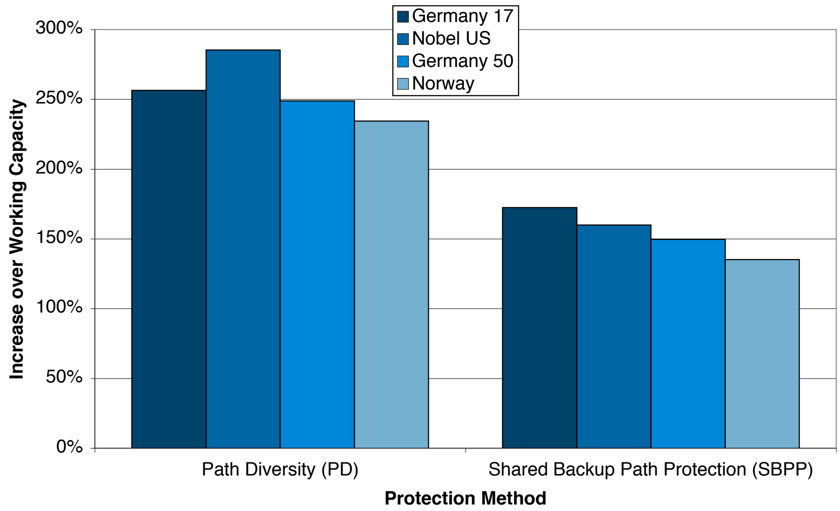


Figure 5.3: Required capacity of PD and SBPP in relation to the working capacity

both mechanisms. The difference between the relative position of PD and SBPP for the Nobel US network can be explained by the fact, that the Nobel US network has a higher load than the Germany 17 network forcing longer paths in the more bandwidth requiring PD case.

We can thus conclude that as suggested by previous research, backup capacity sharing causes a significantly lower bandwidth consumption, even if we do not allow flow bifurcation in the network, which we could prove with the presented planning methodology for the first time to the best of the author's knowledge.

Algorithm 9: Branch-and-Bound Algorithm

Data: $\mathcal{F}$, $\mathcal{D}$, $\mathbb{P}$, $\mathbb{P}'$, $f^w_{d,p}$, $f^b_{d,p,p'}$

begin Branch on fractional working path variables
    forall $d \in \mathbb{D} \wedge$ *!foundfractional* **do**
        forall $p \in \mathbb{P}_d \wedge$ *!foundfractional* **do**
            if $0 < f^w_{d,p} < 1 \wedge$ *!foundfractional* **then**
                $p_1 \longleftarrow p$;
                forall $q \in \mathbb{P}_d \wedge$ *!foundfractional* **do**
                    if $0 < f^w_{d,q} < 1 \wedge p \neq q$ **then**
                        $p_2 \longleftarrow q$;
                        foundfractional $\longleftarrow$ TRUE;
    if *foundfractional* **then**
        $l_1^{v'} \longleftarrow$ UncommonArc(p_1, p_2);
        $l_2^{v'} \longleftarrow$ UncommonArc(p_2, p_1);
        $n_1 \longleftarrow$ CreateNode();
        AddConstraintToNode($\sum_{\substack{p \in \mathbb{P}_d: \\ l_1^{v'} \in p}} f^w_{d,p}$, n_1);
        $n_2 \longleftarrow$ CreateNode();
        AddConstraintToNode($\sum_{\substack{p \in \mathbb{P}_d: \\ l_2^{v'} \in p}} f^w_{d,p}$, n_2);
end
begin Branch on fractional backup path variables
    if *!foundfractional* **then**
        forall $d \in \mathbb{D} \wedge$ *!foundfractional* **do**
            forall $p \in \mathbb{P}_d \wedge$ *!foundfractional* **do**
                forall $p' \in \mathbb{P}'_{d,p}$ **do**
                    if $0 < f^b_{d,p,p'} < 1 \wedge$ *!foundfractional* **then**
                        $p_1 \longleftarrow p$;
                        forall $q' \in \mathbb{P}_{d,p} \wedge$ *!foundfractional* **do**
                            if $0 < f^b_{d,p,q} < 1 \wedge p' \neq q'$ **then**
                                $p_2 \longleftarrow q'$;
                                foundfractional $\longleftarrow$ TRUE;
        if *foundfractional* **then**
            $l_1^{v'} \longleftarrow$ UncommonArc(p_1, p_2);
            $l_2^{v'} \longleftarrow$ UncommonArc(p_2, p_1);
            $n_1 \longleftarrow$ CreateNode();
            AddConstraintToNode($\sum_{\substack{p' \in \mathbb{P}'_{d,p}: \\ l_1^{v'} \in p'}} f^b_{d,p,p'}$, n_1);
            $n_2 \longleftarrow$ CreateNode();
            AddConstraintToNode($\sum_{\substack{p' \in \mathbb{P}'_{d,p}: \\ l_2^{v'} \in p}} f^b_{d,p,p'}$, n_2);
end

5.3 Dual-Homing Protection

Up to now, we solely restricted our view to the backbone network. Due to two plausible reasons, which we will demonstrate further on, we will augment this perspective with knowledge about the connectivity in the neighboring horizontal layer – the access- or metro-network.

In the previous sections, we assumed traffic to exist exclusively between core nodes without paying further attention as to how this traffic actually is generated. Judging from market studies, such as [Cis09], a large portion of the world-wide IP-traffic is simply "Internet traffic". In many cases, even smaller ISPs are able to provide highly resilient Internet connectivity. Tier-1 providers such as AT&T [ATT09] and Verizon [Ver06] offer a multitude of resiliency options to connect a local ISP to their backbone network up to a fully redundant connection, consisting of two disjoint connections to different backbone nodes[17]. Let us take a look at the node `Essen` in the Germany17 network. In a double failure scenario, with both links connecting this node failing, the standard assumption would be, that all traffic originating at this node will not enter the network anymore. If however, customers had a dual-homing solution (connecting for example additionally to `Dortmund`), they would still be connected to the backbone and thus traffic formerly originating at `Essen` would originate at `Dortmund` which is quite different to simply vanishing.

The second reason, why we will incoroporate this property of dual-homing, will be demonstrated in the case study in the remainder of this section: The additional degree of freedom can be used to provide protection against dual-link failures in a comparably cost-effective manner.

5.3.1 Related Work

Neither the idea of dual-homing nor dual-link failure resiliency are completely new. In a line of work started by Wang, Vokkarane, Jue et al. , the authors try to asses the cost-efficiency of dual-homing for single-link failures in the backbone-network networks [VWQ+04, WVQJ04, VWJ05, WVJ+05]. To this end, they develop MIP-models with which they examine the bandwidth-efficiency of randomly generated network instances. We find this rather hard to interpret due to their artificial structure[18]. Furthermore they offer heuristic algorithms for a dynamic setting. In a similarly artificial model, Sasaki and Rozic claim a 32% cost decrease in a shared dual-homing solution compared to a shared protection solution using a MIP-model in [SR08], which is a similar result compared to earlier work by the same author [SS02, SS03].

Designing suitable topologies including dual-homing was introduced by Kim, Chung and Tcha in [KCT95], whose work is founded on the terminal-concentrator assignment

[17] A variation of this scheme is to run two identical networks (called A-plane and B-plane, usually with hardware from different vendors) and connect customers to both of them.

[18] extremely high node-degree of up to 20, uniform demands in wavelength granularity, a very sparse demand matrix (32 connections in a 50 node network), no cost-model

problem by Pirkul, Narasimhan and De [PND88]. Since data on access network structures and traffic models for metro-traffic were not available neither within EIBONE nor in more public locations (as for example in the SNDLib) only a small contribution was made towards this direction: In [SKS07], we present models and optimization algorithms to find suitable regions for metro-networks, minimizing the worst-case cost for inner-regional fibre topology.

Schupke, Autenrieth and Fischer highlighted in [SAF01] the importance of considering multiple failures in large national (i. e. US) and international backbones (pan-European networks), even if the Mean Time to Repair (MTTR) is quite low (in the range of 5h). Clouqueur and Grover further investigate the capacity overhead for dual-link failure protection and use an Integer Linear Program (ILP)-model with a heavily restricted path-set for backup-path optimization in [CG02]. A noteworthy observation in their work is that not all demands are equally expensive, i. e. they show that with a rather moderate increase of the backup capacity budget a considerable number of demands can be protected, thereby strengthening a result by Schupke and Prinz [SP03]. Ramasubramanian and Chandak study link protection schemes for dual-failure protection [RC08] and proove that a three-edge-connected graph is a sufficient condition to allow a solution dual-failure link protection. With an ILP-formulation, they observe that for dual-link failure protection roughly 200% more capacity is required in their artificial and real-life network instances however, they neither employ real-life traffic-data nor cost-models. Interestingly enough Ma, Fayek and Ho come to the same conclusion with a far more realistic model in [MFH08] and show that their LP-model offers a good approximation of the MIP-results. Prinz, Autenrieth and Schupke examine dual-link failure protection in a multi-layer and multi-period scenario in [PAS05] and state that sharing of the second backup-path (which is obviously necessary) can reduce network costs by 20% compared to the non-shared case, however they do not state the protection overhead.

5.3.2 Network Model

Architecture

In order to keep our model computationally feasible and simple enough to isolate the influence of dual-homing, we will start with a rather abstract network model, which reflects the bandwidth-efficiency only and which we will extend to the basic IP over opaque-DWDM architecture presented in Section 3.3.3 with transponder costs taken from the Nobel2 project [HGMS08].

Resilience Strategies

Naturally, at least the same degrees of freedom exist for dual-link failure resilience as for the single-link case. In our studies we will focus on path-based protection/restoration methods, which are failure-independent, leaving the question of shared- or non-shared backup-paths, which we will thus evaluate both. In the remainder of this section, we will use the following naming convention:

- Three link-disjoint paths (i. e. without any sharing) without dual homing will be called *1+1+1I* protection, whereby the "I" indicates that we will ignore traffic for node-pairs for which three edge-disjoint paths cannot be established.
- Similarly, two link-disjoint paths with one additional shareable backup-path will be named *1+1:1I*.
- Three link-disjoint paths with dual-homing, will be called *1+1+1DH*. Two paths have to start and end at the original source and target destinations of the demand, while the third can originate and end at other nodes. Evaluation of the influence of the choice of these nodes will be part of our case study.
- Naturally, sharing is a possibility in the dual-homing scenarios as well, which leads to *1+1:1DH* protection, i. e. the second backup path between the DH-nodes can be shared and the original nodes is shared.

5.3.3 Optimization Model

We will build or model upon the basic model for an opaque network presented in Section 3.3.3. Our first task is to compute suitable path-sets $p = (p_1, p_2, p_3)$ for the desired protection scheme and consequently construct suitable demand constraints and relations to the used capacity on a link. Our demand constraint is always the task to select a sufficient number of path triples p to satisify the demand d:

$$\sum_{p \in \mathbb{P}_d} f_{d,p} \geq 1 \qquad \forall d \in \mathbb{D}' \tag{5.20}$$

Stepping through the strategies outlined above, we find:

- 1+1+1I
  This case is straightforward: We check whether three disjoint paths exist for a given demand d at all (we denote the set of these demands as $\mathbb{D}'$)and if so, construct all of these path triples $p \in \mathbb{P}_d$.
  Since we do not allow sharing, we can easily gather the necessary capacity on a link l by summing over all flow-variables using that edge, i. e. those path triples where l is part of any of p_1, p_2 or p_3.

$$D_d \sum_{\substack{p \in \mathbb{P}_d: \\ l \in p}} f_{d,p} \leq u_l \qquad \forall l \in \mathbb{L}^O \tag{5.21}$$

- 1+1:1I
  Compared to the previous scheme, we are now allowed to share the bandwidth occupied by the third path p_3, which forces us to separately consider every failure state $s \in \mathbb{S}$. We construct our path triple p again to consist of three link-disjoint

paths for those demands $d \in \mathbb{D}'$ where such a tuple exists and ignoring the rest.

$$\sum_{d \in \mathbb{D}} D_d \left(\sum_{\substack{p \in \mathbb{P}_d: \\ l \in p_1 \vee \\ l \in p_2}} f_{d,p} + \sum_{\substack{s \cap p_1 \neq \{\} \wedge \\ s \cap p_2 \neq \{\} \wedge \\ l \in p_3}} f_{d,p} \right) \leq u_l \qquad \forall s \in \mathbb{S}, \forall l \in \mathbb{L}^O \tag{5.22}$$

- 1+1+1DH
  In this case, we generate our path triples depending on whether three disjoint paths exist for a demand d or not. If this is the case, we generate all link-disjoint path triples p as previously. If the reason for this is because of a two-connected source node, we search for a suitable dual-homing node that is neither the original source node (and not the target-node), but allows the construction of a third link-disjoint path with respect to p_1 and p_2. If this is the case because of the target node, we select an alternative target node following the same principle. Should the two-connectedness result from both source and target, we will do both. In all networks examined in the case study later on, we were able to find path triples which allowed protection for all paths. Hence the set of demands $\mathbb{D}'$ which can be protected is equivalent to $\mathbb{D}$.
  Since all difference to 1+1+1I lies within the construction of the path-set p, we can use the same capacity constraint, namely

$$D_d \sum_{\substack{p \in \mathbb{P}_d: \\ l \in p}} f_{d,p} \leq u_l \qquad \forall l \in \mathbb{L}^O. \tag{5.23}$$

- 1+1:1DH
  Finally our last scheme is a combination of the path tuple generation of the 1+1+1DH scheme and the capacity constraint (5.22) from 1+1:1I.

We have to highlight at this point that the set of demands can be vastly different between the resilience schemes, depending very much on the network structure, however it is quite hard to fairly account for demands that cannot be protected. A weak point of the model above is that we do not take the structure of the underlying metro network into account to define our notion of “close” — which is owed to the fact that we do not have metro *and* matching core networks available.

Our objective is either to minimize the used capacity of the network, i. e.

$$\min \sum_{l \in \mathbb{L}^O} u_l \tag{5.24}$$

or to minimize the CAPEX by minimizing the costs caused by the transponders with transponder prices taken from [HGMS08] by using the underlying model presented in 3.3.3.

5.3.4 Case Study

We implemented the above MIP in SCIP Version 1.1 [Ach07] with CPLEX 9.13 [CPL09] using GRAPH 3.01 [Lay09] for data representation and shortest path algorithms. We tested our proposed approaches within two networks, the Nobel US and the Germany 17 network from the SNDLib [OPTW07]. In our first study we want to examine the

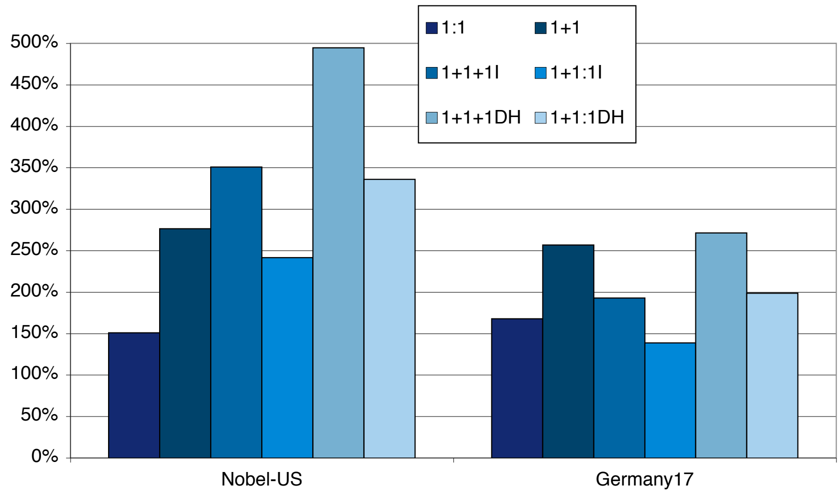


Figure 5.4: Required capacity in relation to the working capacity

bandwidth efficiency of our approaches, i. e. we minimized only the used capacity in the network without any granularities caused for example by transponders. In Figure 5.4, we compare the ratio of the capacity required by the protection mechanism compared and the capacity required without any protection. The dual homing node in the case above was chosen to be the closest node to the original location(s).

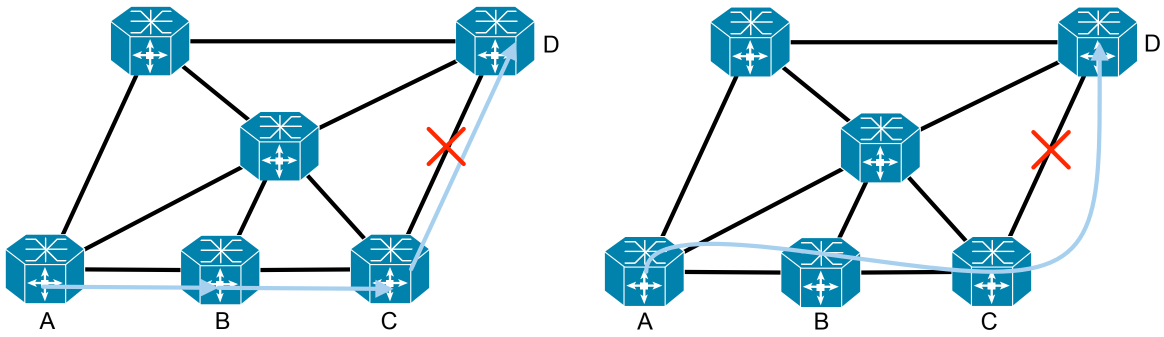
Inspecting the values for the Nobel US network, we can see that three disjoint paths for 1+1+1DH protection cost almost five times as much bandwidth as for the original network. In case of a shared third path, this factor decreases signifcantly to less than 3.5. It is important to realize that 1+1+1I and 1+1:1I cannot protect a share of more than 25% of the demand volume in the network (which is hence ignored within the optimization). It is interesting however that with this rather simple method for determining the second home, we can protect *all* traffic against dual-link failures. Furthermore, if we allow sharing of the third path, our required bandwidth drops below 1+1+1I with the incentive that *all* paths are protected now.

Within the Germany 17 network, this seems to be different at first glance, however the portion of the demand volume that cannot be protected by the single-homing schemes amounts to more than 54%. With dual-homeing the situation improves somewhat, but is far from being satisfactory: We still cannot protect 38% of the demand volume. This is due to the chains of only two-connected nodes for example in the Rhine-Ruhr area. Hence we change our algorithm to find the second home: We allow to search for suitable nodes in a range around the original host and take the closest suitable node. Due to our condition that the alternate source node may not be identical with the target node and vice-versa, we need to have a rather large range of up to 250 km. With this, 1+1+1DH protection would require 474% of the original working capacity which is quite comparable to the Nobel US case. Unfortunately, we cannot provide any useable lower bound on 1+1:1DH, because we were unable to solve the full MIP, however the results we have (gained with a limited path-set) are comparable to those for the Nobel US network.

In order to rule out a large influence due to transponder granularities, we performed the same planning with transponder prices and data from the Nobel2 cost-model [HGMS08]. Generally speaking, we only found smaller differences to the capacity minimization, with the exception for 1+1+1DH in the Nobel US network, where our problem turned out to be infeasible. This is mainly caused by range limitations on the transponders, the crucial point being that there is only a 10 Gbit/s transponder in ultra-long range category. Interpreting the results from a customer perspective (dividing the total costs by the total demand in the network which can be protected), one 1 Gbit/s in the 1+1 protected case costs about 1.30 cost units and in the 1+1:1DH case 1.57. $1 + 1 : 1I$ would cost (quite unsurprisingly) the same as 1+1:1DH. Hence we can conclude that the cost of providing an additional shared protection path from the second home in an already 1+1 protected network would increase the total costs by roughly 25%, which is quite a moderate margin compared to more than 76% for a third static path.

5.4 Failure Localization

The previous chapter highlighted the cost effectiveness of transparent and translucent networks even with advanced transmission techniques. In all these networks we can apply a number of protection methods depending on the capabilities of the equipment. In case of a failure and a successful switchover to backup capacities, the subsequent step is to localize and repair the failure. While the employed technology between a transparent and opaque network is not necessarily considerably different, the information provided about a failure might differ significantly. For example, consider the scenario of a transparent and an opaque link depicted in Figure 5.5. In the opaque case in Figure 5.5a, we have the information that the fibre cut is between nodes C and D, since nodes C and D terminate the link and will thus notice the cut. In the transparent case of Figure 5.5b, only nodes A and E terminate the link, while it passes through all intermediate nodes without "being touched". As a consequence in case of an error, we only have the information, that the error occured between node A and E. Thus, failure localization is more difficult in transparent or translucent networks. The same is true for so-called *soft errors* as for example fibre bending or jamming due to malfunctions. Since we perform full O/E/O-conversion at every node in an opaque network, we gather plenty of information about the signal quality at every node. In the transparent case, we only get information at the end of the light-path again. Localizing multiple softer errors which add up along the light-path will be tedious work. One proposed solution for this problem is to reserve wavelengths for monitoring pur-



(a) Opaque (b) Transparent

Figure 5.5: Failure States

poses and send test signals using these wavelengths along pre-planned paths, such as the so-called *m-cycles* by Zeng et al. in [ZHV04] or spanning trees again by Zeng et al. in [ZHV05]. The drawback of this solution is that a considerable number of wavelengths are necessary and those wavelengths cannot be used for customer traffic any more. In our approach however, we use optical splitters to branch off a small amount of the transmitted signal (usually 1%) and use this portion for monitoring. Similar problems have been studied in different contexts for example in IP networks [BDG04]. In the case of the latter, the essential problem is scalability. All IP nodes offer monitoring capabilities, however activating monitoring generates an overwhelming amount of information. Duffield et al. investigate the information-theoretical aspects of this problem in [DLT05]. Nevertheless due to the limited number of nodes in the network and the high costs of monitoring equipment our focus lies somewhere else: We want to determine the best placement of a limited amount of monitoring equipment for a given network.

5.4.1 Simple Monitoring

Scenario

For the sake of simplicity, we will introduce our problem in its purest form and develop numerous enhancements based on this setting. We consider transparent or translucent networks with a set of admissible light-paths $\mathbb{L}^A$. This path set can be freely defined, following physical and political constraints of the network operator. At the source and target of these light-paths, the connection is terminated, while they pass untouched through intermediate nodes. However, we do have the possibility to install an optical splitter and monitoring equipment at the nodes, which allows us to examine all light-paths passing through the respective node, such as an Optical Spectrum Analyzer (OSA). Our metric to evaluate an equipment placing is the length of the longest unmonitored path in $\mathbb{L}^A$.

Problem Formulation

We model the underlying fibre topology as an undirected graph $\mathcal{F}(\mathbb{V}, \mathbb{F})$, with $\mathbb{V}$ being the set of nodes in our network and $\mathbb{F}$ being the set of fibres. The binary variable s_v for every node $v \in \mathbb{V}$, shall be

$$s_v = \begin{cases} 1 \textit{ iff } \text{monitoring equipment is placed at node } v \text{ and} \\ 0 \text{ else.} \end{cases} \tag{5.25}$$

The path set $\mathbb{L}$ contains all admissible light-paths $\mathbb{L}^A$ *and* all sub-paths, e. g. for the path $A-B-C-D$ from $\mathbb{L}^A$, we also have to include the sub-paths $A-B$, $A-B-C$, $B-C$, $B-C-D$, and $C-D$. We denote the physical length of a path p by L_p, which allows us to determine the length $l \in \mathbb{R}_0^+$ of the longest unmonitored path p via the following set of constraints

$$\left(1 - \sum_{v \in p} s_v\right) L_p \leq l \qquad \forall p \in \mathbb{L}. \tag{5.26}$$

If none of the nodes $v \in p$ along path p has monitoring equipment installed, the term $1 - \sum_{v \in p} s_v$ will be 1. Consequently, the continuous variable l has to be chosen as large as the path length L_p at least. *Iff* however one node is monitored at least, the sum will be less or equal to zero, hence l can be left zero. Quite obviously, we can only monitor light-paths not originating at node v. It will not it help either if we monitor the path at the target node, therefore we exclude the end nodes v_p^s and v_p^t from our sum

$$\left(1 - \sum_{v \in p \setminus \left\{v_p^s, v_p^t\right\}} s_v\right) L_p \leq l \qquad \forall p \in \mathbb{L}. \tag{5.27}$$

Objective Functions

The above formulation enables us to optimize the monitor placement from two perspectives.

Placing Limited Monitoring Equipment First, it is evident, that we can limit the number of monitors to $S \in \mathbb{N}^+$ by adding the constraint

$$\sum_{v \in \mathbb{V}} s_v \leq S. \tag{5.28a}$$

Based on this budget restriction, we can minimize the length of the longest unmonitored path l

$$\min l. \tag{5.28b}$$

Solving the resulting MIP will gain the optimal placement of the available monitoring equipment.

Minimizing the Number of Necessary Monitors Second, we can use the basis above to find the cheapest placement of monitors to keep the longest unmonitored light-path below a desired limit L by adding the constraint

$$l \leq L^{19}. \tag{5.29a}$$

Subsequently we minimize the number of installed monitors

$$\min \sum_{v \in \mathbb{V}} s_v. \tag{5.29b}$$

It is quite obvious, that *upgrade problems* (i. e. monitors have already been installed for nodes $\mathbb{V}_S \subset \mathbb{V}$), can easily be considered by assigning monitor variables $s_v = 1 \quad \forall v \in \mathbb{V}_S$.

From a mathematical point of view, our MIP formulation is related to the classic *flow interception problem* introduced by Hakimi in [Hak64, Hak65] which found considerable use in facility location problems as in [BLF92]. In a more network related context, Scheffel used a similar approach for regenerator allocation in [Sch05].

Simulation Performance

Our MIP formulation consists of $|\mathbb{V}|$ binary variables s_v, one additional continuous variable l in case of (5.28) and $|\mathbb{L}|+1$ constraints in case of (5.28), and $|\mathbb{L}|$ constraints in case of (5.29). While the number of variables is well within practicable limitations in realistic problem instances, the huge number of constraints greatly affects our solving time. For the first objective function (5.28), we can reduce the number of paths we need to take into account using the following argument: In a network with $|\mathbb{V}|$ nodes, and S monitors, the longest unmonitored path can only be $|\mathbb{V}| - S + 1$ hops long, for the worst case, with a monitor located at both end nodes and all unmonitored nodes being on the path. Hence we can reduce our path-set P to paths being at most $|\mathbb{V}| - S + 1$ hops long, which we denote by $\mathbb{L}_{|\mathbb{V}|-S+1}$.

For the second case (5.29), we can perform a worst case estimation, by searching for the hop wise longest path with a physical path length of at most L. Assuming a number of H hops, we acquire a lower bound of $\min \sum_{v \in \mathbb{V}} s_v \geq |\mathbb{V}| - H - 1$ and can reduce the number of paths following the same argument as above.

[19] Quite obviously, the variable l is unnecessary in this formulation as we can reformulate the constraints easily to

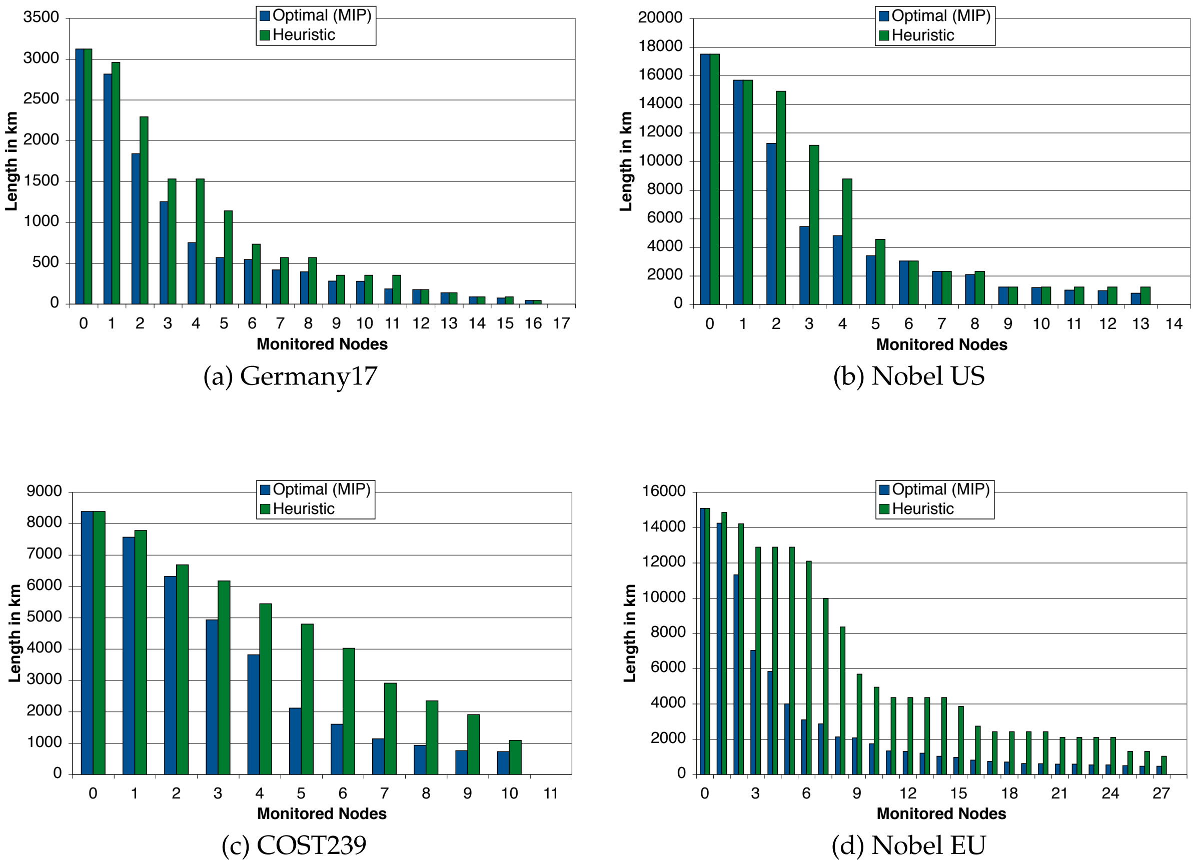
$$\left(1 - \sum_{v \in p \setminus \{v_p^s\}} s_v\right) L_p \leq L \qquad \forall p \in \mathbb{L}.$$

Case Study

We performed a monitor placement for four national and international backbone topologies, allowing from zero to $|\mathbb{V}|$ monitors via our optimization approach and a naive heuristic in which we assign monitors according to the following criteria:

1. Assign monitoring equipment to those M nodes with the highest node degree.
2. If there are several nodes with the same node degree, monitoring equipment will be assigned to the node with the highest average length of adjacent edges.

In order to rule out the influence of routing schemes, we considered a full path set, i. e. $\mathbb{P}$ contained all cycle-free paths in the network. The running times varied greatly, as a consequence of the different network sizes and average node degrees, from minutes to roughly one week in an implementation with CPLEX 9.1 on a 2.4 GHz workstation.



(a) Germany17 (b) Nobel US

(c) COST239 (d) Nobel EU

Figure 5.5: Length of the longest unmonitored light-path l vs the number of monitored nodes S

As we can clearly see in Figure 5.5 the MIP approach offers a distinctive advantage in all networks over the heuristic solution. From a network point of view, we can deduce from Figure 5.6 that networks with a higher node degree such as COST239 require a higher percentage of monitored nodes to reduce the longest unmonitored

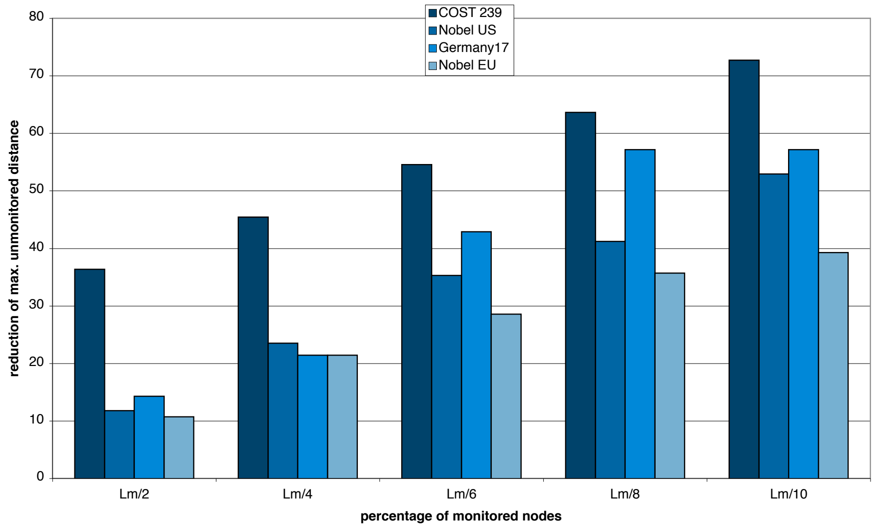


Figure 5.6: Percentage of monitored nodes vs relative length of the longest unmonitored light-path

distance. Recalling that a network with higher node degree offers more alternative paths, this observation is in accordance with our expectations. As a consequence in all networks except COST239, monitoring roughly 50% of the nodes is sufficient to reduce the length of the longest unmonitored path to $\frac{L_m}{10}$.

5.4.2 Finding the Optimal Selection of Monitoring Equipment

Scenario

As Stanic et al. exemplify in [SSC$^+$02], the selection of monitoring equipment is by no means unique. Both, price and detection capabilities (especially of softer errors) vary greatly. Likewise the probability of the different failure types ranges from fairly common fibre cuts to rarer incidents such as jamming (due to a defect transceiver) or fibre bending. Thus our challenge is to decide where to place which monitoring equipment to achieve the best failure localization probability with a given budget. We will start with a very general formulation and show how we use this as a basis for the following case study.

Problem Formulation

Just as in the problem in the previous section, we model our network as an undirected Graph $\mathcal{F}(\mathbb{V}, \mathbb{F})$. We divide the set of monitoring equipment $\mathbb{M}$ into monitors which can examine all incoming fibres $\mathbb{M}_V \subseteq \mathbb{M}$ (and therefore can be placed on a per node basis) and monitors which are limited to a single fibre $\mathbb{M}_E \subseteq \mathbb{M}$. For every node $v \in \mathbb{V}$, we create a set of binary variables $s_{n,v}$ for all $m \in \mathbb{M}_V$ such that

$$s_{m,v} = \begin{cases} 1 \textit{ iff } \text{equipment } m \text{ is placed at } v \\ 0 \text{ otherwise.} \end{cases} \tag{5.30}$$

We build a similar set of variables $s_{v,e,m}$ for all $m \in \mathbb{M}_E$ with the adjacent fibres of v denoted by $\mathbb{F}_v$ as additional index. For a set of failure types $\mathcal{T}$, we denote the monitors capable of detecting a failure $t \in \mathcal{T}$ by $\mathbb{M}_t$. We use a continuous variable $d_{p,t}$ to indicate whether a failure t can be detected on path $p \in \mathbb{L}$ via the constraint

$$d_{p,t} \leq \sum_{v \in p \setminus \left\{ v_p^s, v_p^t \right\}} \left(\sum_{\substack{m \in \mathbb{M}_t \wedge \\ m \in \mathbb{M}_V}} s_{v,m} + \sum_{\substack{m \in \mathbb{M}_t \wedge \\ m \in \mathbb{M}_E}} \sum_{\substack{e \in \mathbb{F}_v \wedge \\ e \in p}} s_{v,e,m} \right). \tag{5.31}$$

In contrast to the previous formulation, $\mathbb{L}$ is equivalent to the admissible paths $\mathbb{L}^A$. The penalty $k_{p,t}$ for an undetectable failure type t on path p is defined by

$$k_{p,t} = \mathsf{P}_{p,t} \cdot W_p \tag{5.32}$$

whereby $\mathsf{P}_{p,t}$ is the failure probability of failure type t on path p and W_p is a path dependent weight to model the relative importance among the paths. It is important to keep in mind, that we do not need to constrain the function behind $\mathsf{P}_{p,t}$ in any way, as long as we can precompute the used values. Consequently we can calculate the total penalty k_p of a path p by

$$k_p = \sum_{t \in \mathcal{T}} (1 - d_{p,t})\, k_{p,t}. \tag{5.33}$$

Just as in the previous formulation the difference in parentheses will be 0 *iff* suitable monitoring equipment is placed along the path. Our goal is to spend a budget B as wisely as possible, that is to minimize the total penalty of the network. Assuming that a monitor m causes a cost of K_m, we can calculate the total cost of a given monitor placement via

$$\sum_{v \in \mathbb{V}} \left(\sum_{m \in \mathbb{M}_V} s_{v,m} \cdot K_m + \sum_{e \in \mathbb{E}_v} \sum_{m \in \mathbb{M}_E} s_{v,e,m} \right) \leq B, \tag{5.34}$$

and minimize

$$K = \min \sum_{p \in \mathbb{L}} \mathsf{P}_p. \tag{5.35}$$

Case Study

With the presented formulation, we are able to describe a variety of monitoring problems. As a proof of concept in **[KM07]**, we gathered failure types from [MT00], failure probabilities from [VCD+05], and monitoring equipment characteristics from [SSC+02], which we summarize in Table 5.3. Prices for monitoring equipment are rather difficult to obtain, especially for the more complex types II and III, hence our prices of

Table 5.3: Failure Detection Capabilities of the Monitoring Equipment and Failure Probabilities

Failure t	ME-I	ME-II	ME-III	Probability $\mathsf{P}_{p,t}$
Node Failure (N)	yes	yes	yes	$8 \cdot 10^{-5}$
Link Failure[20] (L)	yes	yes	yes	$5.09694 \cdot 10^{-4}$
In-Band Jamming (I)	no	no	yes	10^{-5}
Out-Band Jamming (O)	no	yes	yes	10^{-5}
Time Distortion (T)	no	no	yes	10^{-5}

- ME-I €2,200
- ME-II €50,000
- ME-III €201,100

have to be considered as "well-educated guesses". In Figure 5.7 we depict the resulting penalty for optimal monitoring configurations for a budget of €9000 to €180000 with the above presented MIP implemented in CPLEX 9.1.

Quite unsurprisingly, the dominant factor in reducing the total penalty which is $6,044,090$ for a completely unmonitored network, is the detection of fibre cuts. For this reason, all the budget is spent on the cheapest monitors (which can already detect cuts), which reduces the penalty to 4570. Notably enough, we can already reduce the penalty by putting monitoring equipment on half of the nodes to 4650 as the results from our previous section suggest. The remaining penalty will be gradually reduced by increasing the budget and thus placing increasingly expensive monitors. The impact however compared to the cheaper equipment is relatively low, due to the low probability of the softer failures.

Our simulation performance was sufficent, i. e. we could solve every problem in at most 20h to optimality on a 2.4Ghz workstation. More densely meshed networks as for example COST239, for which we carried out a similar study, can considerably increase this time (up to five days in this case). Given the stated purpose, we do not consider these running times to be problematic, since monitoring equipment is usually not bought on a daily basis. Furthermore, the path set will usually be smaller than our full path set due to physical (range limitation, ...) and political (routing policies, ...) restrictions.

[20] The above given numerical value is for a path-length of $120\,\text{km}$ and is calculated as follows: Two multiplexers are needed for every transparent path (availability 0.99995209), one amplifier every $80\,\text{km}$ (availability 0.999968) and a fibre with an availability of 0.9999965 per kilometer. The failure probability of the light-path can thus be calculated via

$$\mathsf{P}_{p,\mathrm{L}} = 1 - 0.99995209^2 \cdot 0.999968^{\left\lceil \frac{L_p}{80\mathrm{km}} \right\rceil} \cdot 0.9999965^{\left\lceil \frac{L_p}{1\mathrm{km}} \right\rceil} \tag{5.36}$$

where L_p represents the length of path.

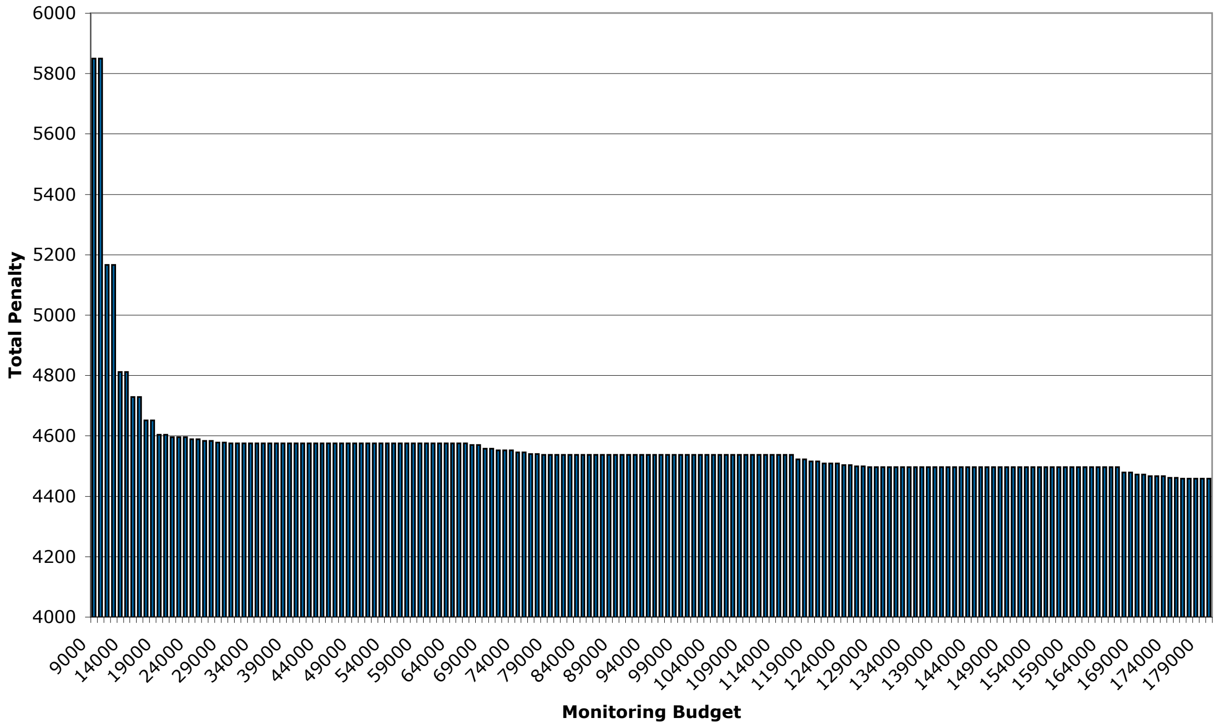


Figure 5.7: Total Penalty vs Monitoring Budget for the Nobel US network

From a more general point of view, our monitoring MIP could be used to model a variety of real-life problems. The basic assumption in our model is, that it is desireable to monitor as many streams (of different kinds) with monitoring equipment of various types causing costs. From traffic monitoring on highways to gas pipelines, this problem appears in many contexts.

5.5 Summary

In this chapter, our focus was on handling network failures: Either on resilience methods to prevent a service interruption for customers or on localization to find a failure faster once it occured.

1. In Section 5.2 our goal was to plan a shared protection scheme optimally. Due to the possibility of sharing, a heuristic reduction of the path set to achieve suitable run-times with black-box optimization algorithms is not likely to achieve good results. To this end, we developed a Branch-and-Price algorithm, which demonstrated that rather large problems can be solved to reasonable bounds rather quickly.
2. In Section 5.3, we identified dual-homing to be a promising way of exploiting existing disjointness in order to provide resilience against dual-link failures comparably cost-effective.

3. In the last part, we introduced the monitoring problem in transparent or translucent networks, developed a MIP formulation and could show in a case study, that only half of the nodes need monitoring capabilities in order to reduce the length of the longest unmonitored path to $\frac{1}{10}$ of the original value. We furthermore enhanced our planning methodology to take different kinds of monitoring equipment with different failure detection capabilities into account.

Open Issues

Our Branch-and-Price algorithm is (as most branching algorithms) hit by symmetries – in case a large number of cost-wise identical solutions exist, the algorithm has to explore the branches in the tree to the very bottom just to discover that the solution is not cheaper than the one already existing. Thus, in order to reduce the remaining gap further, the use of cutting planes (thereby becoming a Branch-Cut-and-Price algorithm) seems like a viable option.
Our optimization algorithm has one noteworthy property that appears interesting for further theoretical research: We have a potentially exponential number of subproblems to solve, because with every additional working path, we have to search for more backup paths. The question that arises is how this relates to the total complexity of the problem, i. e. how this relates to [GLS81]. Existing $\mathcal{NP}$-completeness proofs relate to a different formulation (namely the one with path-pairs). It would be furthermore interesting to evaluate, whether we actually have an exponential number of subproblems: That is, can we construct an example where we need to add *all* working paths to find the "right" backup-/working-path pair or find an upper bound on the number of working paths we have to add?

In order to increase the efficiency of our dual-homing protection method, it might be promising to integrate the choices of the "second home" into the planning alorithm itself instead of preselecting the closest node. Thereby, we could also accomodate further requirements (as for example load limits on the nodes). As noted in the text, our MIP is right now already at the verge of being too large; Before any larger problems can be considered a further speed-up of the algorithms is necessary. Unfortunately double-link failures complicate the pricing algorithms to $\mathcal{NP}$-hardness in most cases [OP08].

6 Connectivity in Hybrid Wireless Optical Broadband Access Networks

"I sense an insatiable demand for connectivity."[1]

While the previous two chapters dealt with fixed networks with a main focus on backbone networks, we will now move to wireless multi-hop networks, which could serve as a future access technology for the *last mile*[2]. As we will show in the first section, connectivity is a challenge in real-life urban scenarios, where access-points are rarely evenly spread across an area.

One of the most promising technologies to improve this situation are beamforming antennas, which offer an additional degree of freedom (the direction of the beams). While they have been exhibited to perform favourably, even with no systematic control of the beam-direction, heuristic approaches to increase the connectivity have been established in the recent literature. A fundamental connectivity question however remained open: If and how can a network with beamforming antennas be configured, so that all nodes are connected? In the third part of this chapter, we will develop MIP-formulations to provide a provable optimum thereby facilitating benchmarking of existing and development of future heuristics. In a case study, we will compare heuristic with optimal results and show that there is indeed considerable room for improvement.

6.1 Availability in WOBANs

Existing research for PONs suggests that the optical portion of the network is relatively uncritical: Optical backbone networks (we refer to the literature and data discussed in 5.1.1) reach very high availabilities way beyond 99% and PONs in access scenarios are believed to perform only slightly worse [TMP06, WC07].

Similarly, fading models form a large body of theory in the research of mobile systems. Data or studies on measured statistical behaviour of wireless links (which we would need for availability considerations) on the other hand are scarce articles.

[1] Clifford Stoll, *Silicon Snake Oil*

[2] In order to reflect its newfound importance, the term *first mile* becomes more and more common.

6.1.1 Link Probabilities in the Wireless Mesh

Base-Station – Base-Station

We found three notable exceptions to this rule. First Lundgren, Nordström and Tschudin [LNT02] performed real-life measurements in a wireless mesh-network to evaluate the performance of their routing protocol and observed significant differences between simulated and measured performance, which they attribute to a "grey zone" of links with neither high nor low packet delivery ratios.
Second, Yarvis et al. [YCK+02] performed similar test within a wireless mesh network inside the Moscone Conference Center. They also found that the unexpected high packet-loss ratios affected the performance of their protocols.
Last, Aguayo et al. from the pioneering MIT RoofNet project (an urban wireless mesh network in Cambridge, MA, USA depicted in Figure 6.1) published link-level measurements in [ABB+04], in which one Base Station (BS) at a time transmitted broadcast packets for 90s while the other BSs tried to receive these packets. The resulting packet

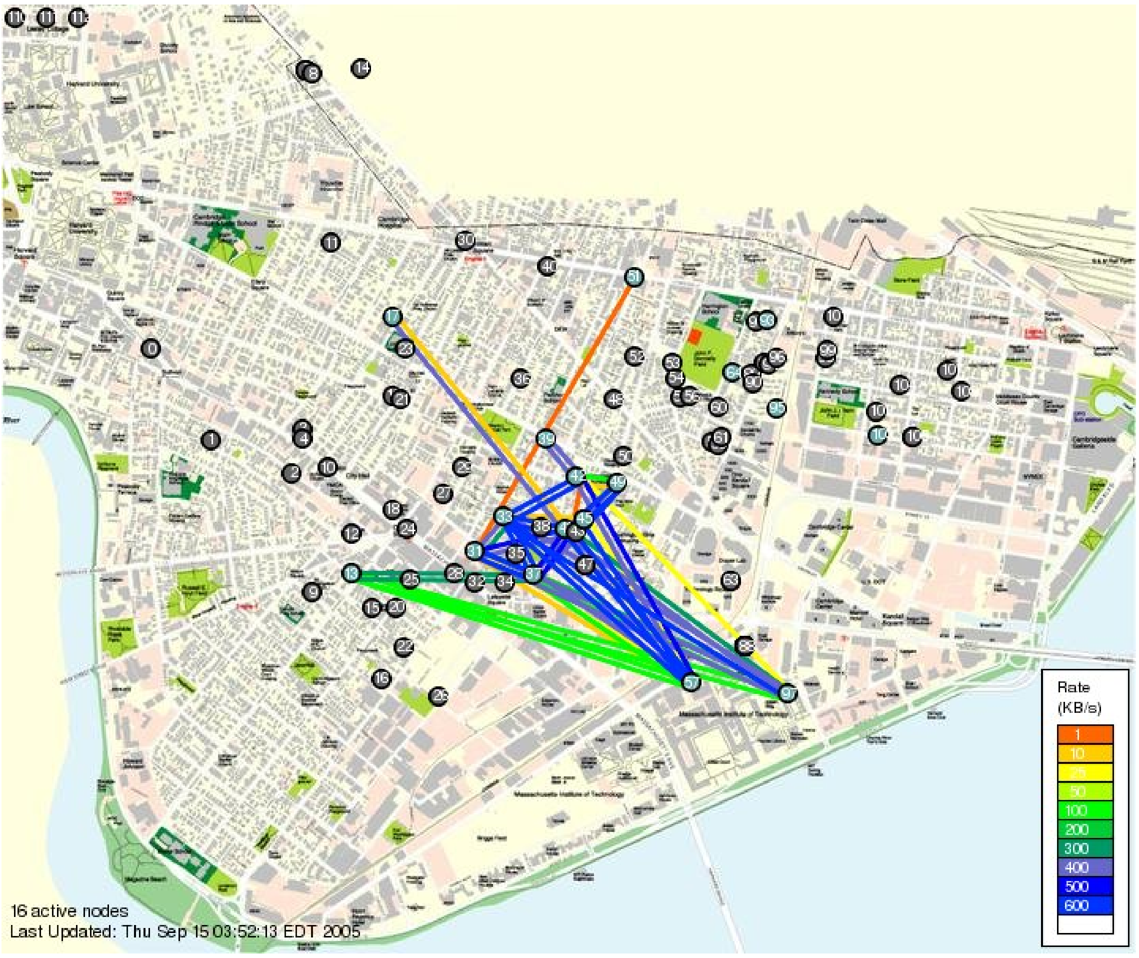


Figure 6.1: MIT RoofNet Layout Snapshot from 2005 [roo]

loss ratios confirmed the findings of the former two publications. Statistical analysis revealed two remarkable facts: Interestingly enough, no larger-scale correlations of the delivery ratios could be found, or in other words: Contrary to the usual expectation, the weather did not affect the link quality. Furthermore, the authors found that while the SNR was a weak predictive measure of the resulting packet loss rates, links with a high SNR were unlikely to exhibit high loss rates.

Hence it is already safe to state at this point, that an improvement of the link quality (i. e. the SNR), will in turn improve the overall performance of the wireless mesh network. In the remainder of this section, we will show how to estimate the resulting end-to-end availabilities when using WOBAN routing algorithms and summarize the results from **[KGS⁺09]**.

Due to the immediate availability of the data, we use the RoofNet scenario as the data basis for our evaluation. To this end, we consider the delivery ratio of a link as its link probability. While this might seem to be a rather harsh assumption, the statistical analysis performed in [ABB⁺04], strongly suggests, that the scenario available in machine-readable form is indeed typical. Furthermore wireless mesh network routing algorithms are suited towards rather rapid changes in the topology, thus as long as no large scale changes appear (which the authors negate), our results should represent the underlying scenario as well.

Mobile Station – Base Station

Unfortunately, no measurements were available from Mobile Stations (MSs) (i. e. users) within the RoofNet. To overcome this weakness, we use a link probability model proposed by Bettstetter and Hartmann [BH05] which incorporates *Shadow Fading*, i. e. tries to accomodate urban environments, namely shadowing through obstacles in the transmission path. Starting with a basic modified free-space [Fri46] path-loss model [PS02, p. 257], we gather

$$P_{L,v_1,v_2} = -10\log_{10}\left(\frac{\lambda}{4\pi}\right)^2 - 10\log_{10}\left(d_{v_1,v_2}^{-a}\right) \tag{6.1}$$

for the path-loss P_L in dB between two nodes v_1 and v_2 set apart by a distance d_{v_1,v_2}. $\lambda = \frac{c}{f}$ is the wavelength of the radio signal, and a is the path-loss exponent modelling various environments. A value of $a = 2$ thus represents the pure free-space transmission, while obstructed paths such as in urban areas or inside buildings reach values of 3 to 6. For the BS–BS links with the reference data obtained from the MIT RoofNet project, we find a value of $a = 2.17$ to cause the smallest mean square error as detailed in **[KGS⁺09]**, which reflects the fact that most of the BSs are mounted on roof-tops. On top of this purely geometric model, which is solely determined by the distance d_{v_1,v_2}, we add a stochastic component X

$$P_{L,v_1,v_2} = -10\log_{10}\left(\frac{\lambda}{4\pi}\right)^2 - 10\log_{10}\left(d_{v_1,v_2}^{-a}\right) - \mathsf{X} \tag{6.2}$$

modelling obstructions in the signal-path known as *shadow fading* [Rap96, p. 104]. The authors of [BH05] use the standard approach and choose X to follow a zero-mean normal distribution with a standard deviation of σ (*log-normal shadowing*). Again a minimization of the mean square error yields a rather large value for $\sigma = 10.65\,\mathrm{dB}$ which is above the upper end of values found for σ in some parts of the literature, for example by Geng and Wiesbeck [GW98, p. 185]. Other studies show even larger σ, most notably Seidel et al. in [SRJ+91] with $\sigma = 11.8\,\mathrm{dB}$ for some German cities. The large value of sigma indicates a great randomness in the path-loss, which is well in line with the findings of the authors of [ABB+04]. Now that we can calculate the path-loss between a MS and a BS, we can determine the link probability quite straigtforward. We can establish a link between two nodes v_1 and v_2, *iff* the power of the received signal is larger than the receiver sensitivity, i. e. larger than a minimum signal power-level $P_{r,\min}$. Thus for a system without antenna gains (or losses for that matter), we can establish a link between v_1 and v_2 *iff*

$$P_t - P_{L,v_1,v_2} - P_{r,\min} \geq 0. \tag{6.3}$$

As Hartmann and Bettstetter further detail, the link probability P_{v_1,v_2} can be calculated as

$$\mathsf{P}_{v_1,v_2} = \frac{1}{2} - \frac{1}{2}\mathrm{erf}\left[\frac{10a}{\sqrt{2}\sigma}\log_{10}\left(\frac{d_{v_1,v_2}}{\exp_{10}\left(\frac{P_t - P_{r,\min}}{a \cdot 10\,\mathrm{dB}}\right)\mathrm{m}}\right)\mathrm{dB}\right] \tag{6.4}$$

which is usually shortened by introducing a reference distance

$$d_0 = \exp_{10}\left(\frac{P_t - P_{r,\min}}{a \cdot 10\,\mathrm{dB}}\right)\mathrm{m} \tag{6.5}$$

to

$$\mathsf{P}_{v_1,v_2} = \frac{1}{2} - \frac{1}{2}\mathrm{erf}\left[\frac{10a}{\sqrt{2}\sigma}\log_{10}\left(\frac{d_{v_1,v_2}}{d_0}\right)\mathrm{dB}\right]. \tag{6.6}$$

6.1.2 Path Availability

With the link probabilities calculated with the values above (and a value of 0.9999 for optical links in the WOBAN scenario), we arrive at an expression for the availability A_p of a whole path p consisting of links $e \in p$ through the network:

$$\mathsf{A}_p = \prod_{e \in p} \mathsf{P}_e. \tag{6.7}$$

6.1.3 Availability Evaluation

For evaluation purposes, we had access to implementation of the following state of the art WOBAN routing algorithms developed at UC Davis, CA, USA in Prof. Muhkherjee's group:

- The *Delay-Aware Routing Algorithm (DARA)* [SYDM08] minimizes the average packet delay in the wireless portion of the WOBAN.
- The *Capacity and Delay Aware Routing (CaDAR)* [RRS+08] distributes traffic among the wireless links and performs delay-aware routing to support higher loads in the wireless portion.
- *Capacity and Flow Assignment (CFA)* problems have been studied and solved to local optimality (i. e. for one flow at a time) in a classic article by Fratta, Gerla and Kleinrock [FGK73].

A performance comparison (regarding capacity and delay) of these routing algorithms is presented in [RRS07]. It is worth noting that none of these algorithms takes the link quality directly into account. Using the routing tables gathered from these three routing algorithms, we can estimate the availability of the resulting paths by following Equation (6.7).

The resulting end-to-end availabilities (from a MS to a BS) are rather low: In the best case, we observe average unavailabilities of roughly 40%. An obvious idea to incorporate the link quality into the routing algorithms is to neglect links below a certain probability. Recalling however that there is no sharp distinction between good an bad links (i. e. a lot of the links are in the "grey zone"), we can immediately see that a high threshold will simply disconnect a considerable number of BSs. Using simple routing algorithms (shortest path with unavailabilities as link weights), we can decrease the average unavailability to roughly 25%, which is still rather high, at least compared to the estimated values of PONs.

In conjunctions with the findings of the three articles cited in the beginning of this section, we can state that connectivity is indeed a challenge in real-life wireless mesh networks which is not cured by current routing algorithms (which admittedly focus on efficiency). While a modified routing scheme might improve the situation somewhat, even then the path availability is far from satisfactory and significantly lower than in conventional wired approaches. For further details concerning the WOBAN-evaluation (differences of the individual routing algorithms, consequences of hypothetical multipath routing, end-to-end path availability within a WOBAN) we refer to **[KGS+09]**.

6.2 Connectivity Bounds in Wireless Mesh Networks with Beamforming Antennas

Considering the age of the cited papers in the previous section, it is more than logical that connectivity has already been identified to be one of the two[3] "most severe limitations" in multi-hop networks [Vil09, p. 6]. In its purest (and from an algorithmic point of view, most difficult) form it appears in adhoc networks, where no central entity is

[3]The other one being throughput, which in return was the main focus of CaDAR and CFA in the previous section.

available to assist in configuration decisions. Within such a network, we usually want to make sure that each node is able to communicate with every other node, which we describe with the graph-theoretical term *connected*. As discussed in the beginning of this chapter, communication is not limited to direct connections (which would require a fully meshed network), but can take several intermediate hops. Obviously, the relatively hard connectivity criterion allows only a binary decision and thus does not differentiate between different degrees of "disconnectivity": In the most extreme case, only a single node cannot reach any other node, which is clearly different from a case where all nodes are isolated. Hence, the *path probability* Π has been defined to measure connectivity in a network as

$$\Pi = \frac{\text{number of connected node pairs}}{\text{number of node pairs}}. \tag{6.8}$$

It is easy to see, that connectivity in a WOBAN can be considered as a relaxation or a special case[4] of mesh networks: Our connectivity requirements will be fulfilled, if a MS can reach a BS, which in turn should be able to reach a BS connected to the PON. Similarly our findings can also be applied to adhoc networks, however *mobility* will have to be considered in those cases. However, we assume static node locations which in turn can be seen as a snapshot of an adhoc network.

Omnidirectional Antennas Cheng and Robertazzi examined connectivity in multihop networks in the context of broadcasting information in [CR89] as early as 1989. Using certain assumptions on the distribution of the omnidirectional nodes and a purely geometrical transmission range, they could demonstrate that a given transmission range above a certain limit depending on the density of the nodes, causes a network to be connected with high probability. Santi, Blough and Vainstein refine this probabilistic approach for omnidirectional nodes in [SBV01] and extend it to the three dimensional case. Bettstetter and Hartmann extend this line of research even further by integrating shadow fading in their models in [BH03, BH05]. It is important to keep in mind however that in a realistic scenario we can hardly increase the transmission power arbitrarily without causing new challenges regarding energy efficiency (which is not only of considerable interest for mobile nodes but also for BSs) and interference as demonstrated by Blough et al. in [BLRS07] which cannot easily be solved via *topology control*.

Persistent Beamforming Another approach to *connectivity shaping* is to keep the transmission power constant, but not omnidirectional, that is to use *directional antennas*. Interestingly enough, Bettstetter et al. observed in [BHM05] that a low-complexity approach regarding both, the directional antenna and the control of the antenna direction (namely simply choosing a random direction[5]) will improve the connectivity. This last result is especially interesting because in their earlier work [BH03, BH05]

[4]Which is sometimes marked as *Wireless Mesh Access Network*.
[5]Hence the name *Random Direction Beamforming (RDB)*.

they experienced the same phenomenon, where a larger variance in the shadow fading had a positive effect on connectivity. This similarity was confirmed analytically for "moderate" shadowing variances in a recent article [ZJDSar] by Zhou et al. An analytical study by Koskinen [Kos06] highlights the drawback of this approach: The probability of isolated nodes also rises (which follows the observation of Bettstetter et al. that the variance of the node degree increases). By applying a random geometric graph theorem by Penrose [Pen99] he further concludes that random direction beamforming neither decreases nor increases the probability of a network to be connected per se. The assumption that a graph resulting from RDB can be treated as a random geometric graph is questioned by Vilzmann [Vil09, p. 71], since contrary to "normal" random geometric graphs, the existence of a link between two nodes is not entireley determined by the distance. This however questions the applicability of Koskinen's last conclusion. An analytical study [DZJ08] by Durrani, Zhou and Jones in turn supports the findings of Bettstetter and Hartmann for path loss exponents $\alpha < 3$.

Naturally, "smarter" approaches to steer the direction of the beamforming antennas have been devised as well. In [VWAH06] Vilzmann et al. develop two distributed algorithms, namely *Maximum Node Degree Beamforming (MNDB)* and *Two-Hop Node Degree Beamforming (TNDB)* which try to maximize the node degree and the two-hop node degree respectively by sweeping the main lobe of a node to gather local information about its neighbours. Related to this approach is an algorithm proposed by Zhou et al. in [ZJDS07] named *greedy beamforming* where the nodes have some knowledge (several variants are considered) about the locations of other nodes. In the same work, they also consider *centre directed beamforming*, where all nodes know the geometric centre of their network and point their beam towards this direction. Their first approach outperforms the latter, but both offer better overall connectivity than RDB.

Dynamic Beamforming All approaches considered so far have in common, that their beamforming configuration (once found) is intended to remain relatively static. This is however by no means a requirement – antenna arrays require no mechanical parts to change the direction of their beamforming pattern. Thus a substantial body of work proposes to adapt the beamforming patterns in the most extreme case on a per packet basis in contrast to our *persistent beamforming*. One of the main challenges in this line of work is the adaption of the Medium Access Control (MAC) layer. Since we will exclusively consider persistent beamforming in the remainder of this chapter, we refer to [Vil09, pp. 104] for a recent and extensive in-depth study on this topic.

Contributions As can be seen from the cited articles above, the topological consequences of beamforming have only attracted considerable attention from the scientific community relatively recently. The existing studies employ either a very simple beamforming scheme (e. g. RDB) and try to gather statistical information about the resulting network structure using analytical methods, or use a smart beamforming scheme and consequently rely on simulations to demonstrate the effectiveness of the respective scheme. A third way, somewhat in between these two extremes, has

been proposed by Hartmann, Kiese and Vilzmann in **[HKV08]**. Using a simplified beamforming model, we could formulate a MIP which provided optimality bounds for MNDB in a given scenario. This bound tells how large the average node degree would be if global information was available in the network and the nodes would steer their nodes according to a globally optimal configuration. While this is rather unrealistic for an adhoc network (which was the prime focus of MNDB), it would be at least thinkable for a mesh network, where the base stations remain comparably static.
Refining this approach, we will develop MIP-models which will provide a configuration for a connected network, if such a configuration exists and otherwise offer the highest path probability possible. We will furthermore show how several other relevant auxilliary objectives can be integrated in this scheme and use large-scale optimization methods to be able to cope with realistic scenario sizes without sacrificing optimality in convenient simulation times. In the concluding case study, we will examine how good MNDB performs against a proovable optimum and discuss whether or not, further optimizing the node degree will be a rewarding goal towards better connectivity.

6.2.1 Network Model

Wireless Channel

Just as in the previous section, we start with a modified free space [Fri46] path-loss model [PS02, p. 257], however this time we take the respective antenna gains into account. Thus, for a node v_2 receiving from a node v_1 transmitting with power p_t at wavelength λ, we can calculate the received power as

$$p_{r,v_1,v_2} = p_t \cdot g_{v_1} \cdot g_{v_2} \cdot \left(\frac{\lambda}{4\pi \cdot d_{v_1,v_2}}\right)^2 \left(\frac{d_0}{d_{v_1,v_2}}\right)^{a-2}, \tag{6.9}$$

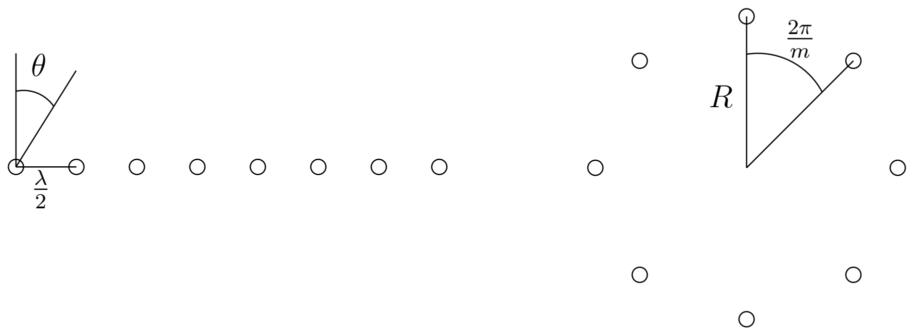
where g_{v_1} and g_{v_2} are the respective antenna gains and d_0 the reference distance similarly defined as in Equation (6.5). As we have stated before, we intend to use directional or beamforming antennas, hence the antenna gains are depending on the direction γ these antennas are actually pointing to, i. e. $g_{v_1} = g_{v_1}(\gamma_{v_1})$ and $g_{v_2} = g_{v_2}(\gamma_{v_2})$ respectively. In this thesis we will assume that we use identical antennas on all nodes, consequently the above can be simplified to $g_{v_1} = g(\gamma_{v_1})$ and $g_{v_2} = g(\gamma_{v_2})$. Transforming Equation (6.9) to the dB-domain gains

$$P_{r,v_1,v_2} = P_t + G(\gamma_{v_1}) + G(\gamma_{v_2}) + \underbrace{10\log_{10}\left(\frac{\lambda}{4\pi \cdot d_{v_1,v_2}}\right)^2 + 10\log_{10}\left(\frac{d_0}{d_{v_1,v_2}}\right)^{a-2}}_{-P_{L,v_1,v_2}}, \tag{6.10}$$

where upper-case variables indicate logarithmic-scales. The expression marked with $-P_{L,v_1,v_2}$ can obviously be interpreted as the path gain, i. e. the opposite-signed path loss.

Beamforming Antennas

We will consider an idealized beam-pattern gained from a so-called *phased array* of antennas. Although these arrays have been mentioned at the beginning of the last century by Braun [Bra09, p. 239], only the rise of cost-efficient means of signal processing during the last 20 years has facilitated their wide use in (commodity) mobile communications, where they are referred to as *smart antennas*.



(a) Uniform Linear Array (ULA) (b) Uniform Circular Array (UCA)

Figure 6.2: Linear and Circular Phased Arrays

Linear Phased Array The layout of a linear phased array is depicted in Figure 6.2a. For perfectly isotropical antenna elements spaced $\frac{\lambda}{2}$ apart, we can calculate the lateral gain[6] in (azimuthal) direction θ by summing over the received signals multiplied with the corresponding (complex) weight x_i

$$g(\theta) = \sum_{i=0}^{m-1} x_i e^{ji\pi \sin(\theta)} \tag{6.11}$$

For a constant phase shift ς between the x_i's, i. e. $x_i = A_i e^{ji\varsigma}$, we can rewrite the above as

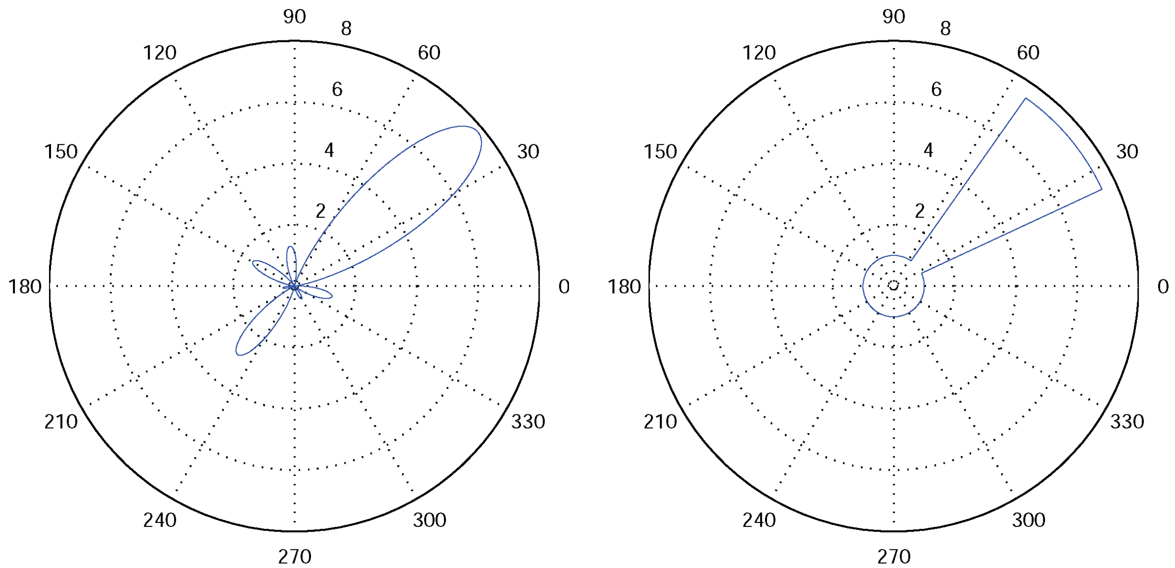
$$g(\theta) = \sum_{i=0}^{m-1} A_i e^{ji(\pi \sin(\theta) + \varsigma)}. \tag{6.12}$$

Close inspection of this equation reveals that we will achieve maximum gain, when the exponent is zero, i. e.

$$\varsigma = -\pi \sin(\theta). \tag{6.13}$$

Consequently, once we choose a direction γ, we can determine the phase shift ς, such that $g(\gamma)$ becomes maximal, hence the main lobe of the resulting pattern points in direction γ.

[6] We assume that we are far away compared to the distance of the $\frac{\lambda}{2}$-spacing.



(a) Realistic Pattern (b) Keyhole Pattern

Figure 6.3: Antenna Patterns

Circular Phased Array In Figure 6.2b the layout of a circular array is depicted. Using the same restrictions as in the previous paragraph (lateral cut of the antenna pattern[7], perfectly isotropical antennas), we gather for the sum of the received signals of the m antenna elements evenly placed on a circle with radius R

$$g(\gamma) = \sum_{i=0}^{m-1} x_i e^{-j\frac{2\pi}{\lambda} R\cos\left(\theta - i\frac{2\pi}{m}\right)}. \tag{6.14}$$

Similarly to the linear array, we will arrive at maximum gain in direction γ, if the exponent is zero for $\theta = \gamma$, thus we dimensions x_i to be

$$x_i = A_i e^{j\frac{2\pi}{\lambda} R\cos\left(\gamma - i\frac{2\pi}{m}\right)}. \tag{6.15}$$

An exemplary gain pattern for a circular array with eight antenna elements is shown in Figure 6.3a. Unfortunately, this pattern is highly non-linear, which makes it rather unsuitable for integration in a MIP. Due to the path of abstraction taken in the following we will refer for further details on beamforming antennas to a state of the art survey on smart antennas [KBB+05] by Kaiser et al. and a classic text on digital beamforming [LL96] considering more general spacings, etc. by Litva and Lo.

Keyhole Pattern To this end, existing literature (e. g. [HE01, Ram01]) uses a simplification, which suits us perfectly: the so-called *keyhole* or *brickwall model* in Figure 6.3b. It condenses the beampattern to three parameters: the aperture α, the gain of the main

[7]While a linear phased array with isotropical antennas is rotationally symmetric, i. e. the elevation is irrelevant, the gain of circular arrays is actually depending on the evelation angle.

lobe G_M (i. e. within the aperture) and the gain of the side lobes G_S (outside the aperture).

Wireless Link

We assume that two nodes v_1 and v_2 will be able to communicate, *iff* the received power P_r exceeds the required reception power $P_{r,\min}$ of the specific receiver or by using Equation (6.10),

$$P_{r,\min} \leq P_t + G(\gamma_{v_1}) + G(\gamma_{v_2}) - P_{L,v_1,v_2} \tag{6.16}$$

In conjunction with our antenna model, we can now calculate two critical distances: First, we can determine the distance d_max when nodes are too far apart to be able to communicate even under best condidtions, i. e. both main lobes pointing to each other

$$d_\mathrm{max} = 10^{\frac{1}{10a}\left(-P_{r,\min}+P_t+2G_\mathrm{M}+10\log_{10}\left(\left(\frac{\lambda}{4\pi}\right)^2 d_0^{a-2}\right)\right)}. \tag{6.17}$$

Second, we can find the distance d_min under which nodes will be always able to communicate because our link-budget (6.16) is fulfilled even with the side-lobe gains G_S:

$$d_\mathrm{min} = 10^{\frac{1}{10a}\left(-P_{r,\min}+P_t+2G_\mathrm{S}+10\log_{10}\left(\left(\frac{\lambda}{4\pi}\right)^2 d_0^{a-2}\right)\right)} \tag{6.18}$$

6.2.2 Optimization Model

Equations (6.17) and (6.18) constrain the nodes we have to consider in our optimization model: Between the nodes of a node-pair $\{v_1, v_2\}$ which are further apart than d_max, we will not be able to establish a (direct) link, no matter how sophisticated we control our beampatterns. In contrast, if a node-pair $\{v_1, v_2\}$ is closer than d_min, these nodes are connected independently of the direction of the main lobe. For all node-pairs between those two extremes however, we have to take the direction of the respective main lobes into account to determine whether a link between them exists. Looking at the basic scenario given in Figure 6.4, we can formulate the following condition that node v_2 lies in the main-lobe (of aperture α) of n_1 with direction γ_{v_1}:

$$\gamma_{v_1} + \frac{\alpha}{2} \geq \beta_{v_1,v_2} \quad \wedge \quad \gamma_{v_1} - \frac{\alpha}{2} \leq \beta_{v_1,v_2}. \tag{6.19}$$

The first inequality ensures that β_{v_1,v_2} is smaller than the upper border angle of the main-lobe, and the second that β_{v_1,v_2} is larger than the lower border angle of the main-lobe. It is important to note however that due to the angle discontinuity from 2π to 0, we have to distinguish two further cases because clearly (6.19) is only valid for $\beta_{v_1,v_2} \in \left[\frac{\alpha}{2}; 2\pi - \frac{\alpha}{2}\right]$:

- For $\beta_{v_1,v_2} \in \left]2\pi - \frac{\alpha}{2}; 2\pi\right]$:

$$\gamma_{v_1} + \frac{\alpha}{2} \geq \beta_{v_1,v_2} \quad \vee \quad \gamma_{v_1} - \frac{\alpha}{2} + 2\pi \leq \beta_{v_1,v_2} \tag{6.20}$$

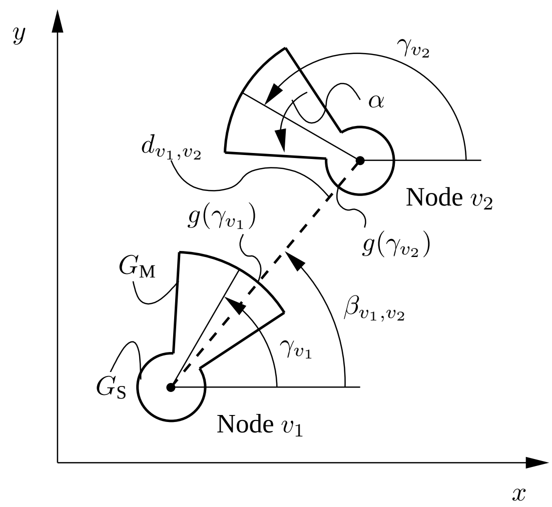


Figure 6.4: Example Configuration

- For $\beta_{v_1,v_2} \in \left[0; \frac{\alpha}{2}\right[$:

$$\gamma_{v_1} + \frac{\alpha}{2} + 2\pi \geq \beta_{v_1,v_2} \quad \vee \quad \gamma_{v_1} - \frac{\alpha}{2} \leq \beta_{v_1,v_2} \tag{6.21}$$

Link Indication

In our optimization model, we will use a graph $\mathcal{G}(\mathbb{V}, \mathbb{E})$ to represent the links we can establish in our network: For every node-pair $\{v_1, v_2\}$ which is closer together than d_max, we create two arcs $e_1 = (v_1, v_2)$ and $e_2 = (v_2, v_1)$. Our first task to formulate an optimization problem is to translate the conditions (6.19) – (6.21) into linear inequalities, which we will exemplify in the following for the first case. For the first part of Equation (6.19), we introduce a continuous variable a'_{v_1,v_2} which we set via

$$a'_{v_1,v_2} = \gamma_{v_1} - \beta_{v_1,v_2} + \frac{\alpha}{2} + 1. \tag{6.22}$$

a'_{v_1,v_2} will become larger than one, *iff* the first condition of (6.19) is met and smaller than one *iff* not. We transform this to a binary decision, where a *binary* variable b'_{v_1,v_2} with a value of zero represents, that the condition is met, and a value of one the opposite, which we achieve via the following constraint:

$$\left(-\beta_{v_1,v_2} + \frac{\alpha}{2}\right) \cdot b'_{v_1,v_2} - a'_{v_1,v_2} + 1 \leq 0. \tag{6.23}$$

With Equation (6.22), a'_{v_1,v_2} can become $\left[-\beta_{v_1,v_2} + \frac{\alpha}{2} + 1; 1\right[$ in the unmet case, which means that we have to set $b'_{v_1,v_2} = 1$ to keep the inequality valid. In case we meet the condition, a'_{v_1,v_2} will be in the range of $\left[1; 2\pi - \beta_{v_1,v_2} + \frac{\alpha}{2} + 1\right]$, which in turn allows for $b'_{v_1,v_2} = 0$.

Directing our attention to the second condition of (6.19), we will use a similar construct. We define a continuous variable a''_{v_1,v_2} to be larger than one *iff* the second condition is met, via

$$a''_{v_1,v_2} = -\gamma_{v_1} + \beta_{v_1,v_2} + \frac{\alpha}{2} + 1. \tag{6.24}$$

If we miss the second condition, a''_{v_1,v_2} will take values of $\left[-2\pi + \beta_{v_1,v_2} + \frac{\alpha}{2} + 1; 1\right[$. In this case, a second *binary* variable b''_{v_1,v_2} will become one and can be zero if not, i. e. a''_{v_1,v_2} takes values between $\left[1; \beta_{v_1,v_2} + \frac{\alpha}{2} + 1\right]$. We achieve this with a constraint following the structure of Inequality (6.23):

$$\left(-2\pi + \beta_{v_1,v_2} + \frac{\alpha}{2}\right) \cdot b''_{v_1,v_2} - a''_{v_1,v_2} + 1 \leq 0 \tag{6.25}$$

Condition (6.19) requires both parts to be fulfilled, which means that both b'_{v_1,v_2} as well as b''_{v_1,v_2} have to be zero at the same time. Finally, a *binary* variable b_{v_1,v_2} will indicate, whether the main lobe of v_1 points towards v_2 ($b_{v_1,v_2} = 1$) or not ($b_{v_1,v_2} = 0$), which we force with

$$b_{v_1,v_2} \leq \frac{1}{2}\left(1 - b'_{v_1,v_2}\right) + \frac{1}{2}\left(1 - b''_{v_1,v_2}\right). \tag{6.26}$$

We can formulate the remaining conditions (6.19) and (6.21) in a similar fashion with the only substantial difference being that the two sub-conditions have to be OR'ed in contrast to the logical AND above. This however is easy to accomodate, i. e. Inequality (6.26) changes to

$$b_{v_1,v_2} \leq \left(1 - b'_{v_1,v_2}\right) + \left(1 - b''_{v_1,v_2}\right). \tag{6.27}$$

In order for a link to exist, the link budget as stated in Equation (6.16) must be sufficient. Consequently if a *binary* variable l_{v_1,v_2} shall indicate with a value of one, that a link exists with antenna directions γ_{v_1} and γ_{v_2}, the constraint

$$(P_t - P_{r,\min} - P_{L,v_1,v_2} + 2 \cdot G_\mathrm{S})\, l_{v_1,v_2} + (G_\mathrm{M} - G_\mathrm{S})\, b_{v_1,v_2} + (G_\mathrm{M} - G_\mathrm{S})\, b_{v_2,v_1} \geq 0 \tag{6.28}$$

must hold for the respective node-pair (v_1, v_2). Since this inequality set forms the foundation of the remainder of this chapter, we will conclude this section with the following remarks:

1. In Inequalities (6.23) and (6.25), a value of one indicates that the respective constraint is met, however a value of zero does not necessarily indicate that the constraint is not met. In respect to our problem we can state, that if we want to set a link variable l_{v_1,v_2} to one, we must fulfill the underlying constraints, but setting a link variable to zero does not necessarily force one or all of the underlying constraints to be failed. In other words, if we want to force a link not to exist (i. e. steer the main lobes away), it will not be sufficient to force $l_{v_1,v_2} = 0$.
2. It might be tempting to use a more straightforward formulation for the link constraint (6.28), namely

$$P_t - P_{r,\min} - P_{L,v_1,v_2} \cdot l_{v_1,v_2} + (1 - b_{v_1,v_2})\, G_\mathrm{S} + b_{v_1,v_2} \cdot G_\mathrm{M} + (1 - b_{v_2,v_1})\, G_\mathrm{S} + b_{v_2,v_1} \cdot G_\mathrm{M} \geq 0,$$

which in fact we did in our early papers. As it is easy to demonstrate, this constraint only works correctly as long as $P_t - P_{r,\min} + 2G_{\mathrm{S}} \geq 0$, which was always the case in our scenarios. Otherwise, main-lobes will have to be steered towards a node in order to fulfill the constraint despite the link being non-existent or the MIP might even become infeasible. With the above formulation, our formulation is even applicable beyond this constraint. We thank the anonymous reviewer of **[KHLV09]** for pointing this issue out.

3. In the remainder of this chapter, we will always assume that we *cannot* steer send and receive gain independently, i. e. the main lobe always points in the same direction for sending and receiving. As a consequence, our graph $\mathcal{G}(\mathbb{V}, \mathbb{E})$ could become undirected and one variable l_e per node pair e will suffice (which we will use in the path approaches later in this section). Another possibility to incorporate this constraint is to force $l_{v_1,v_2} = l_{v_2,v_1}$, which is a bit more convenient for our flow approaches.
4. While we used a modified free-space path-loss model, we are by no means restricted to this. The only requirement we have, is that the path-loss can be precalculated and described by a simple real value.

Complete Formulation

In this section we provide the complete formulation up to the link indicator variables l_e. An edge $e \in \mathbb{E}$ exists for those node-pairs$\{v_1, v_2\} \subset \mathbb{V}$, where the distance d_{v_1,v_2} is smaller than $d_{\max}$.

- $\forall e = \{v_1, v_2\} \in \mathbb{E} : d_{\min} \leq d_{v_1,v_2} \leq d_{\max} \quad \wedge \quad \frac{\alpha}{2} \leq \beta_{v_1,v_2} \leq 2\pi - \frac{\alpha}{2}$

$$a'_{v_1,v_2} = -\beta_{v_1,v_2} + \frac{\alpha}{2} + 1 + \gamma_{v_1} \tag{6.29}$$

$$a''_{v_1,v_2} = \beta_{v_1,v_2} + \frac{\alpha}{2} + 1 - \gamma_{v_1} \tag{6.30}$$

$$\left(-\beta_{v_1,v_2} + \frac{\alpha}{2}\right) \cdot b'_{v_1,v_2} - a'_{v_1,v_2} + 1 \leq 0 \tag{6.31}$$

$$\left(-2\pi + \beta_{v_1,v_2} + \frac{\alpha}{2}\right) \cdot b''_{v_1,v_2} - a''_{v_1,v_2} + 1 \leq 0 \tag{6.32}$$

$$\frac{1}{2} \cdot \left(1 - b'_{v_1,v_2}\right) + \frac{1}{2} \cdot \left(1 - b''_{v_1,v_2}\right) - b_{v_1,v_2} \geq 0 \tag{6.33}$$

- $\forall e = \{v_1, v_2\} \in \mathbb{E} : d_{\min} \leq d_{v_1,v_2} \leq d_{\max} \quad \wedge \quad 2\pi - \frac{\alpha}{2} \leq \beta_{v_1,v_2} \leq 2\pi$

$$c'_{v_1,v_2} = -\beta_{v_1,v_2} + \frac{\alpha}{2} + 1 + \gamma_{v_1} \tag{6.34}$$

$$c''_{v_1,v_2} = \beta_{v_1,v_2} - 2\pi + \frac{\alpha}{2} + 1 - \gamma_{v_1} \tag{6.35}$$

$$\left(-\beta_{v_1,v_2} + \frac{\alpha}{2}\right) \cdot b'_{v_1,v_2} - c'_{v_1,v_2} + 1 \leq 0 \tag{6.36}$$

$$\left(-4\pi + \beta_{v_1,v_2} + \frac{\alpha}{2}\right) \cdot b''_{v_1,v_2} - c''_{v_1,v_2} + 1 \leq 0 \tag{6.37}$$

$$1 - b'_{v_1,v_2} + 1 - b''_{v_1,v_2} - b_{v_1,v_2} \geq 0 \tag{6.38}$$

- $\forall e = \{v_1, v_2\} \in \mathbb{E} : d_{\min} \leq d_{v_1,v_2} \leq d_{\max} \quad \wedge \quad 0 \leq \beta_{v_1,v_2} \leq \frac{\alpha}{2}$

$$d'_{v_1,v_2} = -\beta_{v_1,v_2} + \frac{\alpha}{2} - 2\pi + 1 + \gamma_{v_1} \tag{6.39}$$

$$d''_{v_1,v_2} = \beta_{v_1,v_2} + \frac{\alpha}{2} + 1 - \gamma_{v_1} \tag{6.40}$$

$$\left(-2\pi - \beta_{v_1,v_2} - \frac{\alpha}{2}\right) \cdot b'_{v_1,v_2} - d'_{v_1,v_2} + 1 \leq 0 \tag{6.41}$$

$$\left(-2\pi + \beta_{v_1,v_2} + \frac{\alpha}{2}\right) \cdot b''_{v_1,v_2} - d''_{v_1,v_2} + 1 \leq 0 \tag{6.42}$$

$$1 - b'_{v_1,v_2} + 1 - b''_{v_1,v_2} - b_{v_1,v_2} \geq 0 \tag{6.43}$$

- $\forall e \in \mathbb{E} : d_{\min} \leq d_{v_1,v_2} \leq d_{\max}$

$$(P_t - P_{r,\min} - P_{L,v_1,v_2} + 2 \cdot G_{\text{S}})\, l_{v_1,v_2} + (G_{\text{M}} - G_{\text{S}})\, b_{v_1,v_2} + (G_{\text{M}} - G_{\text{S}})\, b_{v_2,v_1} \geq 0 \tag{6.44}$$

- $\forall e = \{v_1, v_2\} \in \mathbb{E} : d_{v_1,v_2} \leq d_{\min}$

$$l_e = 1 \tag{6.45}$$

With

$$\gamma_{v_1} \in [0, 2\pi] \subset \mathbb{R} \tag{6.46}$$

$$f_{d,p}, \delta_d \in [0, 1] \subset \mathbb{R} \tag{6.47}$$

$$u_e \in [0, H] \subset \mathbb{R} \tag{6.48}$$

$$l_e, b_{v_1,v_2}, b'_{v_1,v_2}, b''_{v_1,v_2} \in [0, 1] \subset \mathbb{N} \tag{6.49}$$

$$a'_{v_1,v_2} \in \left[-\beta_{v_1,v_2} + \frac{\alpha}{2} + 1, -\beta_{v_1,v_2} + \frac{\alpha}{2} + 1 + 2\pi\right] \tag{6.50}$$

$$a''_{v_1,v_2} \in \left[\beta_{v_1,v_2} + \frac{\alpha}{2} + 1 - 2\pi, \beta_{v_1,v_2} + \frac{\alpha}{2} + 1\right] \subset \mathbb{R} \tag{6.51}$$

$$c'_{v_1,v_2} \in \left[-\beta_{v_1,v_2} + \frac{\alpha}{2} + 1, -\beta_{v_1,v_2} + \frac{\alpha}{2} + 1 + 2\pi\right] \subset \mathbb{R} \tag{6.52}$$

$$c''_{v_1,v_2} \in \left[-4\pi + \beta_{v_1,v_2} + \frac{\alpha}{2} + 1, \beta_{v_1,v_2} - 2\pi + \frac{\alpha}{2} + 1\right] \subset \mathbb{R} \tag{6.53}$$

$$d'_{v_1,v_2} \in \left[-\beta_{v_1,v_2} + \frac{\alpha}{2} - 2\pi + 1, -\beta_{v_1,v_2} + \frac{\alpha}{2} + 1\right] \subset \mathbb{R} \tag{6.54}$$

$$d''_{v_1,n_2} \in \left[\beta_{v_1,v_2} + \frac{\alpha}{2} - 2\pi + 1, \beta_{v_1,v_2} + \frac{\alpha}{2} + 1\right] \subset \mathbb{R} \tag{6.55}$$

Optimization Goals

The model presented above does not contain any objective functions, it represents only the constraints needed to steer the main lobes to establish a link. Consequently we will present and discuss various optimization goals on top of this MIP.

Node Degree Maximization In the first paper **[HKV08]** and Vilzmann's Ph.D. thesis [Vil09], the goal was to find an optimal bound for the MNDB algorithm. To this end, they simply maximized the number of links in a given scenario

$$\max \sum_{e \in \mathbb{E}} l_e. \tag{6.56}$$

In exemplary studies they observed by comparing the resulting average node degrees (i. e. the value of (6.56) divided by the number of nodes $|\mathbb{V}|$), that there is indeed still a lot of room for improvment. Nevertheless, the underlying larger question, on how much connectivity can be improved by beamforming antennas remained unanswered – despite achieving substantially higher node degrees with the MIP, some scenarios had a path probability smaller than one, as we will demonstrate again in the case study.

Path Probability In order to close this gap in knowledge, we devise a "true" path probability optimization scheme. We introduce the notion of connectivity via a classic MCF-formulation on top of the presented link model. Using the flow approach as presented in 3.3.2, we create flows (with a commodity of 1) from every node v_s to every other node $\mathbb{V}\setminus\{v_s\}$. Hence, the sum of the outgoing flows f^{v_s} of v_s has to be $|\mathbb{V}| - 1$

$$\sum_{v_2 \in \mathbb{V}_{v_s}} f^{v_s}_{v_s, v_2} = |\mathbb{V}| - 1 \qquad \forall v_s \in \mathbb{V}. \tag{6.57a}$$

As usual, we denote by $\mathbb{V}_{v_s}$ the set of adjacent nodes of v_s. On all other nodes only one commodity (i. e. 1) is allowed to end, while all others have to be passed on towards their respective destinations. This means that the sum of the incoming and outgoing flows can only differ by a difference of 1:

$$-\sum_{v_1 \in \mathbb{V}_{v_t}} f^{v_s}_{v_1, v_t} + \sum_{v_2 \in \mathbb{V}_{v_t}} f^{v_s}_{v_t, v_2} = -1 \qquad \forall v_s \in \mathbb{V}, \forall v_t \in \mathbb{V}\setminus\{v_s\} \tag{6.57b}$$

In order to be useable by the flows, a link has to be established, which we ensure with

$$\sum_{v_s \in \mathbb{V}} f^{v_s}_{v_1, v_2} \leq (|\mathbb{V}| - 1)\, l_{v_1, v_2} \qquad \forall \{v_1, v_2\} \in \mathbb{E}. \tag{6.58}$$

The left-handed side of Inequality (6.58) represents the flow over the edge $\{v_1, v_2\}$. The right-handed side offers a capacity of $(|\mathbb{V}| - 1)$, *iff* the link is established, i. e. $l_{v_1, v_2} = 1$.

Quite obviously, the above MIP will only be feasible[8] if a given scenario can be connected at all, which we found not to be the case in a non-negligible number of scenarios. In such a situation, we are still interested in the highest possible path probability. To ensure feasibility we introduce *dummy variables* $\delta^{v_s}_{v_t} \in [0; 1]$ in the flow constraints

[8] In this case, we could solve the problem as a *Constraint Program*, i. e. without any further objective function.

such that a flow can either be routed via the link-flow variables, or in case this is not possible over the dummy variables.

$$\sum_{v_2 \in \mathbb{V}_{v_s}} f^{v_s}_{v_s,v_2} + \sum_{v_t \in \mathbb{V} \setminus \{v_s\}} \delta^{v_s}_{v_t} = |\mathbb{V}| - 1 \qquad \forall v_s \in \mathbb{V} \tag{6.59a}$$

$$-\sum_{v_1 \in \mathbb{V}_{v_t}} f^{v_s}_{v_1,v_t} + \sum_{v_2 \in \mathbb{V}_{v_t}} f^{v_s}_{v_t,v_2} - \delta^{v_s}_{v_t} = -1 \qquad \forall v_s \in \mathbb{V}, \forall v_t \in \mathbb{V} \setminus \{v_s\} \tag{6.59b}$$

Obviously our goal is to accomodate as many demands as possible without using the dummy variables, i. e. minimizing the number of dummies larger than zero

$$\min \sum_{v_s \in \mathbb{V}} \sum_{v_t \in \mathbb{V} \setminus \{v_s\}} \delta^{v_s}_{v_t} \tag{6.60}$$

As it turned out during our case study, if a scenario can be configured to be connected, there are usually many possibilities to achieve this. Consequently, there is still room for additional objectives, which can prefer some solutions over others. While there are no heuristics at the moment which achieve path probabilities, where additional objectives would be sensible, we can use auxiliary optimization goals to asses theoretical advantages of beamforming antennas.

Path Probability with Additional Objectives

Node Degree Maximization As Vilzmann argues [Vil09, p. 22], a densely meshed network (i. e. a high average node degree) might offer higher resilience against link and node failures (as long as they are uncorrelated). Furthermore, in case broadcasting is a noteworthy function of the network, densely meshed networks offer faster broadcasting. On the other hand interference and routing overhead might be problematic (depending on the used MAC-protocol, . . .) in case the network is under high load.

Integrating this objective in our model is straightforward: We need to extend the objective function to take the number of links into account. Hence (6.60) becomes

$$\min \sum_{v_s \in \mathbb{V}} \sum_{v_t \in \mathbb{V} \setminus \{v_s\}} \delta^{v_s}_{v_t} - A \sum_{e \in \mathbb{E}} l_e. \tag{6.61}$$

Obviously we have to dimension the weight of the link maximization according to our aims. If we only want to consider node degree maximization on top of path probability maximization a suitable choice for A will be $A = \frac{1}{\mathbb{E}}$, i. e. all links are as expensive as a single dummy variable. Thus, the solver will as a primary objective minimize the used dummies and only as a secondary objective maximize the number of $l_e = 1$.

Link Minimization Arguing in the opposite direction as in the previous paragraph, if interference is a limiting factor in the network, it will be desireable to reduce the number of links. Scarce resources for routing on the nodes might be another reason to keep the number of links as low as possible. Our MIP offers two choices, as to how the number of links can be reduced. First, we can simply modify the previous objective (6.61) and switch the sign of A. In this case, the optimizer will only set as few $l_e = 1$ as necessary to achieve a connected network. What will not happen however, is that main lobes will be steered away from other nodes. Consequently, our result will only be of value in those cases, where we are interested in limiting the number of links we are actively using, e. g. to save CPU capacities by simplifying the routing.

If our goal however is to actively steer main lobes away to reduce interference we need to employ a different modification of our MIP. Reinspecting the constraints for link indication, only Equations (6.23) and (6.25) respectively have the ability to force main lobes away. We recall, that a value larger than one for a'_{v_1,v_2} and a''_{v_1,v_2} indicates that the link condition is fulfilled. In contrast a value smaller than one indicates that the condition is not met. Hence, our means to achieve fewer links is to minimize these variables on top of maximizing path probability. Thus, our objective has to be extended to

$$\min \sum_{v_s \in \mathbb{V}} \sum_{v_t \in \mathbb{V} \setminus \{v_s\}} \delta_{v_t}^{v_s} + B \sum_{\{v_1;v_2\} \in \mathbb{E}} \left(a'_{v_1,v_2} + a''_{v_1,v_2} \right). \tag{6.62}$$

Again the weighting factor B has to be dimensioned and similarly to (6.61), the reciprocal of the minimal value of all a'_{v_1,v_2} and a''_{v_1,v_2} will be a suitable choice. It is worth noting that this minimization might make suboptimal trade-offs in the link minimization: Since it does not maximize the number of a'_{v_1,v_2} and a''_{v_1,v_2} which are below zero, but tries to minimize the sum of these variables, it might choose not to set some variables below 1, in order to set others further below 1. In case this also should be avoided, we would need to introduce variables following the same principle as the b'_{v_1,v_2} and b''_{v_1,v_2} just in the "opposite" direction. Due to the number of necessary variables, this would considerably increase the complexity of the MIP, which is why we omitted this objective.

Average Path Length Minimization One of the proposed additional objectives in this section maximized the average node degree in order to achieve a densely meshed network. Within this paragraph, we will devise an objective with a closely related aim, namely to minimize the average path length[9]. If we assume, that all demands are routed along the (hop-wise) shortest path, this objective will gain the best global configuration, provided that all demands are equally important. In order to achieve this goal, we exploit the fact that the routed commodities (at least as a Gedankenmodell) consume a capacity of one along the edges they are using. Since we do not impose a limit on them (every edge has sufficient capacity to hold all demands), minimizing the total used capacity in addition to providing a connected configuration, is equivalent to finding the connected configuration, where the average shortest path

[9] Which is sometimes also called *network diameter*.

is minimal. The capacity along an edge $\{v_1, v_2\}$ can be gained by summing over the corresponding edge-flow variable of a demand via $\sum_{v_s \in \mathbb{V}} f^{v_s}_{v_1,v_2}$. Again, we can simply add this sum (for all edges) to the objective function

$$\min \sum_{v_s \in \mathbb{V}} \sum_{v_t \in \mathbb{V} \setminus \{v_s\}} \delta^{v_s}_{v_t} + C \sum_{v_s \in \mathbb{V}} \sum_{e \in \mathbb{E}} f^{v_s}_e, \tag{6.63}$$

and by dimensioning C to the reciprocal of the maximal value of the last term, i. e. $\frac{1}{|\mathbb{V}| \cdot |\mathbb{E}|}$, we ensure that path probability is still our top priority.

k-Connectivity Last, we want the resulting configuration not only to be connected, but k-connected, which simply means that k edge-disjoint routes have to exist for every demand relation. k-connectivity can be achieved by rather easy means, however we have to go back to a classic flow approach. For notational convenience, we create a fully-meshed demand graph $\mathcal{D}(\mathbb{V}, \mathbb{D})$. Assuming that a demand $d \in \mathbb{D}$ starts at v_s and terminates at v_t, we can formulate a classic flow approach as presented in 3.3.2

$$\sum_{v_2 \in \mathbb{V}_{v_s}} f^d_{v_s,v_t} + k \cdot \delta_d \geq k \qquad \forall d \in \mathbb{D} \tag{6.64a}$$

$$\sum_{v_2 \in \mathbb{V}_v} f^d_{v,v_2} - \sum_{v_1 \in \mathbb{V}_v} f^d_{v_1,v} = 0 \qquad \forall d \in \mathbb{D}, \forall v \in \mathbb{V} \setminus \{v_s, v_t\} \tag{6.64b}$$

$$-\sum_{v_1 \in \mathbb{V}_{v_t}} f^d_{v_1,v_t} - k \cdot \delta_d \leq -k \qquad \forall d \in \mathbb{D} \tag{6.64c}$$

$$\min \sum_{d \in \mathbb{D}} \delta_d. \tag{6.65}$$

In order to ensure k-connectivity, we set the demand to k (e. g. 1 if we want a connected graph, 2 if we want a two-connected graph, ...) as shown above and limit $f^d_{v_1,v_2}$ to 1. Due to this, the solver is forced to search for k disjoint paths as long as this is possible.

6.2.3 Large Scale Optimization Model

During our case study, we experienced performance problems with some of our auxiliary objectives: While path-probability maximization and k-connectivity maximization usually finished in appropriate time frames (minutes), we observed a severe performance degradation with the minimization of the average path length and the node degree maximization (which could not be finished within one day on commodity hardware).

Considering our positive experiences with path-approaches and column generation, we employed the same techniques as already demonstrated in Sections 4.2 and 5.2 respectively to handle those harder cases.

Path Approach

The first step towards the outlined approach is to reformulate the problem in a path-flow fashion. We will again use a fully meshed demand graph $\mathcal{D}(\mathbb{V}, \mathbb{D})$. Now for every demand we consider path flows $f_{d,p}$ flowing along a path p. Consequently the sum of all path flows has to match our demand (similar to the previous section, our demand is 1 in case we desire a connected graph, 2 in case of a 2-connected graph and so on)

$$[\pi_d] \qquad \sum_{p\in\mathbb{P}_d} f_{d,p} + k\cdot\delta_d \geq k \qquad \forall d\in\mathbb{D}. \tag{6.66a}$$

We use dummy variables δ_d to keep the problem feasible and gain the maximum path probability in those cases where the network cannot be connected. We can calculate the sum of all flows over a given edge e by summing over all path-flows using that edge (i. e. $e \in p$). Obviously, in order to be able to use a link e, this link has to be established, that is the corresponding link variable $l_e = 1$:

$$[\sigma_e] \qquad \sum_{d\in\mathbb{D}} \sum_{\substack{p\in\mathbb{P}_d:\\ e\in p}} f_{d,p} \leq |\mathbb{D}|\cdot l_e \qquad \forall e\in\mathbb{E}. \tag{6.66b}$$

The objective functions in question are straightforward to translate:

- Path probability maximization:

$$\min \sum_{d\in\mathbb{D}} \delta_d \tag{6.67}$$

- Path probability and node degree maximization:

$$\min \sum_{d\in\mathbb{D}} \delta_d + A\sum_{e\in\mathbb{E}} l_e \tag{6.68}$$

- Path probability and node degree minimization:

$$\min \sum_{d\in\mathbb{D}} \delta_d + B \sum_{\{v_1;v_2\}\in\mathbb{E}} \left(a'_{v_1,v_2} + a''_{v_1,v_2}\right) \tag{6.69}$$

- Path probability with average path length minimization:

$$\min \sum_{d\in\mathbb{D}} \delta_d + C \sum_{d\in\mathbb{D}} \sum_{p\in\mathbb{P}_d} \sum_{e\in p} f_{d,p} \tag{6.70}$$

- k-Connectivity:
  After adjusting k to the desired degree of connectivity in the demand constraint (6.66a), we limit the amount of flow a demand d can send over edge e to 1

$$\sum_{\substack{p\in\mathbb{P}_d:\\ e\in p}} f_{d,p} \leq 1 \qquad \forall d\in\mathbb{D}, \forall e\in\mathbb{E} \tag{6.71}$$

and minimize the number of used dummy variables as before

$$\sum_{d \in \mathbb{D}} \delta_d. \tag{6.72}$$

Quite unsurprisingly, the path approach in itself is close to being unmanageable in the considered networks: All possible paths via all possible links have to be considered and this number is quite large even for our smallest 30 node scenarios.

6.2.4 Optimization Algorithm

Since the column generation approach is very similar to those previously presented in 4.2.3, we will only briefly describe it this time. We can calculate the reduced costs of a path-flow variable $f_{d,p}$ in the dual system consisting of the corresponding dual variables annotated in angular brackets in Equations (6.66)

$$\pi_d - \sum_{e \in p} \sigma_e \leq 0. \tag{6.73}$$

If we start our optimization with a small subset of the possible paths (or no paths at all since we have dummy variables keeping the MIP feasible even without path-flow variables), a path p not yet in the MIP, which would howevever improve the solution, will violate the above Inequality (6.73), because $\sum_{e \in p} \sigma_e$ will be smaller than π_d thus rendering the whole expression positive. Hence, if we can find a path with this property, we know that it will improve the solution. This path search can again be solved via a shortest-path problem with link-weights σ_e. Once we do not find any further improving paths, we know that we have found the optimal solution, i. e. considering additional paths will not result in a smaller objective value.

6.2.5 Case Study

Scenario Generation

Since adhoc networks are – as their name strongly suggests – unplanned and thus of highly stochastic nature, reference scenarios as employed in the previous chapters are absent. A significant body of research thus assumes homogeneous node distributions of a given density and generates scenarios based on this assumption, i. e. sets of node locations. In other scenarios the nodes are highly mobile and movements of the individual nodes have to be taken into account via *mobility models*. Recalling our envisioned environment, namely urban access networks, we consider this to be a rather bad match: Urban structures (such as houses, roads, ...) simply do not allow a homogeneous distribution of users in many cases. In these applications users tend to form *clusters*, that is they aggregate in groups (i. e. in squares, congress halls, ...). In order to accomodate this notion, we generate clustered scenarios via the following algorithm presented in [Vil09]:

1. On a square area of a given size (in our case $1\,\text{km} \times 1\,\text{km}$), select the locations of a given number of cluster centres (in our studies 5) following a uniform distribution.
2. Associate nodes evenly to cluster centres and locate nodes around these cluster centres following a normal distribution with a standard deviation of 10% of the area length, i. e. in our case $1\,\text{km}$.

For the statistical analysis, we created 100 30 node scenarios, for the exemplary analysis, we generate each one 30, 40 and 50 node scenario.

We assume the nodes to be situated in a perfectly planar environment. Thus, we can calculate the distance d_{v_1,v_2} between two nodes v_1 and v_2 with respective coordinates (x_{v_1}, y_{v_1}) and (x_{v_2}, y_{v_2}) simply by applying the Pythagoras' theorem as

$$d_{v_1,v_2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \tag{6.74}$$

Implementation

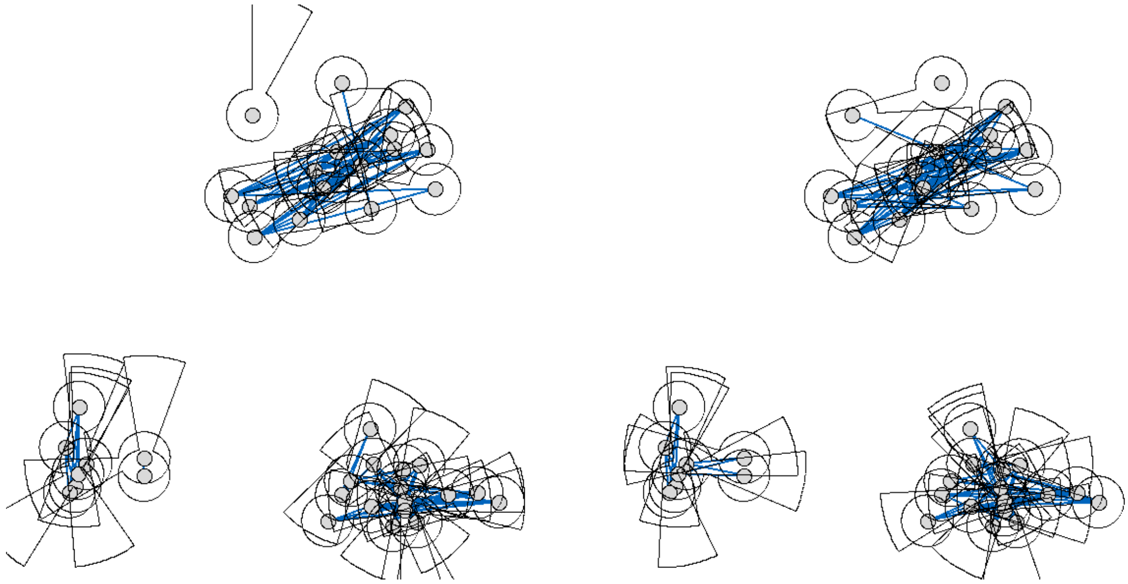
We implemented the above defined MIP in C++ using SCIP [Ach07] as Branch-and-Bound framework and CPLEX 9.13 [CPL09] as LP-solver. We used GRAPH [Lay09] for data representation and its implementation of Dijkstra's algorithm. In order to generate the random variables for the shadow fading, we used the Mersenne Twister 19937 by Matsumoto and Nishimura [MN98] as implemented by the GNU Scientific Library (GSL) Version 1.9 [gsl].

Visualization of the results was performed by PANTS, the tool developed by Vilzmann in his Ph.D. thesis [Vil09].

Exemplary Comparison of Objective Functions

We start our illustration of the different objective functions proposed in Section 6.2.2 by reproducing earlier results from **[HKV08]**. As stated earlier the goal in this work was to compare the node degree achieved by MNDB to a theoretical optimum achieved by a MIP. As we can see in Figure 6.5a, MNDB manages to achieve a relatively densely meshed network, however the three clusters and one node above the upper cluster remain isolated. The node degree achieved by the optimization is even higher (2.8 vs 2.5), however the three islands are still separated, the interesting question being whether this can be changed.

Figure 6.6 containing the results of the path probability optimizations can clearly answer this question positively. As we have already stated during the construction of our optimization model, we found out that we still have a considerable degree of freedom even if we require the resulting network configuration to be connected. Hence Figure 6.6b is still a valid solution for the original path maximization problem and the secondary objective of a maximum average node degree does not impede our original



(a) MNDB (b) Maximum Average Node Degree

Figure 6.5: Node Degree Maximizations for the 40 Node Scenario **[HKV08]**

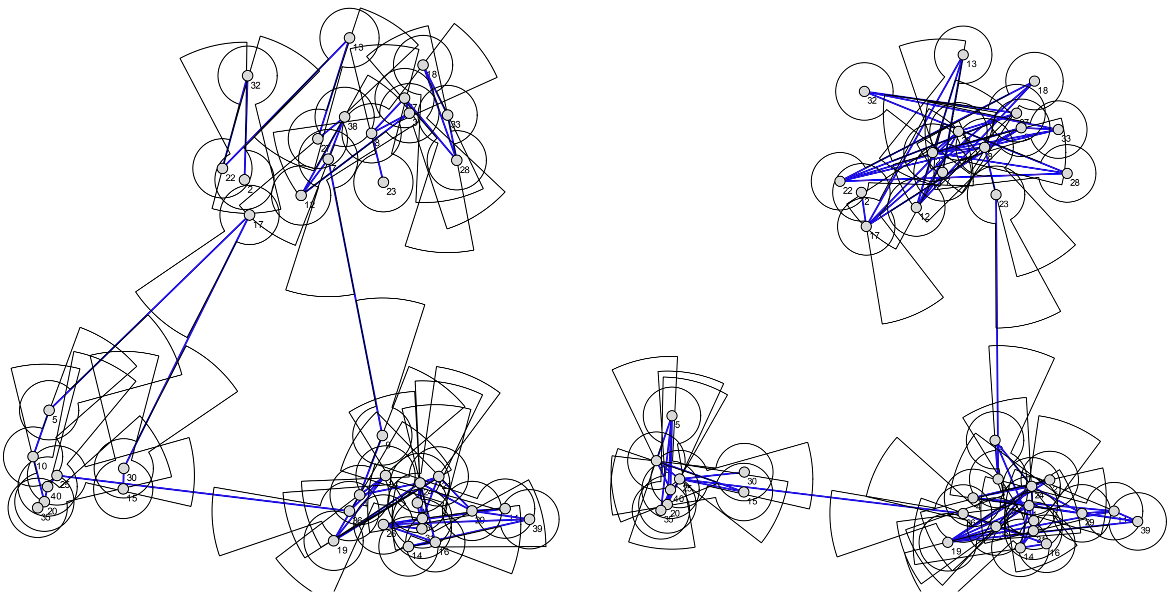
objective. Nevertheless, we can see a clear difference between Figure 6.6a and Figure 6.6b: We only have very few long links (which usually lowers the node degree of the nodes involved), but more keyholes pointing towards the centres of the respective clusters.

A 50 node scenario in Figure 6.7 clearly illustrates the effect of the additional average path length minimization. Figure 6.7a again depicts the "simple" path probability maximization and Figure 6.7b the additional goal of a minimum average path (hop-)length between the nodes. We can see that in the second case many more links "across" the network exist which can serve as a "shortcut".

With these examples, we wanted to illustrate that despite asking for a connected network, we can still heavily influence the resulting network properties. We must state however that at this moment no distributed methods to achieve these additional goals let alone to achieve connectivity are available, which is why we will not elaborate on the other goals presented in the optimization model and instead focus on evaluating the existing methods at hand.

6.2.6 Statistical Analysis

To this end, we generated a set of one hundred independent 30 node scenarios as described in Section 6.2.5. Every one of the scenarios was examined for transmission powers of 20 dBm, 19 dBm, 18 dBm and 17 dBm. We optimized the connectivity in these sets with MNDB and the presented MIP-models to gain an optimal maximum node degree, and a maximum path probability respectively. In this setting we can



(a) Maximum Path Probability

(b) Maximum Path Probability with Maximum Average Node Degree

Figure 6.6: Path Probability Optimizations for the 40 Node Scenario

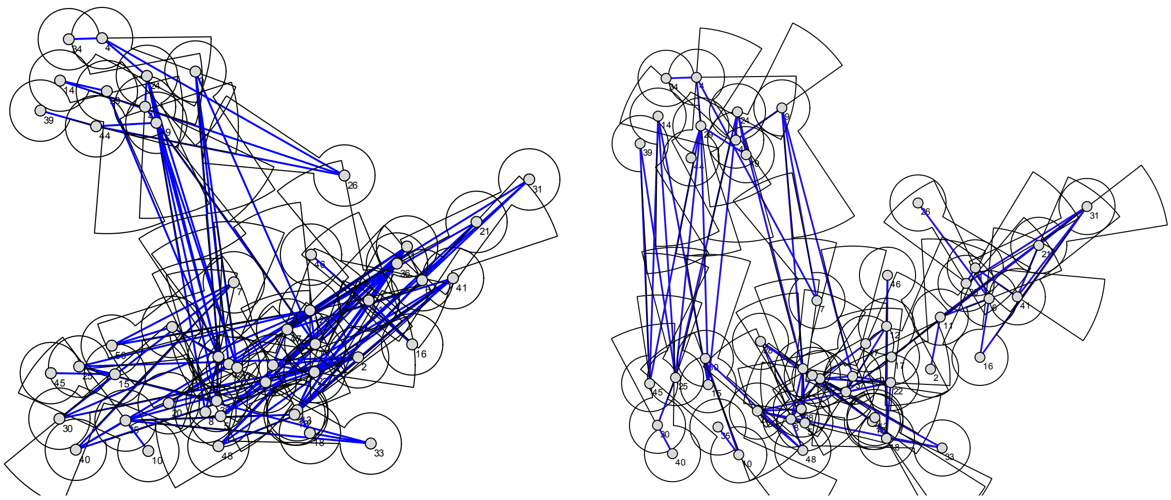
assess whether a network can be connected at all, how well MNDB worked and how well it would work, should it optimally solve its optimization goal (to find the maximal node degree). Figure 6.8a reveals that MNDB succeeded to gain a connected network in a very low number of scenarios, despite the fact that it would have been possible. We can also safely state that the goal of finding the highest average node degree is considerably far away from a high path probability – even with a successful optimization our network remains unconnected.

An inspection of Figure 6.8a shows that despite earlier findings of Vilzmann [Vil09] and Hartmann, Kiese and Vilzmann **[HKV08]**, where a significant gap in the resulting node degree was found between MNDB and an optimal solution, the resulting average path probability is actually quite close.

Therefore, we conclude that further (heuristical) optimization of the node degree will most likely *not* lead to a higher path probability making future research on other optimization goals for distributed heuristics attractive.

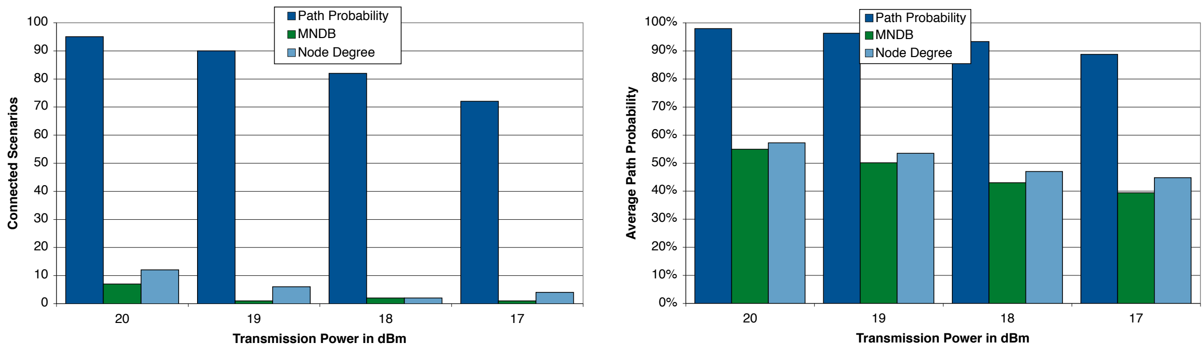
Shadow Fading

In the last section of this case study, we want to examine the influence of shadow fading on the connectivity of the overall network. To this end, we consider shadow-fading from 0 to 12 dB in 2 dB-steps for the 100 scenarios generated for the previous section and a transmission power of 20 dB. Integration of shadow fading in the optimization scheme is straightforward: All we need to do is add the shadow fading



(a) Maximum Path Probability (b) Maximum Path Probability with Minimum Average Path Length

Figure 6.7: Optimization Results for the 50 Node Scenario



(a) Connected Scenarios (b) Average Path Probability

Figure 6.8: Statistical Analysis over 100 Random Scenarios

variable X to our path loss, thereby enhancing the link constraint (6.28) to

$$(P_t - P_{r,\min} - P_{L,v_1,v_2} - \mathsf{X} + 2 \cdot G_{\mathrm{S}})\, l_{v_1,v_2} + (G_{\mathrm{M}} - G_{\mathrm{S}})\, b_{v_1,v_2} + (G_{\mathrm{M}} - G_{\mathrm{S}})\, b_{v_2,v_1} \geq 0. \tag{6.75}$$

It is important to keep in mind that this causes $d_{\min}$ and d_{max} to be individual properties of a node-pair. Again we examine the number of scenarios we can connect with our path probability optimization in Figure 6.9. Bettstetter and Hartmann found shadow fading to increase the connectivity with omnidirectional antennas in [BH05]. Given the results above, where our networks are connected in more than 90% of the scenarios, it is difficult to come to such a general statement. What we can safely state however is that shadow fading even with large variances (such as experienced in the WOBAN scenario of the first section), does not harm the theoretical connectivity of the beam-forming nodes, thus suggesting that beamforming could indeed improve the path probabilities in this rather difficult setting.

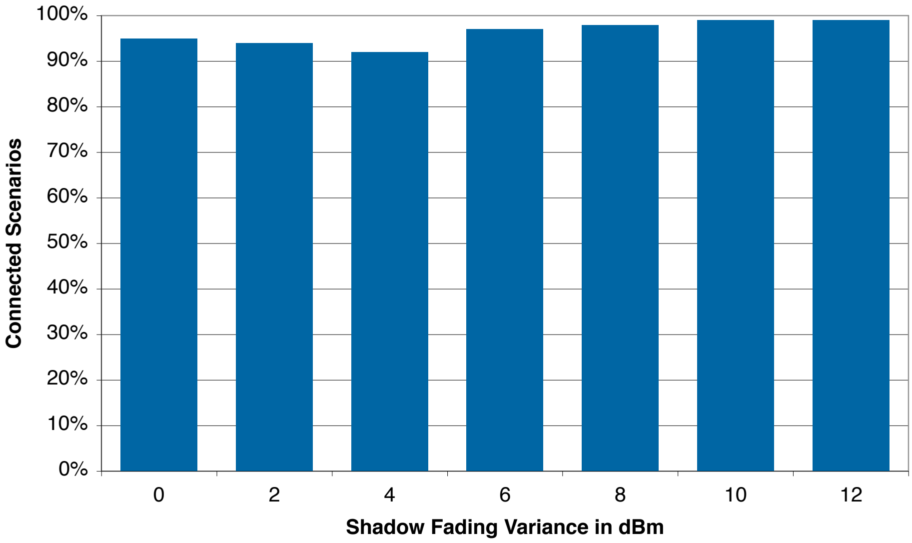


Figure 6.9: Connected Scenarios with Shadow Fading

Optimization Performance

As can already be concluded by the number of problems from our statistical analysis, computation times are quite manageable. For the 30 node scenarios and pure path probability maximization, we could solve most of the problems well below 15 minutes, with a general tendency, that the scenarios took longer to solve with lower transmission powers.

The presented flow approach reached a computational limit, once we tried to integrate our additional objectives, which could be solved impressively fast by the path approach with column generation, while the path approach exhibited worse perfomance than the flow approach in some cases without additional objectives. An exemplary comparison was presented in **[KHL09]**.

6.3 Summary

Following a study on connectivity based on measurements in a real-life wireless mesh network, we have developed a MIP-formulation to optimize the connectivity in a wireless multi-hop network with beamforming antennas. With this approach we were able to find a beamforming configuration to gain a connected network if this possibility existed. Using column-generation, we could increase the computational efficiency to consider additional objectives on top of our foremost goal, connectivity. While this approach is surely too complex to be realized in an adhoc network, where the nodes have no global information and are usually in a rapidly changeing environment, such a planning might still be suitable for wireless mesh access-networks (such as WOBAN), which are quite static. Nevertheless, the results of the optimization can be used to assess the performance of existing algorithms, as we showed with MNDB.

In a statistical analysis we showed, that there is indeed still considerable room for improvement and could also show, that even if MNDB could reach it's optimization goal (the maximum node degree), it would quite likely not result in a higher path probability. A final study could show that shadow fading does not have a significant negative influence on the overall connectivity in a beam-forming wireless mesh network.

Open Issues

While this work made an assessment of existing and future heuristic algorithms possible, it can be accounted as a direct contribution only for wireless mesh access networks, where the mobility of the base stations is limited and hence the effort of the optimization is reasonable. Thus, especially in the case of adhoc networks with beam-forming antennas, improved distributed heuristics to achieve better connectivity are needed. A further interesting point would be an integration in the evaluation framework developed by Utschick et al. [10], which will give hints in which cases the goal of a connected network will be fruitful at all and in which scenarios a different optimization goal would be more rewarding.

[10] Private presentation in August 2009

7 Conclusions

"Wovon man nicht sprechen kann, darüber muss man schweigen."[1]

This work contributed to the research on network planning by extending its scope to new challenges, finding efficient algorithms to solve the arising planning problems and by applying current mathematical research to realistic planning problems.

In Chapter 4, we developed a flexible heuristic to plan grooming problems in optical networks in solution times adequate for a more dynamic network or even on-line planning. We demonstrated that the basic principle of our algorithm can be easily applied to other far more complex problem structures. We furthmore evaluated upcoming changes in the transmission technologies in optical networks which lead to adaptive link speeds. With a MIP-formulation we could show that these adaptive link capacities have a positive impact on the transponder costs of a network.

In Chapter 5, we specified a new branch-and-price approach for planning SBPP. An implementation of this algorithm was shown to be scalable to problem sizes, which are clearly out of scope for standard state-of-the-art solvers. We furthermore proposed the use of dual-homing for dual-link failure protection in backbone networks and could show that it is indeed possible to protect all demands against dual-link failures at moderate costs with this methodology. In the last section, we presented a planning method to place measurement equipment in transparent or translucent optical networks to facilitate failure localization.

In Chapter 6, we showed that connectivity will be a challenge in possible future wireless access networks. We thoroughly studied the influence of beamforming antennas on the connectivity of a wireless mesh network. Our MIP-based planning approach allowed us to find optimal bounds on the connectivity and hence enabled us to assess both the efficienciy of existing and the potential of future heuristic approaches to steer these beamforming antennas.

Since we have already presented open issues during the summaries at the end of each chapter, we would like to point to some more general issues.
First of all, only in the latest research proposals, operational issues of networks are taken into account during the development of new technologies. We would like to extend this notion to network planning as well: A lot of research during the last years focussed on solving larger and more and more complex models often incorporating multiple layers or multiple objectives at once. Quite obviously, a solution generated

[1] Ludwig Wittgenstein, *Tractatus logico-philosophicus*

by these approaches is very efficient, however the interdependencies between formerly separate layers make them very difficult to realize without further support from the management software – which has traditionally been oriented towards a single layer (only recently multi-layer approaches like GMPLS have appeared). The question that arises from this setting (multi-layer optimization and integrated multi-layer network management) is how tolerant these solutions will be regard imprecise input data (such as traffic forecasts). The mathematical methodology to evaluate these questions (sometimes also dubbed as *robust planning*) is at least performance-wise insufficient.

Furthermore, traffic engineering as it is right now (which we condense to "the ability to setup tunnels with guaranteed bandwidth") requires an hideous amount of planning especially when protection capacity has to be shared and once this planning has taken place, additional demands or traffic peaks might require further planning because we could not take the uncertainty of our demands into account during the first planning phase. Automation to be able to cope with at least moderate traffic variations (besides simple overprovisioning as it done now) might be a more efficient way than huge centrally controlled entities. This however will by no means lead to the extinction of network planning — planning for worst-case behaviour of these algorithms, the underlying dimensioning, connections between the different networks, topology planning, performance evaluations, ... All these questions cannot be answered with certainty without reliable network planning.

Reference Networks

Network Name	Nodes	Edges	Demands	Source
COST239	11	52	1 – 110	COST 239
Germany 17	17	26	121	Nobel/SNDLib
Germany 50	50	88	662	EIBONE/SNDLib
Nobel US	14	21	91	Nobel/SNDLib
Norway	27	51	702	SNDLib

Acronyms

ADSL Asymmetric Digital Subscriber Line

APS Automated Protection Switching

ATM Asynchronous Transfer Mode

BS Base Station

BPSK Binary Phase-Shift Keying

CAPEX Capital Expenditures

CEF Cisco Express Forwarding

DVB-S Digital Video Broadcasting – Satellite

DWDM Dense Wavelength Division Multiplex

EIBONE Efficient Integrated Backbone

FCC Federal Communications Commission

FDM Frequency Divisioned Multiplex

FFT Fast Fourier Transformation

FRR Fast ReRoute

GRASP Greedy Randomized Adaptive Search Procedure

IETF Internet Engineering Task Force

IGP Interior Gateway Protocol

ILP Integer Linear Program

IP Integer (Linear) Program

IPv4 Internet Protocol

ISO International Standards Organization

ISP Internet Service Provider

IS-IS Intermediate System - Intermediate System

LAN Local Area Network

LP Linear Program

MAC Medium Access Control

MCF Multi-Commodity Flow

MIP Mixed-Integer Linear Program

MNDB Maximum Node Degree Beamforming

MPLS Multiprotocol Label Switching

MS Mobile Station

MTBF Mean Time Between Failures

MTD Maximum Transmission Distance

MTTR Mean Time to Repair

NSRC Network Reliability Steering Committee

QPSK Quadrature Phase-Shift Keying

OFDM Orthogonal Frequency Division Multiplex

OOFDM Optical Orthogonal Frequency Division Multiplex

OPEX Operational Expenditures

OSA Optical Spectrum Analyzer

OSI Open Systems Interconnect

OSNR Optical Signal-to-Noise Ratio

OSPF Open Shortest Path First

OTU Optical Transport Unit

PD Path Diversity

PDH Plesiosyncronous Hierarchy

PON Passive Optical Network

POTS Plain Old Telephone Service

RDB Random Direction Beamforming

RMP Restricted Master Problem

RSVP Resource ReSerVation Protocol

RWA Routing and Wavelength Assignment

SBPP Shared Backup Path Protection

SDH Synchronous Digital Hierarchy

SNR Signal-to-Noise Ratio

SONET Synchronous Optical Networking

SRLG Shared Risk Link Group

T-MPLS Transport MPLS

TNDB Two-Hop Node Degree Beamforming

VPN Virtual Private Network

WDM Wavelength Division Multiplex

WLAN Wireless LAN

WOBAN Hybrid Wireless-Optical Broadband-Access Network

List of Figures

List of Tables

Bibliography

Publications by the Author

[BSR+09] Adriana Bocoi, Matthias Schuster, Franz Rambach, Moritz Kiese, Christian-Alexander Bunge, and Bernhard Spinnler. Reach-dependent capacity in optical networks enabled by OFDM. In *Proceedings of the Optical Fiber Communication Conference and Exposition (OFC)*, San Diego, CA, USA, March 2009.

[EKW08] Jörg Eberspächer, Moritz Kiese, and Roland Wessäly. Optimierung von Kommunikationsnetzen. In Martin Grötschel, Klaus Lucas, and Volker Mehrmann, editors, *Produktionsfaktor Mathematik*. Springer, Berlin Heidelberg, 2008. 1, 17

[GK06] Claus Gruber and Moritz Kiese. Optimization of resilient networks with column generation. In *Proceedings of the INFORMS Telecommunications Conference*, Dallas, TX, USA, March 2006. 75, 79

[HKV08] Christian Hartmann, Moritz Kiese, and Robert Vilzmann. Optimizing the node degree in wireless multihop networks with single-lobe beamforming. In *Proceedings of the INFORMS Telecommunications Conference*, Mar 2008. 113, 122, 128, 129, 130, 143

[KA07] Moritz Kiese and Achim Autenrieth. A fast heuristic for DWDM routing and grooming. In *Proceedings of the International Workshop on Design and Reliable Communication Networks (DRCN)*, La Rochelle, France, October 2007. 54

[KGS+09] Moritz Kiese, Elisabeth Georgieva, Dominic Schupke, Biswanath Mukherjee, and Jörg Eberspächer. Availability evaluation of hybrid wireless optical broadband access networks. In *Proceedings of the IEEE International Conference on Communications (ICC) Optical Networks and Systems Symposium*, Dresden, Germany, June 2009. 109, 111

[KHL09] Moritz Kiese, Christian Hartmann, and Julian Lamberty. Optimality criteria for the connectivity of adhoc networks with beamforming antennas. In *Proceedings of the International Network Optimization Conference (INOC)*, Pisa, Italy, April 2009. 132

[KHLV09] Moritz Kiese, Christian Hartmann, Julian Lamberty, and Robert Vilzmann. On connectivity limits in adhoc networks with beamforming antennas. *EURASIP Journal on Wireless Communications and Networking*, Optimization Techniques in Wireless Communications, 2009. 119

[KHV09] Moritz Kiese, Christian Hartmann, and Robert Vilzmann. Optimality bounds of the connectivity of adhoc networks with beamforming antennas. In *IEEE Global Communications Conference (GLOBECOM) Ad Hoc Sensor and Mesh Networking Symposium*, Honolulu, Hawaii, USA, Nov 2009.

[KM07] Moritz Kiese and Carmen Mas Machuca. Optimal placement of different types of monitoring equipment in transparent optical networks. In *Proceedings of the Workshop on Monitoring, Attack Detection and Mitigation (MONAM)*, Toulouse, France, November 2007. 103

[KMSE09] Moritz Kiese, Velislava Marcheva, Dominic Schupke, and Jörg Eberspächer. Diverse routing based on shared risk link groups. In *Proceedings of the International Workshop on Design and Reliable Communication Networks (DRCN)*, Washington DC, USA, October 2009. 52, 77

[KS08] Moritz Kiese and Thomas Stidsen. A general model for path-based protection methods. In *Proceedings of the INFORMS Telecommunications Conference*, College Park, MD, USA, March 2008. 79

[KS09] Moritz Kiese and Matthias Schuster. Exploiting transponder performance in optical OFDM networks. In *Proceedings of the Optical Fiber Communication Conference and Exposition (OFC)*, San Diego, CA, USA, March 2009.

[MK07] Carmen Mas Machuca and Moritz Kiese. Optimal placement of monitoring equipment in transparent optical networks. In *Proceedings of the International Workshop on Design and Reliable Communication Networks (DRCN)*, La Rochelle, France, October 2007.

[SBK+09] Matthias Schuster, Adriana Bocoi, Moritz Kiese, Franz Rambach, Christian-Alexander Bunge, and Bernhard Spinnler. Advantage of reach-dependent capacity in optical networks enabled by OFDM. In *Proceedings of the International OFDM-Workshop*, Hamburg, Germany, September 2009.

[SKS07] Matthias Scheffel, Moritz Kiese, and Thomas Stidsen. A clustering approach for scalable network design. In *Proceedings of the International Network Optimization Conference (INOC)*, Spa, Belgium, April 2007. 93

[SrPR+07] Thomas Stidsen, Bjørn Petersen, Kasper Bonne Rasmussen, Simon Spoorendonk, Martin Zachariasen, Franz Rambach, and Moritz Kiese. Optimal routing with single backup path protection. In *Proceedings of the International Network Optimization Conference (INOC)*, Spa, Belgium, April 2007. 79

General Literature

[ABB+04] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. T. Morris, "Link-level measurements from an 802.11b mesh network," in *ACM SIGCOMM*, Portland, OR, USA, Aug. 2004. 108, 109, 110

[ACD+00] K. Ambs, S. Cwilich, M. Deing, D. J. Houck, D. F. Lynch, and D. Yan, "Optimizing restoration capacity in the AT&T network," *INTERFACES*, vol. 30, no. 1, pp. 26–44, Jan. 2000. 1

[Ach07] T. Achterberg, "Constraint Integer Programming," Ph.D. dissertation, Technische Universität Berlin, 2007. 67, 84, 88, 95, 128

[AMOT90] R. K. Ahuja, K. Mehlhorn, J. Orlin, and R. E. Tarjan, "Faster algorithms for the shortest path problem," *Journal of the ACM*, vol. 37, no. 2, pp. 213–223, Apr. 1990. 34

[App69] L. H. Appelgren, "A column generation algorithm for a ship scheduling problem," *Transportation Science*, vol. 3, no. 1, pp. 53–69, feb 1969. 33

[ATJ92] R. Anbil, R. Tanga, and E. L. Johnson, "A global approach to crew-pairing optimization," *IBM Systems Journal*, vol. 31, no. 1, pp. 71–78, 1992. 81

[BDG04] Y. Breitbart, F. Dragan, and H. Gobjuka, "Effective network monitoring," in *Proceedings of the International Conference on Computer Communications and Networks (ICCCN)*, Oct. 2004, pp. 394–399. 98

[Bel58] R. E. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, no. 1, pp. 87–90, 1958. 87

[BGJ08] R. Borndörfer, M. Grötschel, and U. Jaeger, *Produktionsfaktor Mathematik*. Berlin and Munich, Germany: Springer, 2008, ch. Planungsprobleme im öffentlichen Verkehr, pp. 127–156. 1

[BH03] C. Bettstetter and C. Hartmann, "Connectivity of wireless multihop networks in a shadow fading environment," in *Proceedings of the ACM international workshop on Modeling analysis and simulation of wireless and mobile systems (MSWIM)*, San Diego, CA, USA, 2003, pp. 28–32. 112

[BH05] C. Bettstetter and C. Hartmann, "Connectivity of wireless multihop networks in a shadow fading environment," *Wireless Networks*, vol. 11, no. 5, pp. 571–579, Sep. 2005. 109, 112, 131

[Bha94] R. Bhandari, "Optimal diverse routing in telecommunication fiber networks," in *Proceedings of IEEE Infocom*, Jun. 1994, pp. 1498–1508. 34, 88

[Bha99] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*. New Jersey, NY, USA: Kluwer Academic Publishers, 1999. 34, 52

[BHM05] C. Bettstetter, C. Hartmann, and C. Moser, "How does randomized beamforming improve the connectivity of ad hoc networks?" in *IEEE International Conference on Communications*, vol. 5, Seoul, Korea, May 2005, pp. 3380–3385. 112

[BHV00] C. Barnhart, C. A. Hane, and P. H. Vance, "Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems," *Operations Research*, vol. 48, no. 2, pp. 318–326, Apr. 2000. 86, 87

[Bix02] R. E. Bixby, "Solving real-world linear programs: A decade and more of progress," *Operations Research*, vol. 50, no. 1, pp. 3–15, 2002. 1

[BJN+98] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, "Branch-and-price: Column generation for solving huge integer programs," *Operations Research*, vol. 46, no. 3, pp. 316–329, 1998. 86

[Bla77] R. G. Bland, "New finite pivoting rules for the simplex method," *Mathematics of Operations Research*, vol. 2, no. 2, pp. 103–107, 1977. 25

[BLF92] O. Berman, R. C. Larson, and N. Fouska, "Optimal location of discretionary service facilities," *Transportation Science*, vol. 26, no. 3, pp. 201–211, Aug. 1992. 100

[BLRS07] D. M. Blough, M. Leoncini, G. Resta, and P. Santi, "Topology control with better radio models: Implications for energy and multi-hop interference," *Performance Evaluation*, vol. 64, no. 5, pp. 379–398, Jun. 2007. 112

[Bor77] K. H. Borgwardt, "Untersuchung zur Asymptotik der Mittleren Schrittzahl von Simplexverfahren in der Linearen Optimierung," Ph.D. dissertation, Universität Kaiserslautern, Kaiserslautern, BRD, 1977. 27

[Bor87] K. H. Borgwardt, *The simplex method*, ser. Algorithms and Combinatorics. Berlin Heidelberg: Springer, 1987. 27

[Bra09] K. F. Braun, "Electrical oscillations and wireless telegraphy," in *Physics 1901-1921*, ser. Nobel Lectures. Amsterdam, Netherlands: Elsevier Publishing Company, Dec. 1909. 115

[BT97] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, 1st ed., ser. Athena Scientific Optimization and Computation Series. Cambridge, MA, USA: Athena Scientific, Jan. 1997. 22, 24, 26, 27, 30, 31

[Čer85] V. Černý, "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm," vol. 45, pp. 41–51, Jan. 1985. 53

[CG02] M. Clouqueur and W. D. Grover, "Mesh-restorable networks with complete dual failure restorability and with selectively enhanced dual-failure restorability properties," in *Proceedings of the Optical Networking and Communications Conference (Opticomm)*, Boston, MA, USA, Aug. 2002. 93

[Cha66] R. W. Chang, "Synthesis of bandlimited orthogonal signals for multichannel data transmission," *Bell System Technical Journal*, vol. 45, pp. 1775–1796, Dec. 1966. 61

[Chv83] V. Chvátal, *Linear Programming*. W. H. Freeman and Company, Jan. 1983. 24, 25

[Cis09] "Cisco visual networking index: Forecast and methodology, 2008—2013," Cisco Systems, Inc., Whitepaper, Jun. 2009. 92

[CLR90] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA, USA: the MIT Press, McGraw-Hill Book Company, 1990. 52, 87

[CR89] Y.-C. Cheng and T. G. Robertazzi, "Critical connectivity phenomena in multihop radio models," *IEEE Transactions on Communications*, vol. 37, no. 7, pp. 770–777, Jul. 1989. 112

[Cra92] D. Crawford, "Fiber optic cable dig-ups – causes and curses," in *Network Reliability: A Report to the Nation*. Alliace for Telecommunications Industry (ATIS), 1992. 72

[CS07] M. W. Chbat and H.-J. Schmidtke, "Falling boundaries from metro to ULH optical transport equipment," in *Proceedings of the Optical Fiber Communication and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, San Diego, CA, USA, Mar. 2007, pp. 1–6. 7

[Dak65] R. J. Dakin, "A tree-search algorithm for mixed integer programming problems," *The Computer Journal*, vol. 8, no. 3, pp. 250–255, 1965. 31

[Dan63] G. B. Dantzig, *Linear Programming and Extensions*. Princeton, NJ, USA: Princeton University Press, 1963. 23, 24, 25, 26

[Dav00] B. S. Davie, *MPLS: Technology and Applications*, 1st ed. 525 B Street, Suite 1900, San Diege, CA 92101-4495 USA: Academic Press, 2000. 56

[DG06] E. A. Doumith and M. Gagnaire, "Traffic grooming in multi-layer WDM networks: Meta-heuristics versus sequential algorithms," in *Proceedings of the Open European Summer School*, Stuttgart, Germany, Sep. 2006, pp. 129–136. 45

[DGG+99] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merive, "A flexible model for resource management in virtual private networks," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. Cambridge, MA, USA: ACM, 1999, pp. 95–108. 19

[DHM57] M. Doelz, E. Heald, and D. Martin, "Binary data transmission techniques for linear systems," *Proceedings of the IRE*, vol. 45, no. 5, pp. 656–661, May 1957. 61

[Dij59] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959. 34, 48, 67, 84

[DLT05] N. Duffield, C. Lund, and M. Thorup, "Learn more, sample less: control of volume and variance in network measurement," *IEEE Transactions on Information Theory*, vol. 51, no. 5, pp. 1756–1775, May 2005. 98

[DPI01] B. J. Dixon, R. D. Pollard, and S. Iezekiel, "Orthogonal frequency-division multiplexing in wireless communication systems with multimode fiber feeds," *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, no. 8, pp. 1404–1409, Aug. 2001. 61

[DS98] G. Dahl and M. Stoer, "A cutting plane algorithm for multicommodity survivable network design problems," *INFORMS Journal on Computing*, vol. 10, no. 1, pp. 1–11, Dec. 1998. 74

[DW60] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, no. 1, pp. 101–111, Jan. 1960. 28

[DW00] A. Dwivedi and R. E. Wagner, "Traffic model for USA long-distance optical network," in *Proceedings of the Optical Fiber Communication and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, vol. 1, 2000, pp. 156–158. 19

[DZJ08] S. Durrani, X. Zhou, and H. M. Jones, "Connectivity of wireless ad hoc networks with random beamforming: An analytical approach," in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Cannes, France, Sep. 2008, pp. 1–5. 113

[DZP+08] M. Dzida, M. Zagożdżon, M. Pioró, T. Śliwiński, and W. Ogryczak, "Path generation for a class of survivable network design problems," in *Proceedings of the Conference on Next Generation Internet Networks (NGI)*, Cracow, Poland, Apr. 2008, pp. 31–38. 79

[EAB09] T. Engel, A. Autenrieth, and J.-C. Bischoff, "Packet layer topologies of cost optimized transport networks multi-layer netwok optimization," in *Proceedings of the International Conference on Optical Network Design an Modeling (ONDM)*, Feb. 2009, pp. 1–7. 20

[ER59] P. Erdős and A. Rényi, "On random graphs," *Publicationes Mathematicae Debrecen*, vol. 6, no. 290-297, p. 156, 1959. 33

[Far02] J. Farkas, "Über die Theorie der einfachen Ungleichungen," *Journal für die Reine und Angewandte Mathematik*, no. 124, pp. 1–27, 1902. 24

[FDvdB+08] C. Fludger, T. Duthel, D. van den Borne, C. Schulien, E.-D. Schmidt, T. Wuth, J. Geyer, E. De Man, K. Giok-Djan, and H. de Waardt, "Coherent equalization and POLMUX-RZ-DQPSK for robust 100-GE transmission," *Journal of Lightwave Technology*, vol. 26, no. 1, pp. 64–72, Jan. 2008. 61

[FF58] L. R. Ford, Jr. and D. R. Fulkerson, "A suggested computation for maximal multi-commodity network flows," *Management Science*, vol. 5, no. 1, pp. 97–101, Oct. 1958. 28

[FGK73] L. Fratta, M. Gerla, and L. Kleinrock, "The flow deviation method: An approach to store-and-forward communicaiton network design," *Networks*, vol. 3, no. 2, pp. 97–133, May 1973. 111

[Fis07] T. Fischer, "Online-routing in selektiv regenerierenden DWDM-Kernnetzen unter Berücksichtigung der Übertragungsgüte," Ph.D. dissertation, Technische Universität München, Munich, Germany, Dec. 2007. 42

[Fou27] J. B. J. Fourier, "Analyse des travaux de l'acadamie royale des sciences, pendant l'annee 1824, partie mathematique," *Histoire de l'Academie Royale des Sciences de l'Institut de France*, pp. xlvii – lv, 1827. 21

[FR89] T. A. Feo and M. G. Resende, "A probabilistic heuristic for a computationally difficult set covering problem," *Operations Research Letters*, vol. 8, no. 2, pp. 67–71, Apr. 1989. 45, 53

[Fri46] H. T. Friis, "A Note on a Simple Transmission Formula," *Proceedings of the IRE*, vol. 34, no. 5, pp. 254–256, May 1946. 109, 114

[FT00] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, vol. 2, 2000, pp. 519–528. 55

[GDW00] N. Ghani, S. Dixit, and T.-S. Wang, "On IP-over-WDM integration," *IEEE Communications Magazine*, vol. 38, no. 3, pp. 72–84, Mar. 2000. 12, 56

[GG63a] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting-stock problem," *Operations Research*, vol. 9, no. 6, pp. 849–859, Nov. 1963. 28

[GG63b] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting stock problem part ii," *Operations Research*, vol. 11, no. 6, pp. 863–888, Nov. 1963. 28

[GJ79] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman and Company, 1979. 79

[GKT51] D. Gale, H. W. Kuhn, and A. W. Tucker, *Activity Analysis of Production and Allocation*. New York, NY, USA: Wiley, 1951, ch. Linear Programming and the Theory of Games, pp. 317–329. 24

[GLS81] M. Grötschel, L. Lovász, and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, vol. 1, no. 2, pp. 169–197, Jun. 1981. 28, 106

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization (Algorithms and Combinatorics)*. Berlin, Germany: Springer, 1988. 28

[Gom58] R. E. Gomory, "Outline of an algorithm for integer solutions to linear programs," *Bulletin of the American Mathematical Society*, vol. 64, no. 5, 1958. 29, 30

[GPJR$^+$05] O. Gonzalez, R. Perez-Jimenez, S. Rodriguez, J. Rabadan, and A. Ayala, "OFDM over indoor wireless optical channel," *IEE Proceedings Optoelectronics*, vol. 152, no. 4, pp. 199–204, Aug. 2005. 61

[Gro04] W. D. Grover, *Mesh-Based Survivable Networks*. Upper Saddle River, New Jersey, NY, USA: Prentice Hall PTR, 2004. 72, 73, 74, 75, 77

[Gru07] C. G. Gruber, "Design and optimization of resilient multipath networks," Ph.D. dissertation, TU München, Munich, Germany, Jun. 2007. 16, 17, 36, 72, 73, 74, 75, 76

[GW98] N. Geng and W. Wiesbeck, *Planungsmethoden für die Mobilkommunikation*. Berlin and Heidelberg, Germany: Springer, 1998. 110

[Hak64] S. L. Hakimi, "Optimum locations of switching centers and the absolute centers and medians of a graph," *Operations Research*, vol. 12, no. 3, pp. 450–459, May 1964. 100

[Hak65] S. L. Hakimi, "Optimal distribution of switching centers in a communication network and some related graph theoretic problems," *Operations Research*, vol. 13, no. 3, pp. 462–475, May 1965. 100

[HE01] C. Hartmann and J. Eberspächer, "Adaptive radio resource management in F/TDMA cellular networks using smart antennas," *European Transactions on Telecommunications*, vol. 12, no. 5, pp. 439–452, Oct. 2001. 116

[HGMS08] R. Hülsermann, M. Gunkel, C. Meusburger, and D. A. Schupke, "Cost modeling and evaluation of capital expenditures in optical multilayer networks," *Journal of Optical Networking*, vol. 7, no. 9, pp. 814–833, Aug. 2008. 44, 63, 93, 95, 97

[HJJ03] D. Henderson, S. H. Jacobson, and A. W. Johnson, *Handbook of Metaheuristics*. Dordrecht, Netherlands: Kluwer Academic Publishers, 2003, ch. The Theory and Practice of Simulated Annealing, pp. 287–319. 53

[HL96] K. Hafner and M. Lyon, *Where Wizards Stay Up Late – the Origins of the Internet*. New York, NY, USA: Touchstone, 1996. 5

[HMS03] J. Hershberger, M. Maxel, and S. Suri, "Finding the k shortest simple paths: A new algorithm and its implementation," in *Workshop on Algorithm Engineering and Experiments*, Baltimore, MD, USA, Januar 2003. 86

[HV06] H. Höller and S. Voß, "A heuristic approach for combined equipment-planning and routing in multi-layer SDH/WDM networks," *European Journal of Operational Research*, vol. 171, no. 3, pp. 787–796, Jun. 2006. 45

[ICBD04] G. Iannaccone, C.-N. Chuah, S. Bhattacharyya, and C. Diot, "Feasibility of IP restoration in a tier 1 backbone," *IEEE Network*, vol. 18, no. 2, pp. 13–19, Mar. 2004. 75, 76

[JMST09] S. Jansen, I. Morita, T. Schenk, and H. Tanaka, "121.9-gb/s PDM-OFDM transmission with 2-b/s/Hz spectral efficiency over 1000 km of SSMF," *Journal of Lightwave Technology*, vol. 27, no. 3, pp. 177–188, Feb. 2009. 61

[Joh77] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM*, vol. 24, no. 1, pp. 1–13, Jan. 1977. 34

[Joh05] E. L. Johnson, "IFORS' Operational Research Hall of Fame Ralph E. Gomory," *International Transactions in Operational Research*, no. 12, pp. 539–543, 2005. 29

[Kan60] L. V. Kantorovich, "Mathematical Methods of Organizing and Planning Production," *Management Science*, vol. 6, no. 4, pp. 366–422, Jul. 1960. 21

[KBB+05] T. Kaiser, A. Bourdoux, H. Boche, Javier, R. Fonollosa, J. B. Andersen, and W. Utschick, Eds., *Smart Antennas – State of the Art*, ser. EURASIP Book Series on Signal Processing and Communications. New York, NY, USA: Hindawi Publishing Corporation, 2005. 116

[KCDGV83] S. Kirkpatrick, J. C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983. 53

[KCT95] H.-J. Kim, S.-H. Chung, and D.-W. Tcha, "Optimal design of the two-level distributed network with dual homing local connections," *IIE Transactions*, vol. 27, no. 5, pp. 555–563, Oct. 1995. 92

[Kha79] L. G. Khachian, "A polynomial algorithm in linear programming," *Soviet Mathematics Doklady*, vol. 20, pp. 191–194, 1979. 27

[KHT95] R. Kawamura, H. Hadama, and I. Tokizawa, "Implementation of self-healing function in ATM networks based on virtual path concept," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 1, Apr 1995, pp. 303–311. 74

[KM72] V. L. Klee and G. J. Minty, "How good is the simplex algorithm?" in *Inequalities III*, O. Shisha, Ed., Orlando, FL, USA, 1972, pp. 159–175. 26

[KOR+06] A. M. Koster, S. Orlowski, C. Raack, G. Baier, and T. Engel, "Single-layer cuts for multi-layer network design problems," Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, Berlin-Dahlem, Germany, Tech. Rep. 07-21, Aug. 2006. 32

[Kos06] H. Koskinen, "Analytical study of connectivity in wireless multihop networks utilizing beamforming," in *Proceedings of the ACM international symposium on Modeling analysis and simulation of wireless and mobile systems (MSWiM)*, Terromolinos, Spain, 2006, pp. 212–218. 113

[KPP06] V. Kaibel, M. Peinhardt, and M. E. Pfetsch, "Orbitopal fixing," Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, Berlin-Dahlem, Germany, Tech. Rep. 06-48, 2006. 32

[Kra02] O. Krauss, *DWDM and Optical Networks*, B. Siemens Aktiengesellschaft and Munich, Eds. Erlangen, Germany: Publicis Corporate Publishing, Jan. 2002. 59

[KRJ00] B. Krongold, K. Ramchandran, and D. Jones, "Computationally efficient optimal power allocation algorithms for multicarrier communication systems," *IEEE Transactions on Communications*, vol. 48, no. 1, pp. 23–27, Jan. 2000. 61

[KRR91] N. Karmarkar, M. G. Resende, and K. Ramakrishnan, "An interior point algorithm to solve computationally difficult set covering problems," *Mathematical Programming*, vol. 52, pp. 597–618, 1991. 27

[KST94] R. Kawamura, K. Sato, and I. Tokizawa, "Self-healing ATM networks based on virtual path concept," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 1, pp. 120–127, jan 1994. 74

[KT95] R. Kawamura and I. Tokizawa, "Self-healing virtual path architecture in ATM networks," *IEEE Communications Magazine*, vol. 33, no. 9, pp. 72–79, Sep. 1995. 74

[LD60] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, 1960. 30

[LDA07] A. J. Lowery, L. B. Du, and J. Armstrong, "Performance of optical OFDM in ultralong-haul WDM lightwave systems," *Journal of Lightwave Technology*, vol. 25, no. 1, pp. 131–138, Jan. 2007. 61

[Lev65] Y. A. Levin, "On an algorithm for the minimization of convex functions," *Soviet Mathematics Doklady*, vol. 6, pp. 286–290, 1965. 27

[LKS91] J. K. Lenstra, A. H. R. Kan, and A. Schrijver, Eds., *History of Mathematical Programming*. Amsterdam, Netherlands: Elsevier Science Publishers B. V., 1991. 21

[LL96] J. Litva and T. K.-Y. Lo, *Digital Beamforming in Wireless Communications*. Norwood, MA, USA: Artech House Publishers, 1996. 116

[LMSK63] J. D. Little, K. G. Murty, D. W. Sweeney, and C. Kerel, "An algorithm for the travelling salesman problem," *Operations Research*, vol. 11, no. 6, pp. 972–989, 1963. 30

[LNT02] H. Lundgren, E. Nordström, and C. Tschudin, "Coping with communication gray zones in IEEE 802.11b based ad hoc networks," in *Proceedings of the ACM International Workshop on Wireless Mobile Multimedia (WOWMOM)*. Atlanta, Georgia, USA: ACM, Sep. 2002, pp. 49–55. 108

[LS99] J. T. Linderoth and M. W. Savelsbergh, "A computational study of search strategies for mixed integer programming," *INFORMS Journal on Computing*, no. 11, 1999. 31

[LTWW94] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking (TON)*, vol. 2, no. 1, pp. 1–15, Feb. 1994. 10

[LYTL04] S. Lee, M. Yuang, P.-L. Tien, and S.-H. Lin, "A lagrangean relaxation-based approach for routing and wavelength assignment in multigranularity optical WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 9, pp. 1741–1751, Nov. 2004. 45

[Mar03] F. Margot, "Exploiting orbits in symmetric ILP," *Mathematical Programming*, vol. 98, no. 1-3, pp. 3–21, 2003. 32

[MBRM96] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEEE/ACM Transactions on Networking*, vol. 4, no. 5, pp. 684–696, Oct. 1996. 45

[Mey03] U. Meyer, "Average-case complexity of single-source shortest-paths algorithms: lower and upper bounds," *Journal of Algorithms*, vol. 48, no. 1, pp. 91–134, Aug. 2003. 34

[MFH08] H. Ma, D. Fayek, and P.-H. Ho, "Availability-constrained multipath protection in backbone networks with double-link failure," in *IEEE International Conference on Communications (ICC)*, Beijing, China, May 2008, pp. 158–164. 93

[Mit96] J. E. Mitchell, *Interior Point Methods for Combinatorial Optimization*. Kluwer Academic Publishers, 1996, ch. Interior Point Methods of Mathematical Programming. 27

[MN98] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, Jan. 1998. 128

[MS69] K. T. Marshall and J. W. Suurballe, "A note on cycling in the simplex method," *Naval Research Logistics Quarterly*, vol. 16, pp. 121–137, 1969. 26

[MT00] C. M. Machuca and P. Thiran, "An efficient algorithm for locating soft and hard failures in WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1900–1911, Oct. 2000. 103

[NP91] G. L. Nemhauser and S. Park, "A polyhedral approach to edge coloring," *Operations Research Letters*, vol. 10, no. 6, pp. 315–322, Aug. 1991. 33

[nsr08] "2006-2007 biennial report," Network Reliability Steering Committee, Tech. Rep., Aug. 2008. 72

[NW88] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York, NY, USA: John Wiley & Sons, 1988. 31, 32

[Ogg01] C. Oggerino, *High Availability Network Fundamentals*. Indianapolis, IN, USA: Cisco Press, may 2001. 72

[OP08] S. Orlowski and M. Pióro, "On the complexity of column generation in survivable network design with path-based survivability mechanisms," Zuse Institut Berlin and Warsaw University of Technology, Berlin and Warsaw, Tech. Rep., Dec. 2008. 29, 84, 106

[OPTW07] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," in *Proceedings of the International Network Optimization Conference (INOC)*, Spa, Belgium, Apr. 2007. 18, 63, 96

[Orl09] S. Orlowski, "Optimal design of multi-layer telecommunication networks," Ph.D. dissertation, Technische Universität Berlin, Berlin, Germany, May 2009. 19, 58, 68, 69, 74, 88

[PAS05] R. G. Prinz, A. Autenrieth, and D. A. Schupke, "Dual failure protection in multilayer networks based on overlay or augmented model," in *Proceedings of the International Workshop on Design of Reliable Communication Networks (DRCN)*, Ischia, Italy, Oct. 2005, pp. 179–186. 93

[PCCF09] P. Poggiolini, A. Carena, V. Curri, and F. Forghieri, "Evaluation of the computational effort for chromatic dispersion compensation in coherent optical PM-OFDM and PM-QAM systems," *Opt. Express*, vol. 17, no. 3, pp. 1385–1403, Jan. 2009. 61

[Pen99] M. D. Penrose, "On k-connectivity for a geometric random graph," *Random Structures and Algorithms*, vol. 15, no. 2, pp. 145–164, Sep. 1999. 113

[Pet04] S. Pettie, "A new approach to all-pairs shorts paths on real-weighted graphs," *Theoretical Computer Science*, vol. 312, no. 1, pp. 47–74, Jan. 2004. 34

[Pet08] S. Pettie, *Encyclopedia of Algorithms*. Springer, 2008, ch. All Pairs Shortest Paths in Sparse Graphs, pp. 28–30. 88

[PM04] M. Pióro and D. Medhi, *Routing, Flow and Capacity Design in Communication and Computer Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2004. 36, 42

[PND88] H. Pirkul, S. Narasimhan, and P. De, "Locating concentrators for primary and secondary coverage in a computer communications network," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 450–458, Apr. 1988. 93

[PR91] M. Padberg and G. Rinaldi, "A branch-and-cut algorithm for the resolution of large-scale symmetric travelling salesman problems," *SIAM Review*, vol. 33, no. 1, pp. 60–100, Mar. 1991. 32

[PS02] J. G. Proakis and M. Salehi, *Communication Systems Engineering*, 2nd ed. Upper Saddle River, New Jersey, NY, USA: Prentice Hall, Inc., 2002. 60, 61, 109, 114

[Ram01] R. Ramanathan, "On the performance of ad hoc networks with beamforming antennas," in *Proceedings of the ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)*, Long Beach, CA, USA, Oct. 2001, pp. 95–105. 116

[Rap96] T. S. Rappaport, *Wireless Communications*. Piscataway, NJ, USA: Prentice-Hall, Inc., 1996. 109

[RC08] S. Ramasubramanian and A. Chandak, "Dual-link failure resiliency through backup link mutual exclusion," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 1, pp. 157–169, Feb. 2008. 93

[RF81] D. M. Ryan and B. A. Foster, "An integer programming approach to scheduling," in *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling*, A. Wren, Ed. North-Holland Publishing Company, 1981, pp. 269–280. 86

[Rie03] A. Riedl, "Routing optimization and capacity assignment in multi-service IP networks," Ph.D. dissertation, Technische Universität München, Munich, Germany, Dec. 2003. 19

[RR03] M. G. Resende and C. C. Ribeiro, *Handbook of Metaheuristics*. Dordrecht, Netherlands: Kluwer Academic Publishers, 2003, ch. Greedy Randomized Adaptive Search Procedures, pp. 219–249. 53

[RRS07] A. S. Reaz, V. Ramamurthi, and S. Sarkar, "Capacity and delay aware routing in hybrid wireless-optical broadband access network," in *Proceedings of the International Symposium on Advanced Networks and Telecommunication Systems (ANTS)*, Mumbai, India, Dec. 2007, pp. 1–2. 111

[RRS+08] A. S. Reaz, V. Ramamurthi, S. Sarkar, D. Ghosal, S. Dixit, and B. Mukherjee, "CaDAR: An efficient routing algorithm for wireless-optical broadband access network," in *Proceedings of the IEEE Conference on Communications (ICC)*, Beijing, China, May 2008, pp. 5191–5195. 111

[RS96] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 840–851, Jun. 1996. 45

[SAF01] D. A. Schupke, A. Autenrieth, and T. Fischer, "Survivability of multiple fiber duct failures," in *Proceedings of the International Workshop on the Design of Reliable Communication Networks (DRCN)*, Budapest, Hungary, Oct. 2001, pp. 7–10. 93

[Sal67] B. Saltzberg, "Performance of an efficient parallel data transmission system," *IEEE Transactions on Communication Technology*, vol. 15, no. 6, pp. 805–811, Dec. 1967. 61

[SBV01] P. Santi, D. M. Blough, and F. Vainstein, "A probabilistic analysis for the range assignment problem in ad hoc networks," in *Proceedings of the ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)*. Long Beach, CA, USA: ACM, 2001, pp. 212–220. 112

[Sch05] M. C. Scheffel, "Regenerator allocation strategies for optical transparency domains considering transmission limitations," in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 3, Seoul, Korea, May 2005, pp. 1771–1776. 43, 100

[SDM07] S. Sarkar, S. Dixit, and B. Mukherjee, "Hybrid wireless-optical broadband access network (WOBAN): A review of relevant challenges," *Journal of Lightwave Technology*, vol. 25, no. 11, pp. 3329–3340, Nov. 2007. 14

[Sha49] C. E. Shannon, "Communication in the presence of noise," *Proceedings Institute of Radio Engineers*, vol. 37, pp. 10–21, 1949. 62

[SHK08] B. Spinnler, F. N. Hauske, and M. Kuschnerov, "Adaptive equalizer complexity in coherent optical receivers," in *Proceedings of the European Conference on Optical Communication (ECOC)*, Sep. 2008, pp. 127–128. 61

[Sho72] N. Z. Shor, "Utilization of the operation of space dilation in the minimization of of convex functions," *Cybernetics and System Analysis*, vol. 6, no. 1, pp. 7–15, Dec. 1972, translated from *Kibernetika*, No 1, pp.6-12, January-February, 1970. 27

[SK06] T. Stidsen and P. Kjærulff, "Complete rerouting protection," *Journal of Optical Networking*, vol. 5, no. 6, pp. 481–492, May 2006. 75

[Sma83] S. Smale, "On the average number of steps of the simplex method of linear programming," *Mathematical Programming*, vol. 27, pp. 241–262, 1983. 27

[SP03] D. A. Schupke and R. G. Prinz, "Performance of path protection and rerouting for WDM networks subject to dual failures," in *Proceedings of the Optical Fiber Communication and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, vol. 1, San Diego, CA, USA, Mar. 2003, pp. 209–210. 93

[SPG+06] M. C. 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 *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, San Francisco, CA, USA, Nov. 2006, pp. 1–6. 42, 44, 45, 52, 54

[Spo08] S. Spoorendonk, "Cut and column generation," Ph.D. dissertation, DIKU Department of Computer Science, University of Copenhagen, Copenhagen, Denmark, Oct. 2008. 79

[SR08] G. Sasaki and C. Rozic, "Cost efficient survivable IP over WDM with dual homing," in *Proceedings of the Optical Fiber Communication and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, San Diego, CA, USA, Feb. 2008, pp. 1–3. 92

[SRC84] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems (TOCS)*, vol. 2, no. 4, pp. 277–288, Nov. 1984. 75

[SRJ+91] S. Seidel, T. Rappaport, S. Jain, M. Lord, and R. Singh, "Path loss, scattering and multipath delay statistics in four european cities for digital cellular and microcellular radiotelephone," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 4, pp. 721–730, Nov. 1991. 110

[SS02] G. H. Sasaki and C.-F. Su, "The cost of survivable IP over WDM," in *Proceedings of the OptoElectronics and Communications Conference (OECC)*, Kanagawa, Japan, Jul. 2002, pp. 396–397. 92

[SS03] G. H. Sasaki and C.-F. Su, "The interface between IP and WDM and its effect on the cost of survivability," *IEEE Communications Magazine*, vol. 41, no. 1, pp. 74–79, Jan. 2003. 92

[SSC+02] S. Stanic, S. Subramaniam, H. Choi, G. Sahin, and H.-A. Choi, "On monitoring transparent optical networks," in *Proceedings of the International Conference on Parallel Processing Workshops*, Vancouver, B.C., Canada, Aug. 2002, pp. 217–223. 102, 103

[SSG03] D. A. Schupke, M. C. Scheffel, and W. D. Grover, "An efficient strategy for wavelength conversion in WDM p-cycle networks," in *Proceedings of the International Workshop on Design of Reliable Communication Networks (DRCN)*, Ischia, Italy, Oct. 2003, pp. 221–227. 43

[SST07] M. Spyropoulou, S. Sygletos, and I. Tomkos, "Investigation of multiwavelength regeneration employing quantum-dot semiconductor optical amplifiers beyond 40Gb/s," in *Proceedings of the International Conference on Transparent Optical Networks (ICTON)*, vol. 1, Rome, Italy, Jul. 2007, pp. 102–105. 8

[ST84] J. W. Suurballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, no. 2, 1984. 34

[ST04] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *Journal of the ACM*, vol. 51, no. 3, pp. 385–463, May 2004. 27

[Ste96] S. G. Steinberg, "Netheads vs bellheads," *Wired Magazine*, vol. 4, no. 10, 1996. 10

[Str72] J. E. Strum, *Introduction to Linear Programming*. San Francisco, CA, USA: Holden-Day, 1972. 25

[SYDM08] S. Sarkar, H.-H. Yen, S. Dixit, and B. Mukherjee, "A novel delay-aware routing algorithm (DARA) for a hybrid wireless-optical broadband access network (WOBAN)," *IEEE Network Magazine*, vol. 22, no. 3, pp. 20–28, May 2008. 110

[SYMT07] W. Shieh, X. Yi, Y. Ma, and Y. Tang, "Theoretical and experimental study on pmd-supported transmission using polarization diversity in coherent optical ofdm systems," *Optics Express*, vol. 15, no. 16, pp. 9936 – 9947, Aug. 2007. 61

[Tan03] A. S. Tanenbaum, *Computer Networks*, 4th ed. New Jersey, NY, USA: Pearson Education, Inc., 2003. 6

[TBFC+04] G. Tosi-Beleffi, D. Forin, F. Curti, F. Matera, A. Reale, S. Betti, S. Monterosso, A. Fiorelli, and M. Guglielmucci, "3R all optical regeneration," in *Proceedings of the International Conference on Transparent Optical Networks (ICTON)*, vol. 2, Wroclaw, Poland, Jul. 2004, pp. 55–58. 8

[TMP06] M. Tornatore, G. Maier, and A. Pattavina, "Capacity versus availability trade-offs for availability-based routing," *Journal of Optical Networking*, vol. 5, no. 11, pp. 858–869, Nov. 2006. 107

[Tod02] M. J. Todd, "The many facets of linear programming," *Mathematical Programming*, vol. 91, no. 3, pp. 417–436, Feb. 2002. 27

[TPŻ09] A. Tomaszewski, M. Pióro, and M. Żotkiewicz, "On the complexity of resilient network design," *Networks*, May 2009. 29, 84

[TZ93] T. Terlaky and S. Zhang, "Pivot rules for linear programming: A survey on recent theoretical developments," *Annals of Operations Research*, no. 46, pp. 203–223, 1993. 25

[Van94] F. Vanderbeck, "Decomposition and column generation for integer programs," Ph.D. dissertation, Université Catholique de Louvain, Louvain-La-Neuve, France, Sep. 1994. 33

[VCD+05] S. Verbrugge, D. Colle, P. Demeester, R. Hülsermann, and M. Jäger, "General availability model for multilayer transport networks," in *Proceedings of the International Workshop on Design of Reliable Communication Networks (DRCN)*, Oct. 2005, pp. 85–92. 72, 103

[Vil09] R. Vilzmann, "Wireless multi-hop networks with beamforming antennas and multi-user detection," Ph.D. dissertation, TU München, Munich, Germany, Mar. 2009. 111, 113, 122, 123, 127, 128, 130

[von47] J. von Neumann, "On a maximization problem," Institute for Advanced Study, Tech. Rep., 1947. 23

[VWAH06] R. Vilzmann, J. Widmer, I. Aad, and C. Hartmann, "Low-comp-lexity beamforming techniques for wireless multihop networks," in *Proceedings of the Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, vol. 2, Reston, VA, USA, Sep. 2006, pp. 489–497. 113

[VWJ05] V. M. Vokkarane, J. Wang, and J. P. Jue, “Coordinated survivability in IP-over-optical networks with IP-layer dual-homing and optical-layer protection,” in *International Conference on Broadband Networks (BROADNETS)*, vol. 1, Boston, MA, USA, Oct. 2005, pp. 241–250. 92

[VWQ+04] V. M. Vokkarane, J. Wang, X. Qi, R. Jothi, B. Raghavachari, and J. P. Jue, “Dynamic dual-homing protection in WDM mesh networks,” in *Proceedings of the IEEE Conference on Communications (ICC)*, vol. 3, Paris, France, Jun. 2004, pp. 1644–1648. 92

[WC07] L. Wosinska and J. Chen, “Reliability performance of passive optical networks,” in *IEEE International Conference on Transparent Optical Networks (ICTON)*, vol. 3, Rome, Italy, Jul. 2007, pp. 121–124. 107

[WM97] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Transactions on Evolutionary Computing*, vol. 1, no. 1, pp. 67–82, Apr. 1997. 39

[Wol98] L. A. Wolsey, *Integer Programming*. New York, NY, USA: John Wiley & Sons, 1998. 31, 86

[WVJ+05] J. Wang, V. M. Vokkarane, R. Jothi, X. Qi, B. Raghavachari, and J. P. Jue, “Dual-homing protection in IP-over-WDM networks,” *Journal of Lightwave Technology*, vol. 23, no. 10, pp. 3111–3124, Oct. 2005. 92

[WVQJ04] J. Wang, V. M. Vokkarane, X. Qi, and J. P. Jue, “Dual-homing protection in WDM mesh networks,” in *Proceedings of the Optical Fiber Communication and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, vol. 1, San Diego, CA, USA, Feb. 2004. 92

[YCK+02] M. D. Yarvis, W. S. Conner, L. Krishnamurthy, J. Chhabra, B. Elliott, and A. Mainwaring, “Real-world experiences with an interactive ad hoc sensor network,” in *Proceedings of the International Conference on Parallel Processing - Workshops (ICCP-W)*, Vancouver, B.C., Canada, Aug. 2002, pp. 143–151. 108

[YN77] D. Yudin and A. Nemirovskii, “Informational complexity and efficient methods for the solution of convex extremal problems,” *Ekon. Mat. Metody*, vol. 13, pp. 25–45, 1977. 27

[YSM08] Q. Yang, W. Shieh, and Y. Ma, “Bit and power loading for coherent optical ofdm,” *IEEE Photonics Technology Letters*, vol. 20, no. 15, pp. 1305–1307, Aug. 2008. 61

[ZHV04] H. Zeng, C. Huang, and A. Vukovic, “Monitoring Cycles for Fault Detection in Meshed All-Optical Networks,” in *Proceedings of the International Conference on Parallel Processing - Workshops (ICCP-W)*, Montreal, Canada, Aug. 2004. 98

[ZHV05] H. Zeng, C. Huang, and A. Vukovic, "Spanning-tree based monitoring-cycle construction for fault detection and localization in mesh AONs," in *IEEE International Conference on Communications*, vol. 3, Seoul, Korea, May 2005, pp. 1726–1730. 98

[ZJDS07] X. Zhou, H. M. Jones, S. Durrani, and A. Scott, "Effects of beamforming on the connectivity of ad hoc networks," in *Proceedings of the Australian Communication Theory Workshop (AusCTW)*, Adelaide, Australia, Feb. 2007, pp. 13–18. 113

[ZJDSar] X. Zhou, H. M. Jones, S. Durrani, and A. Scott, "Connectivity analysis of wireless ad hoc networks with beamforming," *IEEE Transactions on Vehicular Technology*, 2009 (to appear). 112

[ZM02] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 122–133, Jan. 2002. 45

[Zym07] A. Zymolka, "Design of survivable optical networks by mathematical optimization," Ph.D. thesis, TU Berlin, 2007. 43

[ZZM03a] K. Zhu, H. Zhu, and B. Mukherjee, "Traffic engineering in multigranularity heterogeneous optical WDM mesh networks through dynamic traffic grooming," *IEEE Network*, vol. 17, no. 2, pp. 8–15, Mar. 2003. 45

[ZZM03b] K. Zhu, H. Zhu, and B. Mukherjee, "A review of traffic grooming in WDM networks: architectures and challenges," *Optical Networks Magazine*, vol. 4, no. 2, Mar. 2003. 45

Standards

[AMA+99] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS," RFC 2702, IETF Network Working Group, Sep. 1999. 73

[BZB+97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – version 1 functional specification," RFC 2205, IETF Network Working Group, Sep. 1997. 56, 74

[Def81] "Internet protocol," RFC 791, Defense Advanced Research Projects Agency, Sep. 1981. 5

[FAV09] A. Farrel, A. Ayyangar, and J. P. Vasseur, "Inter-domain MPLS and GMPLS traffic engineering – Resource Reservation Protocol-traffic engineering (RSVP-TE) extensions," RFC 5151, IETF Network Working Group, Feb. 2009. 10

[ITU98] *Optical Interfaces for multichannel systems with optical amplifiers*, ITU-T Recommendation G.692, Oct. 1998. 59

[ITU06a] *Ethernet Linear Protection Switching*, ITU-T Recommendation G.8031/Y.1342, Jun. 2006. 74, 75

[ITU06b] *Optical Transport Network (OTN): Linear Protection*, ITU-T Recommendation G.873.1, Mar. 2006. 73, 75

[ITU06c] *Protection switching for MPLS networks*, ITU-T Recommendation Y.1720, Dec. 2006. 73, 77

[NEH+96b] ——, "Transmission of flow labelled IPv4 on ATM data links," RFC 1954, IETF Network Working Group, May 1996. 56

[PSA05] P. Pan, G. Swallow, and A. Atlas, "Fast reroute extensions to RSVP-TE for LSP tunnels," RFC 4090, IETF Network Working Group, May 2005. 56, 76

[RTF+01] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, and A. Conta, "MPLS label stack encoding," RFC 3032, IETF Network Working Group, Jan. 2001. 56

[RVC01] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," RFC 3031, IETF Network Working Group, Jan. 2001. 56

Cited websites

[1] (2007, Oct.) The 100GET project. [Online]. Available: http://www.celtic-initiative.org/Projects/100GET/ 41

[ATT09] (2009) AT&T managed internet service. [Online]. Available: http://businessesales.att.com/products/product_mis.jhtml 92

[CEF] "Cisco IOS switching services configuration guide, release 12.1." 56

[CPL09] (2009, Jun.) Ilog cplex. [Online]. Available: http://www.ilog.com 1, 67, 84, 88, 95, 128

[EIB05] (2005) The EIBONE project. [Online]. Available: http://www.pt-it.pt-dlr.de/de/983.php 18, 58

[gsl] the GNU scientific library (GSL). [Online]. Available: http://www.gnu.org/software/gsl/ 128

[Lay09] (2009, Apr.) Layonics GbR. [Online]. Available: http://www.layonics.com 34, 48, 67, 88, 95, 128

[mer] Meraki Inc. homepage. [Online]. Available: http://www.meraki.com 14

[NOB04] (2004, Feb.) The NOBEL project. [Online]. Available: http://www.ist-nobel.org/Nobel/servlet/Nobel.Main 18, 44

[roo] The RoofNet project. Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory. [Online]. Available: http://pdos.csail.mit.edu/roofnet/doku.php 14, 108, 143

[snd09] (2009, Aug.) Sndlib. [Online]. Available: http://sndlib.zib.de 18, 19

[Sul06] B. Sullivan. (2006, Dec.) Why cell phone outage reports are secret. [Online]. Available: http://redtape.msnbc.com/2006/12/why_cell_phone_.html 72

[Ver06] (2006) Fact sheet: Internet dedicated access. [Online]. Available: http://www.verizonbusiness.com/resources/factsheets/fs_internet-dedicated-access_en_xg.pdf 92