Design of Survivable Optical Networks by Mathematical Optimization

vorgelegt von
Diplom-Mathematiker
Adrian Zymolka

Von der Fakultät II - Mathematik und Naturwissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
Dr.rer.nat.

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Ulrich Pinkall

1. Berichter: Prof. Dr. Dr. h.c. Martin Grötschel

2. Berichter: Prof. Dr. Sven O. Krumke

Tag der wissenschaftlichen Aussprache: 20. Dezember 2006

Berlin 2007

D 83

Abstract

The cost-efficient design of survivable optical telecommunication networks is the topic of this thesis. In cooperation with network operators, we have developed suitable concepts and mathematical optimization methods to solve this comprehensive planning task in practice.
Optical technology is more and more employed in modern telecommunication networks. Digital information is thereby transmitted as short light pulses through glass fibers. Moreover, the optical medium allows for simultaneous transmissions on a single fiber by use of different wavelengths. Recent optical switches enable a direct forwarding of optical channels in the network nodes without the previously required signal retransformation to electronics. Their integration creates ongoing optical connections, which are called lightpaths.
We study the problem of finding cost-efficient configurations of optical networks which meet specified communication requirements. A configuration comprises the determination of all lightpaths to establish as well as the detailed allocation of all required devices and systems. We use a flexible modeling framework for a realistic representation of the networks and their composition. For different network architectures, we formulate integer linear programs which model the design task in detail. Moreover, network survivability is an important issue due to the immense bandwidths offered by optical technology. Operators therefore request for designs which perpetuate protected connections and guarantee for a defined minimum throughput in case of malfunctions. In order to achieve an effective realization of scalable protection, we present a novel survivability concept tailored to optical networks and integrate several variants into the models.
Our solution approach is based on a suitable model decomposition into two subtasks which separates two individually hard subproblems and enables this way to compute cost-efficient designs with approved quality guarantee. The first subtask consists of routing the connections with corresponding dimensioning of capacities and constitutes a common core task in the area of network planning. Sophisticated methods for such problems have already been developed and are deployed by appropriate integration. The second subtask is characteristic for optical networks and seeks for a conflict-free assignment of available wavelengths to the lightpaths using a minimum number of involved wavelength converters. For this coloring-like task, we derive particular models and study methods to estimate the number of unavoidable conversions. As constructive approach, we develop heuristics and an exact branch-and-price algorithm. Finally, we carry out an extensive computational study on realistic data, provided by our industrial partners. As twofold purpose, we demonstrate the potential of our approach for computing good solutions with quality guarantee, and we exemplify its flexibility for application to network design and analysis.

Deutsche Zusammenfassung

In der vorliegenden Arbeit befassen wir uns mit der Planung und dem Entwurf kostengünstiger, ausfallsicherer optischer Telekommunikationsnetze. In Kooperation mit mehreren Netzbetreibern haben wir geeignete Konzepte und mathematische Optimierungsmethoden entwickelt, mit denen sich diese anspruchsvolle Planungsaufgabe praxisnah lösen lässt.
In modernen Telekommunikationsnetzen wird zunehmend optische Technologie eingesetzt. Zur Datenübertragung werden dabei digitale Informationen in Form kurzer Lichtpulse durch Glasfasern geleitet. Insbesondere erlaubt das optische Medium, auf einer Faser zeitgleich mehrere Übertragungen durch Verwendung unterschiedlicher Wellenlängen zu realisieren. Zu den modernsten Komponenten gehören optische Schalteinheiten, die eine direkte Weiterleitung optischer Übertragungen in den Netzknoten ermöglichen, ohne die Signale wie bisher in elektronische Form zurückwandeln zu müssen. Durch ihre Integration entstehen durchgängig optische Verbindungen, die als Lichtwege bezeichnet werden.
Wir untersuchen das Problem, zu gegebenen Kommunikationsanforderungen möglichst kostengünstige Konfigurationen von optischen Netzen zu bestimmen. Eine solche Konfiguration umfasst die Festlegung aller einzurichtenden Lichtwege sowie aller dazu benötigten Geräte und Systeme. Dazu bilden wir die Netzwerke und ihre Bestandteile in einer flexiblen, realitätsnahen Art ab. Für unterschiedliche technologische Auslegungen optischer Netze formulieren wir ganzzahlige lineare Programme, die das Planungsproblem detailliert modellieren. Aufgrund der enormen Übertragungsbandbreiten spielen für Netzbetreiber ferner Sicherheitsaspekte eine wichtige Rolle, um auch in Störsituationen noch einen definierten Mindestverkehr garantieren und geschützte Verbindungen aufrechterhalten zu können. Zur effektiven Realisierung von Lichtweg-Konfigurationen mit skalierbaren Ausfallsicherheitseigenschaften entwickeln wir ein speziell auf optische Netze zugeschnittenes Konzept und betten es in verschiedenen Varianten innerhalb der Modelle ein.
Unser Lösungsansatz basiert auf einer geeignet gewählten Dekomposition der Modelle, die zwei individuell schwierige mathematische Teilprobleme trennt und dadurch eine Berechnung günstiger Konfigurationen mit nachweisbarer Güte ermöglicht. Das erste Teilproblem beinhaltet das Routing der Verbindungen mit Dimensionierung von Kapazitäten, eine gemeinsame Kernaufgabe in der Netzplanung, für die bereits ausgereifte Methoden zur Verfügung stehen und eingesetzt werden. Im zweiten Teilproblem fokussieren wir auf die für optische Netze charakteristische Aufgabe, eine konfliktfreie Zuordnung vorhandener Wellenlängen zu den Lichtwegen unter möglichst wenig Verwendung aufwändiger Wellenlängenkonversionen zu finden. Wir leiten spezielle Modelle her, diskutieren Abschätzungen zur Anzahl unvermeidbarer Konversionen und entwickeln als konstruktive Verfahren neben Heuristiken eine exakte Branch-and-Price-Methode. In umfassenden Rechenstudien auf realistischen Daten, die von unseren Kooperationspartnern bereitgestellt wurden, weisen wir einerseits das Potential der entwickelten Lösungsmethodik zur Berechnung geeigneter Netzentwürfe mit garantierter Güte nach und zeigen andererseits die flexiblen Einsatzmöglichkeiten im Rahmen der Planung und Analyse optischer Netze auf.

Danksagung

Bei der Erstellung dieser Arbeit habe ich viel Unterstützung und Zuspruch erfahren dürfen, und es ist mir ein besonderes Anliegen, mich dafür bei allen von Herzen zu bedanken.

In erster Linie danke ich Herrn Grötschel für die Möglichkeit, mich mit diesem spannenden Thema auseinandersetzen zu können, Rat und Anleitung bei der Durchführung der Projekte und der Erstellung von Publikationen sowie das von ihm geschaffene Umfeld am Konrad-Zuse-Zentrum, das mir einen hervorragenden Rahmen für fruchtbare wissenschaftliche Kooperation bot.

Ebenso gebührt Arie M.C.A. Koster allergrößter Dank. Ich habe nicht nur von seiner verlässlichen, aufmerksamen und geduldigen Betreuung und unserer stets inspirierenden Zusammenarbeit profitiert, sondern fand in ihm einen Freund und eine vielfältige Begleitung, die manchem Loch zu entkommen und manche Klippe zu umschiffen half.

Weiterhin danke ich allen Mitarbeitern am Institut, die Verwaltung eingeschlossen, für die stets entgegenkommende, kooperative und kollegiale Atmosphäre. Dies gilt insbesondere für Diana Poensgen und Sebastian Orlowski, mit denen ich das Vergnügen hatte, Raum und Schokolade zu teilen. Besonders dankbar bin ich Andreas Bley, Andreas Eisenblätter, Arie Koster, Diana Poensgen, Sebastian Orlowski und Roland Wessäly für die sorgfältige Durchsicht des Manuskripts und ihre wertvollen Anregungen und Vorschläge, die wesentlich zur Gestaltung dieser Arbeit beigetragen haben. Ein ganz spezieller Dank geht auch an atesio und vor allem Roland Wessäly und Andreas Eisenblätter, die mir mit Rat und Tat zur Seite standen. Ferner möchte ich mich bei meinen Ansprechpartnern aus der Industrie für die freundliche Bereitstellung der Daten und die jederzeit angenehme und anregende Kooperation bedanken.

Meine Eltern haben nicht nur den Grundstein für alles Erreichte gelegt, sondern mir auch die notwendige Freiheit gelassen und mich dabei jederzeit mit unerschütterlichem Glauben und unermüdlichem Zuspruch begleitet und gefördert. Ihnen widme ich dafür meinen größten Dank und Respekt!

Meiner Frau Ulrike schließlich kann ich nicht mit Worten danken. Ihre Liebe und ihr Rückhalt bilden schlichtweg das, was meine Welt im Innersten zusammenhält.

Bald kamen sie an einem großen Elektroladen vorbei.[...]

Alberto zeigte auf das große Schaufenster und sagte: „Da siehst Du das 20. Jahrhundert, Sofie. Seit der Renaissance war die Welt gewissermaßen explodiert. Die Europäer begannen, um die ganze Welt zu reisen. Und heute geschieht etwas, das wir als umgekehrte Explosion bezeichnen können.“

„Wie meinst Du das?“

„Ich meine, daß die ganze Welt zu einem einzigen Kommunikationsnetz zusammengezogen wird. Vor nicht allzu langer Zeit waren die Philosophen noch viele Tage lang mit Pferd und Wagen unterwegs, um sich in der Welt zu orientieren – oder um andere Denker zu treffen. Heute können wir überall auf diesem Planeten sitzen und uns alle menschliche Erfahrung auf einen Computerbildschirm holen.“

„Das ist ein phantastischer Gedanke[...].“

„Die Frage ist, ob die Geschichte sich einem Ende nähert – oder ob wir im Gegenteil auf der Schwelle zu einer ganz neuen Zeit stehen. [...]
Die technische Entwicklung war – nicht zuletzt was die Kommunikation betrifft – in den letzten dreißig, vierzig Jahren dramatischer als in der gesamten, vorigen Geschichte zusammen. Und noch immer ist, was wir erleben, vielleicht nur der Anfang...“

(aus Jostein Gaarder, '*Sofies Welt*', 1993)

Contents

Introduction

This thesis deals with mathematical optimization methods for the design of survivable optical networks in telecommunications. The work on this topic has been initiated and motivated by a couple of industrial projects at the Zuse Institute Berlin (ZIB) with Telekom Austria AG and T-Systems International GmbH (formerly T-Systems Nova), continued in ongoing cooperation. Characteristic for research driven by real-world applications, the tasks to solve are typically tackled by a team of experts from different fields. In our case, the team consisted of the practitioners at our partners and a group of colleagues and myself at the optimization department at ZIB, headed by Martin Grötschel. Therefore, this thesis is written from our common perspective including all people I had the opportunity to cooperate with.

The first project with Telekom Austria started at the beginning of 2000 and was supposed to create a software tool for direct support of the network planning process in practice. Much attention was paid to operate on an as accurate as possible model of the optical network on disposition and its functionality in order to ensure that generated designs are in fact realizable in the field. Though not all involved details are incorporated in the following, the study has fundamentally determined our particular perception of optical networks and their working. The developed tool, in a preliminary version presented at CeBit 2001, was finally delivered to our partner in December 2001 and served further as groundwork for our continued research on the issue.
The second project with T-Systems Nova (meanwhile T-Systems International) was accompanied by a parallel project on dynamic call admission and other online tasks in traffic engineering. It was funded by the DFN-Verein, the German research network operator, and started in autumn 2000 with the goal to economically evaluate optical network designs under various settings and architectural prerequisites. To this end, we extended and enhanced our framework, models, and algorithms to integrate variable configuration alternatives. The continuously enhanced methodology has been successfully applied in several computational studies during the project and, perpetuating the fruitful cooperation up to date, it has proven a vital base for further investigations and innovations.
In consequence, we focus on two important aspects for the design of optical networks:

- economic efficiency, i.e., cost minimization accompanied by a mathematically proven quality guarantee, and
- a realistic model of optical networks and their various devices, covering all relevant properties and rules of interaction.

Optical technology is nowadays broadly employed in modern telecommunication networks. After emerging on the commercial markets, optical transmission systems were soon recognized as the future technology superseding electronic transmission in those (sub-) networks where large amounts of data are to be shipped, predominantly in the so-called backbones. Consequently, a rapid penetration of these networks by optical fibers took place, further driven by ongoing progress in extending throughput capabilities. In parallel, engineers and physicists intensified their efforts to overcome bottlenecks resulting from physical limitations in order to enable more and more functionalities being directly applied to the optical signals, like switching or signal regeneration. As a result, such new devices either are nowadays commercially available or are expected to be soon. Currently, first generation optical networks with restricted functionality in the optical domain are already state-of-the-art in practice, and though still high prices for innovative devices slow down the process, a migration to second generation optical networks will, following common opinion, take place in the near future. Our methods, explicitly taking upgrade planning into account, provide helpful support for this purpose, since they are flexible with respect to the architecture and able to cope with the design of networks of either generation.

The great advantage of optical technology, to transport immense data volumes with maximum possible speed, bears a risk, too. An unexpected disruption of working connections can yield tremendous loss of payload (as optical signals cannot be buffered) and breakdown of numerous services. Hence, network providers are in particular interested to design their high capacity optical networks in a survivable manner, taking preventive care for failure situations in order to speed up the recovery process. Therefore, we put special emphasis on the issue of survivability in optical networks. Various protection mechanisms have already been considered, and each one comes with individual properties regarding occupation of additional backup capacity, recovery time, and complexity of network management. We present a new scheme, called Demand-wise Shared Protection, which has been particularly tailored to optical networks and offers a special balance among these criteria.

Mathematical methods for network design in telecommunications have a long standing history, as such problems show a high combinatorial complexity which is hard to master by planners without appropriate support. Besides optical networks, various network architectures and technologies have been and are currently under investigation, like networks based on the Internet Protocol (IP) or by use of the Synchronous Digital Hierarchy (SDH), a widespread data transport technology. The common core task of such design problems typically consists of finding the routing of traffic connections in combination with a corresponding dimensioning of capacities required for their accommodation. For optical technology, this task extends due to a new feature: a spectrum of different wavelengths used for parallel transmissions. As a consequence, carrying out a wavelength assignment for the connections becomes part of the network planning problem. From our perspective, the complete task can be roughly subsumed as follows:

> Given is a network topology consisting of nodes and (potential) links between node pairs. For each node and link, existing and installable devices are specified. Communication forecasts define a set of demands for connections to be established between pairs of nodes together with

additional survivability requirements.
Minimizing the total cost for installations, find a feasible design of the network that meets the traffic demands. An optical network design comprises to select the required devices at each link and node such that sufficient capacities are provided for the connection establishment, and to determine a suitable routing of all needed optical connections with specified wavelength to use on each traversed fiber link such that any two connections sharing a fiber use different wavelengths.

This brief description outlines a comprehensive and complex problem, especially when focussing on a detailed network representation. A vast variety of alternative equipment configurations and routing possibilities in combination with associated wavelength assignments (or path colorings) has to be evaluated, being compatibly plugged together for construction of a feasible network design at least possible cost. We develop dedicated mathematical optimization methods providing a structured approach to determine suitable solutions. As fundament, we derive integer linear programs to model the problem formally. For increasing computational tractability, we propose a suitable decomposition approach carried out such that limited degree of accuracy is sacrificed in order to obtain good designs with approved quality guarantee. Our way of decomposing grants access to sophisticated methods for the generic core task, and special emphasis is put on the extension feature modeled as generalized coloring problem. Besides its complexity, we study primal and dual approaches, combining fast construction heuristics and lower bounds for the optimum value. For the bounds, particular strength is achieved from the devised path packing formulation solved by a column generation algorithm which is further developed to an exact branch-and-price method. The suitability of both methods and their results is evaluated in an extensive computational study.

Outline. The outline of the thesis follows a natural way when developing mathematical methods and tools for practical applications: first specifying the real-world problem, then translating it into suitable mathematical models, next finding a strategy how to approach their solving, followed by working out details and required algorithms, and finally applying the developed methodology to evaluate its performance and capabilities. This order of matters is reflected by the structuring of chapters.
In Chapter 1, we give an introduction into telecommunication networks and optical technology. We explain the practical background and discuss the layout and organization of optical networks, including the issue of network survivability. Since plenty of optional settings are possible, we make also basic assumptions defining the scope of problems which are considered.
In Chapter 2, the focused optical network design problems are formalized and described by mathematical models. A general framework provides the basis for representation of networks and their composition. For three characteristic architectures, we then discuss the modeling by integer linear programs in detail, together with selected variations and extensions. In addition, we introduce the novel survivability concept Demand-wise Shared Protection and explain its integration within the program formulations.
Chapters 3 and 4 are dedicated to the solution methodology. In Chapter 3, we explain our basic approach to solve optical network design problems by decomposition

into two subsequent subproblems and discuss its advantages. For the first of these subproblems, the dimensioning and routing subtask, we then present our solution methodology including alternative ways of application.
In Chapter 4, we investigate the second subproblem: wavelength assignment with converters, forming the characteristic subtask in optical network design. We discuss the complexity of the problem and derive integer linear program formulations. In addition to a combinatorial method, the linear program relaxations provide lower bounds on the number of unavoidable conversions. As constructive approaches, we propose several fast heuristics and provide finally an exact branch-and-price method.
In Chapter 5, we report on the results obtained with the developed methods when computing optical network designs for realistic instances. The purpose of this computational study is twofold, serving on the one hand as performance evaluation of our solution approach and, exploiting its flexibility, on the other hand as a demonstration of possible ways of utilization for design analyses and comparisons. The first part, devoted to assessment of entire configurations, is organized in several case studies on different settings in view of survivability schemes, upgrade planning, alternative hardware prerequisites, and architectures. In the second part, we appraise the various algorithms focusing on the wavelength assignment subproblem.
Some concluding remarks on the achieved results and directions for further research close the thesis. The appendix provides an overview on used notation as well as a detailed compilation of the original computational data for both input and results that have been discussed in more significant aggregation in Chapter 5.

Chapter 1

Optical networks in practice

The idea of using light as long distance information carrier arose as early as 1958, when the laser was conceived. A controlled emission of defined light signals builds the basis for photonic communication. Next milestones have been the development of guided wave transmission in the mid 1960s and the fabrication of first low-loss glass fibers around 1970, making the optical transmission practical for telecommunication networks. Although first considered as a curiosity technology, operators soon recognized the potential and the superior properties offered by optics as medium, and the light entered the networks.
The installation of the first fiber connections in the early 1970s marks the beginning of the evolution of optical networks. Due to physical enhancements, the new technology did not only compete with established systems, but soon outperformed them in effectivity. In fact, the bitrate-distance product offered by optical transmission roughly grew exponentially over the years. As a result, more and more copper cables were replaced by fibers. The 1980s brought further major breakthroughs which enlarged the application field for optics and supported to its penetration of the telecommunication infrastructures. The first fiber trans-Atlantic cable laid in 1988 provides a striking example. Consequently, fiber systems became the dominating long-distance connection technology from the late 1980s on.
Until the end of the 1990s, optics was used in telecommunication networks as pure transmission technology, whereas electronics accessed and handled the data flows in the network nodes. In the meantime, the rapidly increasing optical bandwidth capacities overstretch the capabilities of electronic equipment which is subject to fundamental limitations. As a consequence, more and more functionality "goes optic", from switching optical channels directly up to refreshing the signals, exchanging wavelengths, and more.
Nowadays, modern optical telecommunication networks form complex infrastructures composed of a plenty of various devices. Operating such a network is a challenging task. In particular, it requires an accurate planning, taking into account all technological "needs" and "offers" of the optics, in order to end up with a working network which provides the best return on the spent investments.

In this opening chapter, we give a brief introduction to optical telecommunication networks. The demanding task to accurately plan such complex infrastructures re-

quires a basic understanding of the applied technology and their mechanisms as well as general networking aspects. For the scope of this thesis, we confine the description on those facts that are relevant for the design of optical networks, taking the perspective of the network planners, not of the engineers. The presentation aggregates various aspects in our appropriate perception and is guided by experiences we learned from the practitioners at our industrial cooperation partners. Although many further issues are indispensable for the practical operation of optical networks, we leave out detailed physical and technical explanations. The books of Mukherjee [123], Ramaswami and Sivarajan [147], and Stern and Bala [154] provide a comprehensive introduction to optical networking with an engineering-oriented emphasis.

We begin with the main ingredients of optical networks, the involved hardware devices and their operational properties. Due to the rapid progress in research and product development, the attempt to give a full account of the state-of-the-art technology is futile. Moreover, network operators do not upgrade their hardware with any innovation arising, but make only major reformations from time to time. The network planning thus has to deal with combinations of new and old (if not ancient) technologies, while foreseen advancements of the next years should also be prepared within the designs to provide the future. Therefore, the description is not restricted to hardware that currently is available, but also includes those components that will be available soon, or are under development due to a demanding market. Occasionally remarked technical quantities serve just to illustrate the concerned magnitudes without claiming to represent current technological capabilities.
After having discussed the possible devices and their functionalities, we consider fundamentals of operating optical networks. The physical network and its layout build the groundwork for managing transmissions. To transport the data traffic, connections between network nodes have to be set up. We describe optical connections and discuss their operational issues. Moreover, we introduce the concept of a lightpath as a characteristic connection type for optical networks.
Since the network operation shall finally be robust against failures, survivability is a further important issue. We discuss some basic aspects and present known concepts to realize survivable networks. The most relevant schemes are categorized and compared with respect to several practical criteria.
The migration of real-world networks has brought up several constitutive network architectures, subsuming combinations of the used technology and applied concepts. To illustrate the evolution, we briefly sketch the progression of optical networks in practice and present main types of architectures that consolidated over time. For design tasks, we focus on these basic network types and their relevant properties.
The chapter is completed with a description of the planning task(s) for optical networks. A frictionless operation relies on a proper network configuration. We specify this comprehensive term and identify configuration subproblems which compile highly related issues. Moreover, planners in practice are confronted with various questions to answer. We discuss some major aspects and present typical planning types and variants. These descriptions serve as orientation for the final specification of those practice-relevant optical network design problems that are studied in this thesis and for which we develop appropriate solution methods in the subsequent chapters.

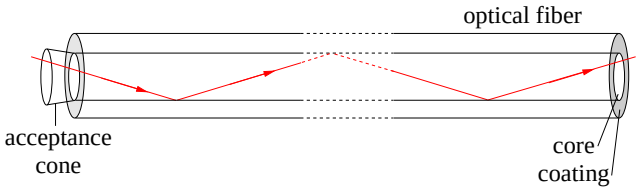


Figure 1.1: *Guided wave propagation through an optical fiber.*

1.1 Optical technology

For a long time, telecommunications was a pure electronic domain. Initially, networks of copper cables grew rapidly and brought phone services into everyone's life. As next step, data entered the networks. The extension of computer utilization and the digitalization—an ongoing trend—turned communication networks into a platform for improved services and applications, for instance in e-commerce. The continuous increase of data traffic demanded for more and more bandwidth to be provided by the networks. For this, the electronic equipment was at some point found to be constricted by physical limits. To achieve further advancement, there was need for a new medium.

1.1.1 Optical transmission

Using light as data transmission medium is based on the principle of guided wave propagation. An emitted wave propagates through a carrier material whose shape guides the wave through space to reach the intended target. For optics, the carrier material usually consists of the highly transparent glass (silica) core of an *optical fiber*. Light waves entering the fiber propagate along the glass core, being reflected whenever they hit upon the side borders of the glass.[1] Forwarded this way, the waves can only abandon the glass core at the end of the fiber (see Figure 1.1). This enables the directed transmission of light between any two points connected by optical fibers.

Optical channel. To transport data optically, the digital information delivered by electronic devices passes a *transmitter* which converts the bits into short light pulses by a laser. The light then propagates along fibers until it reaches a *receiver* where the signals are reconverted by photodiodes or photodetectors and handed back to the electronic equipment. Transmitters and receivers build the interfaces between electronics and optics. We call the optical part of such a connection on a fiber an *optical channel*, as illustrated in Figure 1.2. The shorter the emitted light pulses are, the more bits can be transmitted within some time period. The *bitrate* of a transmission system is defined as number of bits per second (b/s) that can be transmitted/received. The achievable bitrate is given by the physical skills of transmitter, receiver, and fiber. In today's practice, systems with a bitrate of

[1] Note that 'reflection' is just a simplifying description. There are different physical principles applied to direct the light waves along a fiber core, depending on the properties of the carrier material (step- vs. graded-indexed, single- vs. multi-mode etc.) as well as the specific type of light waves (mode, polarization etc.) at hand.

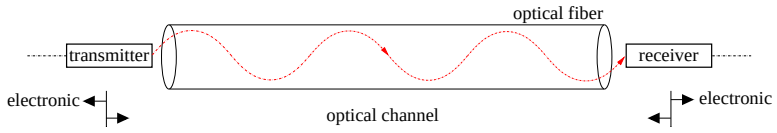


Figure 1.2: *Optical data transmission on fibers.*

10 Gigabit per second (Gb/s) are widely used. Systems with 40 Gb/s are also commercially available, although rarely applied (see Freeman [51]). News from the laboratories indicate that manufacturers already head for 160 Gb/s systems and beyond.

Signal degradation. Under ideal conditions, the received optical signals would have exactly the same shape as emitted. Unfortunately, real systems do not provide ideal conditions. Depending on the material of the glass core and its quality, there are also undesirable effects (accidental such as material impairments, but also systematical such as dispersion, attenuation, cross-talk, and others). These effects disturb the light propagation on optical fibers. Hence, the signal quality suffers a loss, limiting the distance over which an optical signal can be carried without loosing or falsifying the information. More precisely, the emitted light wave carries a rectangularly shaped amplitude coding the 0-1 bit information stream as light off/light on. On its way, it blurs to a wave with fuzzy shaped amplitude which only resembles at the original form. The receiving photodetector reinterprets the light information as bits by applying a threshold to the light amplitudes, as depicted in Figure 1.3. The longer the distance of the optical transmission is, the more the original amplitude shape is tampered. When exceeding a certain length, the light wave does not allow anymore for a correct reconversion of the original information, and bit failures occur. In practice, the *bit error rate* is used as a measure for the signal quality, defined as the ratio of incorrectly interpreted bits over all transmitted bits. The lower the bit error rate is, the less coding effort and redundant information is required for error detection and correction. Currently, fibers available at the market typically set 70 to 120 km as transmission bound without additional equipment and a bit error rate ranging between 10^{-7} and 10^{-9}.

Signal regeneration. To extend the maximum optical transmission distance on fibers, so-called *amplifiers* are used to refresh the signals optically. As the name indicates, an amplifier simply scales up the amplitude of the light wave to allow for a correct signal reconversion after an increased total transmission length. The

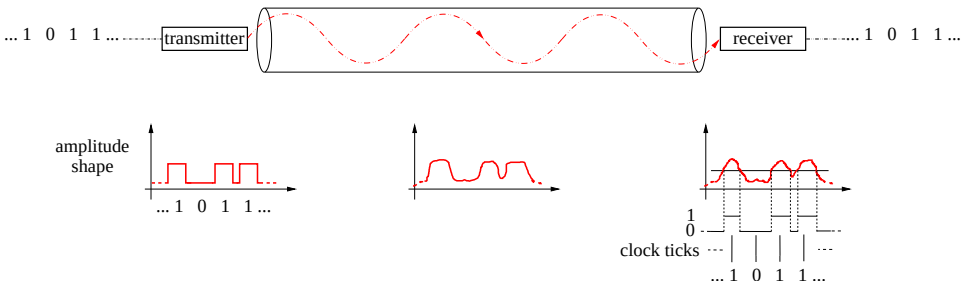


Figure 1.3: *Transformation of emitted light pulses along fibers and their restoration by a photodetector.*

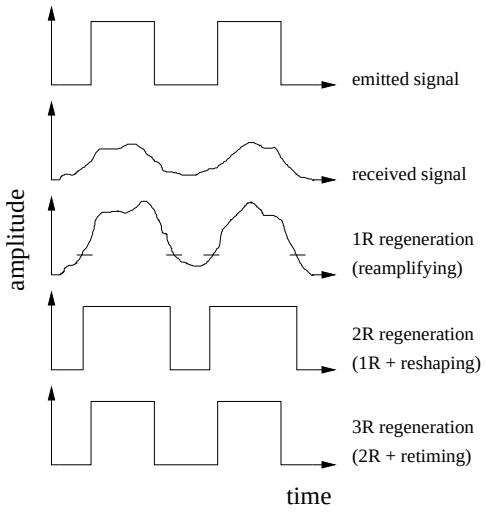


Figure 1.4: *Optical signal regeneration stages.*

maximum distance can be further elongated by placing amplifiers consecutively in regular distances. Such a *reamplification* cannot be repeated arbitrarily often, because the light wave blurring does not only influence the amplitude heights but also spreads the widths of the original signals (as wave crests and troughs). At some point, wave equalization effects dominate and make a correct signal interpretation impossible. Amplifiers are usually placed each 60 to 80 km at a fiber, enabling to transmit optical signals over up to around 1000 km, with special ultra long-haul equipment even several thousand km.
Reamplification of light waves is just the simplest kind of signal refreshing. After having passed an amplifier, the light wave shows heightened amplitudes, but still has a fuzzy waveform. A more extensive signal regeneration consists in restoring the original rectangular shape of the wave by use of a threshold similar to a receiver processing. This *reshaping* process reproduces a rectangular waveform, but the rectangle sizes may differ from the original ones due to the distorting effects. A full restoration of the original signal thus requires an additional step to unify the bit lengths: The emitting has to undergo a *retiming*. This step finally results in a rectangular waveform with steady sized rectangle lengths. It equals the original light emission as long as the received signals have been interpreted correctly, i.e., no bit errors occurred. Figure 1.4 depicts the three stages of optical signal regeneration: reamplification, reshaping, and retiming. Since each stage also involves the former stages in this order, they are also denoted as 1R, 2R, and 3R regeneration (cf. the ITU standardization recommendation in G.872 [160]).

Regenerator. The differentiation between the regeneration stages results from their distinct difficulties for optical realization. As indicated above, 1R regeneration can be optically performed by rather simple amplifiers. Using a pump laser, power is injected to increase the light wave's energy and thereby its amplitudes. Reshaping and retiming, however, are not that easy to realize. In practice, a full signal regeneration currently requires to transform the signal back to electronic form by a receiver and then pass it back to a laser reemitting light pulses. Due to the signal transformation from *optics* to *electronics* and back to *optics* again, this process is abbreviated as *o-e-o conversion*, and the executing hardware device called

transponder is built of a receiver and a transmitter back to back. Recently, progress in optical 3R regeneration has been reported, see Nolting [128]. The integration of *optical 3R regenerators* would overcome the need to interrupt optical transmissions by o-e-o conversions and lift the ban of a maximum distance. To simplify the further description, we call any device that performs full signal regeneration a *regenerator* independent of the applied technique.

Further extensions of the maximum optical transmission length have been attempted by experiments with various fiber materials and exploiting several light wave properties, such as polarization, modulation, and others. While some compositions have transcended certain disturbing effects, other distorting influences become more dominant for them. Up to date, no optical transmission system without limitation on the maximum distance to bridge is available. Research, however, yielded a variety of different kinds of devices, including fibers, transmitters, receivers, and amplifiers, which are now available at the market. Each device type comes along with specific properties that make it the better choice for specific situations.

Compared to electronic signals on copper cables, optical transmission on fibers provides many obvious advantages: higher bitrates, lower loss of signal quality, immunity against electromagnetic influences and noise, lighter and more supple carrier material, corrosion resistance, and advanced security (since optical signals are more difficult to tap). From an operator's point of view, however, the most striking argument is given by the incurred cost per transmission unit. At first sight, optical systems have a much higher price than electronic devices, but they also offer a much higher transport capacity. Electronic transmission speed has reached its limits imposed by physics, with a maximum bitrate around 10 Gb/s. So, further capacity expansion is only possible by massive parallel use of devices. This correlates with a cost explosion when demanding a moderate capacity increase, say by a factor of two or four. Optical transmission already enables higher bitrates than electronics, and further advancements in ultra-short light pulse emission give rise to expect further bitrate boosts. Moreover, optics provides further unique opportunities for capacity expansions with low effort: multiplexing.

1.1.2 Wavelength Division Multiplexing (WDM)

In general, the term *multiplexing* refers to techniques which allow for a shared use of a scarce resource. The basic idea of this principle is to divide a common resource appropriately into 'smaller parts' which can be used in parallel to fulfill a certain task. In this way, the resource utilization is multiplied without enlarging the resource itself. In most cases, the possibility to multiplex a resource is based on advanced technology which allows to use the smaller parts.

Multiplexing techniques. In telecommunication networks, multiplexing has been applied successfully in several manners. The most simple case consists in the parallel use of a conduit by multiple cables, which is known as Space Division Multiplexing. It is also applied in optical networks by bundling many fibers into one cable. A more prominent example of the principle is given by Time Division Multiplexing (TDM). In order to benefit from an increased transmission speed without

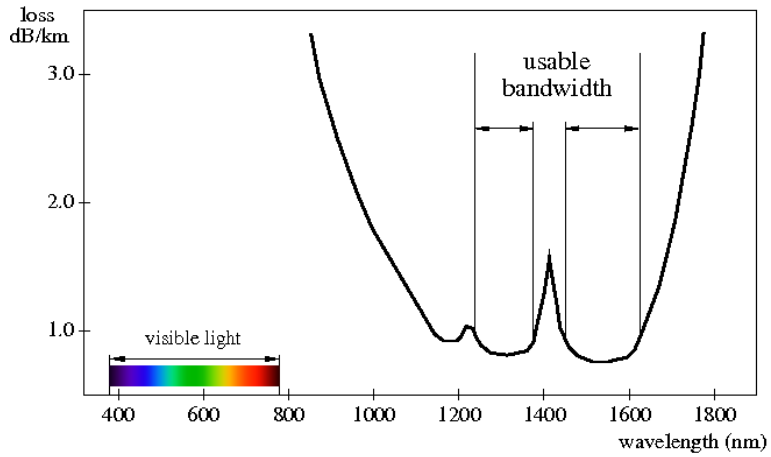


Figure 1.5: *Spectral windows of an optical fiber with low-attenuation.*

increasing the capacity granularity of the connections, a high-bandwidth bitstream is subdivided into several time slots, each carrying traffic on a lower bitrate. As last example with an obvious similarity to light wave transmission, we describe an application from wireless networks using radio waves to transmit information through the air. Equipment improvements have limited the frequency deviation during the transmission, allowing to divide the whole usable bandwidth into small frequency windows which can be used simultaneously for many parallel transmissions without interfering each other. This is called Frequency Division Multiplexing.

WDM. *Wavelength Division Multiplexing* (WDM) is based on an idea very similar to Frequency Division Multiplexing. Instead of radio waves and frequencies, light waves propagate on optical fibers with a certain *wavelength*. The usable spectrum for optical transmission is restricted to specific wavelengths which provide low signal quality loss, see Figure 1.5. While visible light has wavelengths of 380 to 780 nm, the preferred bandwidth for optical signals ranges from 1270 to 1610 nm, excluding small bands with high attenuation. These voids separate the full range into windows. Within the spectral windows, all wavelengths have (nearly) equivalent propagation properties and, moreover, deviate not much from the emitted wavelength during the transmission. By use of high precision transmitters, an arriving light wave covers a small spectral range around the emission wavelength at the fiber end, but this range does not fill the whole low-attenuation spectrum. Hence, it is possible to transmit several optical channels in parallel by use of different wavelengths. As long as the chosen wavelengths are not too close together, the received light waves can be properly separated. This way, it is possible to divide the available spectrum into different wavelength ranges which can be used simultaneously.

Enabling equipment. For the generation of multiple wavelengths, the optical channel access points can be equipped with series of transmitters, each emitting a fixed wavelength of the supported spectrum. Alternatively, a *tunable laser* allows to control the emission wavelength on demand. The improved scalability of the access points makes tunable lasers the favorite transmitter choice for network operators. Figure 1.6 sketches WDM on an optical fiber and the corresponding equipment. At the beginning, each optical channel is set up by a transmitter on a different wave-

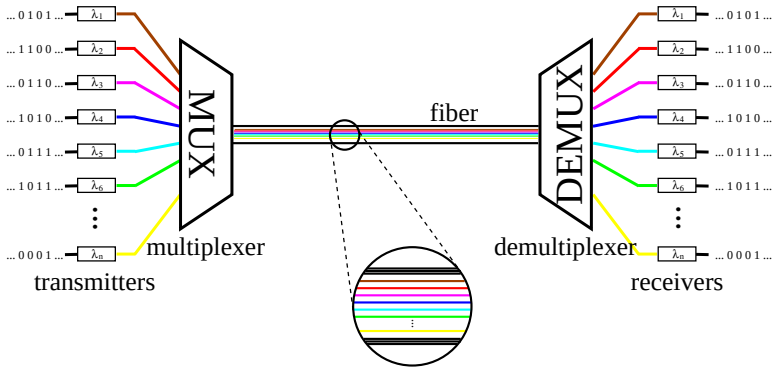


Figure 1.6: *Wavelength Division Multiplexing (WDM).*

length (represented by different colors) within the usable spectral range. To highlight the fact that WDM is applied, we call such a channel also *wavelength channel*. A *wavelength multiplexer* combines these channels to a mixed light transmission sent onto a single fiber. At the end of the fiber, a *wavelength demultiplexer* decomposes the combined transmission into the different wavelengths and guides every wavelength channel onto a separate fiber for individual access. We subsume the couple of a multiplexer and a demultiplexer as *WDM system*.
Bundling multiple wavelength channels to a combined transmission allows for the efficient use of fibers. Within the bundle, each wavelength channel represents a separate optical transmission which underlies the disturbing effects described in the previous subsection and thus needs to be refreshed on long fibers. A so-called *Erbium-Doped Fiber Amplifier* (EDFA) allows to reamplify all combined signals at once. For this, the channels traverse a piece of specific fiber whose doping additive is spurred by pumping light. This power injection reamplifies a whole spectral range, covering the low-attenuation range of optical fibers. Since demultiplexing and multiplexing is not necessary for this way of signal refreshment, EDFAs constitute a major breakthrough for operating WDM, allowing to benefit from high capacity against low effort.
A further advantage of applying WDM on optical fibers consists in the opportunity to vary the transmission capacity of a fiber just by the exchange of the equipment placed at the fiber's end. Having in mind that the major cost portion in expansion of a network's transmission capacities in absence of multiplexing techniques is incurred by "digging" work for the installation of new cable (or fiber) connections, the beneficial impact of WDM becomes clear. Hereby, capacity variation does not only refer to increase, but can as well mean to downsize a fiber's capacity if, for instance, a former 'data highway' becomes obsolete due to adding shortcut connections which then bear the lion's share of traffic.

WDM capabilities. In the early 1990s, WDM was first used to multiply the channel capacities of optical fibers. Henceforth, this technique was improved continuously, step by step enabling more channels and higher bitrates to be multiplexed onto a single fiber, see Figure 1.7, up to so-called *Dense Wavelength Division Multiplexing* (DWDM) used today. The light propagation properties of a fiber and the quality of the applied transmitters determine how many wavelength channels of which bitrate may be multiplexed. The progress in fiber and transmitter technology

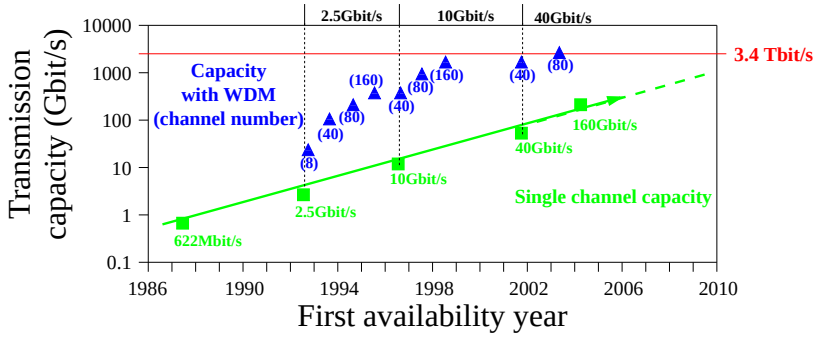


Figure 1.7: *WDM enabled bandwidths on a single optical fiber (see Ehrhardt [41]).*

thus drives also the development of advanced WDM systems. For the planning of optical networks, however, one must take into account that not any WDM system can be installed on any type of fiber. Nowadays, WDM systems with 160 wavelengths each operated at 10 Gb/s over up to 4000 km are common, while the laboratories already in 2001 reported on successfully operated systems combining 273 wavelengths with 40 Gb/s each—a total of 10.9 Terabits per second—over 117 km (cf. [126]) or 300 wavelengths with 11.6 Gb/s each—so roughly 3.4 Terabit per second—over 7380 km, see [105]. To illustrate this volume: 3.4 Terabit corresponds to the contents of a stack of completely filled CDs that reaches the height of about 63 meters (or 6.3 meters without CD covers)—transmitted on a single fiber within a second, say, from Berlin to New York!

1.1.3 Switching

With the application of WDM, optical transmission becomes a powerful technology for high-capacity long-distance data transfer. To interconnect all points, it is theoretically possible to establish a direct fiber link for each pair of nodes. Alas, this yields a costly and wasteful infrastructure. A more efficient way is to use a sparse network structure and to appropriately forward channels along a path from origin to destination. This requires a flexible handling of channels within the nodes.
When an optical channel reaches the end of a fiber, the data flow has to be guided right to continue its way through the network. This guiding functionality is called *switching*. In old telephony networks, the requested call connections were switched manually by physically linking the associated lines. To forward an optical channel from one fiber to another, it is similarly possible to agglutinate the two fibers, called *splicing*. Once established, such a fiber pasting is fixed and cannot be changed easily. This method hence yields a static pattern of established connections between the fibers. However, modern voice and data networks need higher flexibility to set up and shut down connections with less effort. For this, software-controlled reconfigurable switching devices are employed in two types.

Electronic switching. For electronic transmission channels, the switching is performed by a so-called *Digital Cross-Connect* (DXC). A DXC offers a number of interface ports to the transmission links. There are incoming ports on which channels enter the DXC and outgoing ports on which data flows leave the DXC. Internally,

a reconfigurable switching matrix maps every incoming port to an outgoing port, guiding the data flow accordingly.
To employ DXCs in optical networks, the optical signals have to undergo an o-e-o conversion for the switching. Each channel first traverses a receiver to be transformed back to electronic form before entering a switch. The DXC then directs the signals either to access equipment if terminating at this node, or via a transmitter towards the next fiber link to traverse.
With the rapid growth of fiber transmission capacity, especially driven by the introduction of WDM, electronic switching equipment turned more and more into a throughput bottleneck. Only a cascading of switching matrices and interface equipment is able to cope with the large number of high-speed data streams delivered on each fiber. As desirable alternative, switching optical channels directly eliminates the need of electronic devices. The development of this innovation therefore became a main research topic during the advent of WDM networks in the 1990s. The first commercially available devices have then been presented at the beginning of the 21st century.

Optical switching. An *Optical Cross-Connect* (OXC) offers the same functionality as a DXC without o-e-o conversion. If WDM is employed on the links, the bundled signals are de-/multiplexed outside the OXC to provide access to single wavelength channels. Each light wave entering the OXC at an input port is then guided towards an output port according to the current configuration of the switching matrix. Several switching techniques are applied, for example, by Micro-Electrical-Mechanic Systems (MEMS) using small adjustable mirrors to reflect the lightbeams towards the outgoing fiber, or by passing a fluid medium in which a refracting bubble can be created if the lightbeam shall be deflected (a technology evolved from ink printer operations, see Ferguson [47] for more details). The switching matrices are always constructed such that any input port can be mapped to any output port. Hence, an $n \times n$ OXC can switch n incoming channels arbitrarily to the n output ports it offers. Moreover, the wavelength of a channel does not influence its switching possibilities and vice versa, i.e., an OXC can switch every wavelength, and the switching does not change the wavelength of operation unless further devices are applied.
The full switching flexibility provided by an OXC is not always needed. Especially in WDM ring networks, most of the wavelength channels at a node carry transit traffic to be handed over to the next fiber link. Only few optical channels have to be branched off, i.e., are dropped out of or added to the bundled stream. Such a restricted switching functionality is provided by an *Optical Add/Drop Multiplexer* (OADM). An OADM allows to access a small number of specified wavelengths while bypassing all others. It can be seen as a small OXC offering few configurable ports. Figure 1.8 illustrates the functionality of an OXC and an OADM schematically. There are further switching devices, such as passive star couplers, which are not discussed here in detail. To simplify further explanations, we subsume DXCs and OXCs (in any variant) as *switches*.
Like any manipulation of a transmitted light wave, direct switching of optical channels also influences the signal's quality. Depending on the applied technique, the light wave is disturbed by impairments of mirrors and traversed mediums as well as slightly improper adjustments. Hence, optical switches in general tamper an optical channel, which again shortens the maximum transmission length.

Figure 1.8: *Schematic functionality illustration of two optical switching devices: an OXC (at left) and an OADM (at right).*

1.1.4 Wavelength conversion

In networks combining WDM and optical switching, the wavelength of an optical channel becomes an important issue. Assume that an optical channel operated on wavelength λ enters an OXC and is switched onto a next fiber to continue its path through the network. If another channel already occupies wavelength λ on the same fiber, a *conflict* would occur: Any wavelength on a fiber cannot be used by more than one channel. In case other wavelengths on the outgoing fiber are still available, a *wavelength conversion* would overcome such a blocking situation. A device allowing to exchange the wavelength of an optical channel is called a *wavelength converter*. Studies have shown the profitability of wavelength conversion in various measures such as blocking probability for new connections, network load, or realizable traffic, see Ramamurthy and Mukherjee [145] for a survey.

Technical realization. A simple way to accomplish this functionality is by o-e-o conversion using a transponder. The alternative, a direct optical wavelength exchange, is physically a difficult task. The main basic approaches separate into two categories: those that apply optical gatering effects, and those that apply some kind of wavelength mixing. However, to our knowledge there is no commercially available device that performs optical wavelength conversion by now. Hence, this function is realized via o-e-o conversion, with a full signal regeneration as beneficial side effect. Moreover, an o-e-o conversion provides an unlimited conversion range, i.e., any wavelength can be converted into any other, and can be applied on single wavelength channels separately.

At this point, we have introduced all main 'players' in an optical network: transmitters, receivers, fibers, WDM systems, amplifiers, regenerators, switches, and wavelength converters. For understanding their coordinated interplay in order to realize communication, we will next discuss the operation of optical networks.

1.2 Operation of optical networks

The very first telecommunication networks have been pure telephony networks. While their assembly clearly marks a revolutionary progress for mankind, their operation was organized quite simple: Each network user was connected to the nearest telephone exchange by wire, and setting up a call to another network user was real-

ized by a kind operator who manually linked the appropriate lines (whereupon the final connection could pass several exchanges). This homey picture of how a network is operated has meanwhile changed: Modern high-speed data networks form complex puzzles designed to master a maximum throughput of information. In this section, we briefly outline those operational aspects of optical networks which are relevant for the planning task: the physical network structure, data traffic, connection types, and virtual connection structures.

1.2.1 Physical network

Basically, a physical telecommunication network consists of a set of *nodes* and a set of *links* each connecting a certain pair of nodes. The nodes represent the access points through which data traffic is either brought into the network or discharged from it. There can also be pure transit nodes where traffic is only bypassed but not accessed. Together, the nodes and links form the *physical topology* of the network.

Hardware installation. The hardware devices discussed in the previous section are installed either on the physical links or at the nodes of an optical network. The links carry the transmission equipment by which the data is transported between the nodes: fibers (together with amplifiers) and WDM systems (with the multiplexer at the begin and the demultiplexer at the end of a fiber). Regenerators can be installed on fibers which are too long for direct transmission. As only single channels can be fully regenerated, an additional demultiplexer and multiplexer pair is needed to enable access to each channel in such a case. Since individual (wavelength) channels are anyway accessed at the nodes for switching, regenerators are usually placed at nodes as well. Transmitters, receivers, switches, and wavelength converters are always located at the nodes.

Signaling and transport. The physical network is composed of two substructures: the *signaling network* and the *transport network*. Signaling prepares the data transmissions by accomplishing a proper adjustment of all installed devices (esp. switches) and keeps track of their states. This also includes the detection of failures and initiation of recovery operations. A physical topology with adjusted devices then forms the transport network on which the data traffic is processed. Although signaling is vital for the operation of telecommunication networks, we do not investigate it further. It transacts on a separate infrastructure and uses special equipment. In this thesis, we focus on the design of transport networks.

Network levels. Transport networks are usually structured hierarchically: Subnetworks at several levels are connected to the next higher level by well-defined interface points located at the nodes, see Figure 1.9. Each subnetwork covers a certain geographical region and constitutes an autonomous (partial) network. From lower to higher levels, the traffic is more and more aggregated, with the largest bandwidths occurring in the *backbone* or *core network*. Optical networks, best suited for such requirements, occur therefore mostly as backbones, although operators tend to use more and more optics throughout the entire infrastructure, even in access networks, as documented by the recently launched slogans as Fiber To The Home (FTTH), Fiber To The Building (FTTB), Fiber To The Curb (FTTC), and similar

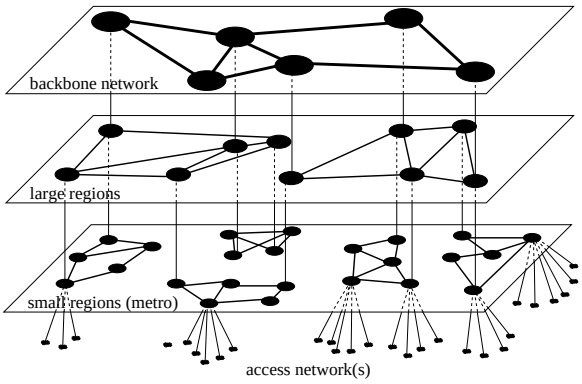


Figure 1.9: *Hierarchically structured network.*

concepts. However, we focus on the backbone in this thesis.

1.2.2 Data traffic

The installed equipment enables the transport of data through the network. There are diverse types of *(data) traffic* which differ in the *service quality* for the client. For instance, the provider can offer a service with a transmission guarantee or an assured connection availability, e.g., for phone calls, emergency services, or video-on-demand. These services need to preallocate the required bandwidth in the network. In contrast to this *allocated traffic*, transmissions or connections without such a guarantee are operated as *best-effort traffic*, i.e., the traffic can be dropped (or postponed) and connections can be preempted in case a failure occurs or higher quality service traffic enters the network. A typical example is email delivery.

Protocols. Data traffic in a network is managed by *protocols* which organize the transmissions. Various protocols are in use. Among the most prominent is the Internet Protocol (IP) which portions data into small packets. A header with origin, destination, and other attributes is added to each packet. The packets are then pushed into the network and take their routes independently. In each traversed node, a so-called *router* reads the headers and decides individually for each packet to which next router it is sent until the destination is reached. Due to this mechanism, IP traffic is called *packet-switched*. At the recipient node, the contents of the packets are finally recomposed to the original data form. Since an IP data transmission does not use a single dedicated connection to be preestablished for the transport, but transmits the data packets on (potentially) many different routes, IP is said to be *connectionless*.
Connection-oriented protocols base on another principle. Prior to an actual data transfer, an end-to-end connection called *circuit*[2] is established in the network and provides a certain transmission bandwidth between the connected nodes. The data

[2] Originally, *circuit* refers to a telephony connection, where a conversation required an individual wire for each direction. The bidirectional end-to-end connection by twisted pair cables in fact could be seen as a circuit. Over time, circuit became established as term for any form of end-to-end connections, although today's high-speed data circuits need not be composed of opposite connection pairs.

is then conveyed along this preconfigured circuit, which makes the traffic *circuit-switched*. An example is the *Synchronous Digital Hierarchy* (SDH) providing *Synchronous Transport Modules* (STMs) with various bandwidths as preconfigurable circuits.
Optical transport networks are circuit-switched as well. They offer preconfigured end-to-end optical circuits on which the data is directly transmitted between the nodes as *bitstream*. A bitstream enters the network through a transmitter at its *origin node* and is sent to a receiver at its *destination node*. Any bitstream is operated at a bitrate determined by the interface devices, the transmitter and the receiver. The market offers devices operating on different bitrates, e.g., 2.5 Gb/s or 10 Gb/s. In principle, circuits with different bitrates could be used simultaneously in a network, since most hardware, as an OXC or WDM system, is transparent for different bitrates. Furthermore, this allows for *grooming*, merging several lower bitrate streams into one with higher bitrate for a common connection part. However, parallel use of bitstreams with different bitrates and grooming substantially increases the complexity of planning and traffic management and also complicates the operation with further disturbances, e.g., as a high bitrate connection disturbs lower bitrate connections more than reverse. Therefore, optical networks in practice are often operated with a unique bitrate. We do not take different bitrates or grooming into account and assume all bitstreams to have equal bitrates.

Network layer. The infrastructure for each type of traffic forms a *layer* in the network which is characterized by the applied protocol or technology, for instance the IP layer, the SDH layer, or the optical layer. Since the protocols do not matter about the transported content, the layers can be embedded into each other, i.e., IP traffic can be transported on an SDH layer, which itself can be client of an optical layer to transmit its STMs (full of IP packets). Due to the individual payload granularity, some layers, like IP or the Asynchronous Transfer Mode (ATM) technology, qualify more for transport of user data with usually moderate volumes, while the optical layer or SDH with capacities like STM-16 (i.e., 2.5 Gb/s) are better suited for higher network levels where such bandwidths are required for transmission of aggregated traffic flows. From an operational point of view, the embeddings produce traffic that is organized by protocol stacks, such as ATM over SDH or the above example IP over SDH over WDM (where WDM stands for the optical layer).

It is also a matter of network planning to select which layers to employ and to devise their embedding. These strategic decisions are based on the services to realize. They are usually taken prior to the network (layer) design. Currently, first attempts are undertaken to carry out a multi-layer design simultaneously in simplified settings (see, for instance, Orlowski and Wessäly [133]). Note that the design of optical networks with different bitrates and grooming forms such a multi-layer problem as well. Traditionally, however, the layers are considered individually, and we follow this direction, focusing on the optical layer in higher detail. For a single layer design, the planner has just to take care for sufficient transmission capacities in the appropriate unit. So, we abstract in the following from the specific transmission contents and focus on the circuits in the optical layer and their properties.

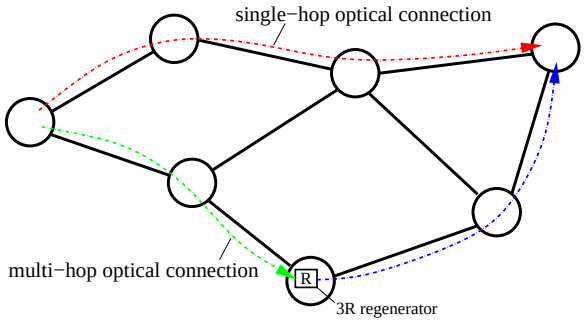


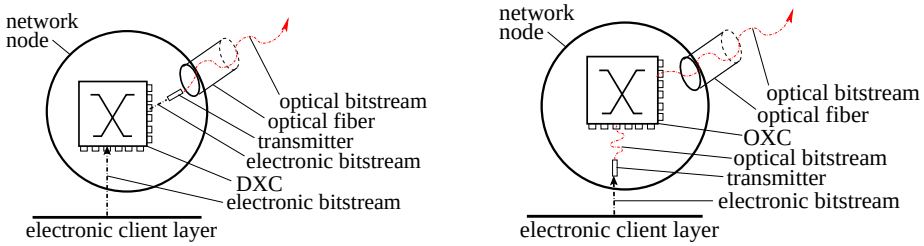
Figure 1.10: *Optical network with a single-hop and a multi-hop connection of the same node pair, the latter being fully regenerated in the bottom node.*

1.2.3 Optical connections and lightpaths

We refer to circuits as end-to-end connections in the optical layer as *optical connections*, with optical transmission on all links. In the nodes, however, signal switching can be performed in two ways. Switching traffic by a DXC implies an o-e-o conversion of the bitstream, i.e., the optical signal has to be transformed to electronic form before traversing the switch and afterwards converted back to optical form before being sent onto the next fiber. Superseding such o-e-o conversions, an OXC can directly handle the optical signals, forming non-interrupted optical connections which carry the information on a whole path without intermediate conversion to electronic signals. Such an ongoing optical connection is called a *lightpath*. Unless further equipment is employed, lightpaths have two characteristic properties: a restricted length due to the limited distance of an optical transmission, and wavelength continuity, i.e., operate on the same wavelength across all links and nodes.

Conversion and regeneration. In case of optical switching, the signal transmitted along an optical connection can also undergo 3R regenerations or wavelength conversions. Today, both manipulations are performed by use of transponders, i.e., as an o-e-o conversion, while direct optical realization is under development. In order to anticipate technical progress, we abstract from the current realization and rather interpret regeneration and wavelength conversion in a more conceptual way as follows. Since wavelength conversion manipulates only a certain signal feature, we consider it as an optical function which does not interrupt a lightpath. Hence, a converter breaks wavelength continuity (at its installation node), but not the total length restriction of a lightpath. In contrast, a regenerator fully refreshes the optical signal, which allows to bridge another full transmission distance. Therefore, we understand a 3R regeneration as terminating a lightpath and setting up a new one to continue, including a possible wavelength exchange.

Single- vs. multi-hop. An optical connection can be composed of one or more consecutive lightpaths, each counted as a *hop*. Hence, a *single-hop* optical connection consists of a single lightpath covering the entire bitstream path, from origin to destination. If the optical signal along the path is regenerated in some node, the arriving lightpath is terminated, and a new lightpath begins, making the connection *multi-hop*. Figure 1.10 shows an example for both kinds of an optical connection. In case only DXCs are employed throughout all nodes, each lightpath spans only



(a) *DXC.* (b) *OXC.*

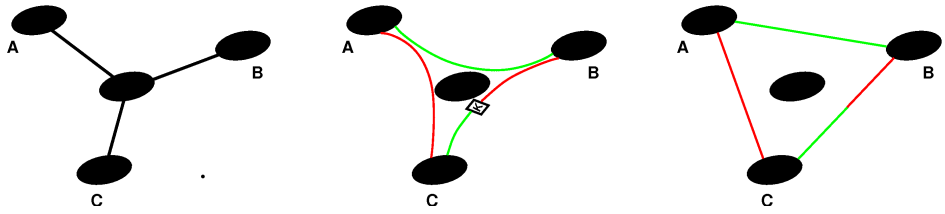
Figure 1.11: *Bitstream switching in a connection endnode equipped with a DXC (a) and an OXC (b).*

a single link, and the hop number of each optical connection corresponds to the number of traversed physical links.

Capacity consumption. Along its preconfigured path, an optical connection consumes an optical channel on each traversed link and a pair of an input and an output switching port in each traversed node. To keep the network flexible, the connections are also switched in their origin and destination nodes, otherwise their first and last link would be fixed and could not be changed, impeding dynamic circuit rearrangements according to a changed traffic pattern. So, a bitstream entering the network in the origin node first traverses a switch which forwards it onto the first fiber. Similarly in the destination node, an arriving bitstream is forwarded by a switch towards the client layer. As a consequence, each optical connection also consumes switching ports in its endnodes.
Figure 1.11 illustrates the bitstream at its origin node switched by a DXC or an OXC onto the first fiber to traverse. If switching is performed by DXCs, an optical connection consumes additionally a transmitter at the beginning of every link (cf. Figure 1.11(a)) and a receiver at the end. If OXCs perform the switching, only one transmitter is employed before the first OXC and one receiver after the last OXC, and moreover a 3R regenerator each time the lightpath is regenerated, and a wavelength converter each time a wavelength exchange is necessary.
Another capacity consumption issue concerns the physical placement of regenerators and wavelength converters at the nodes. Theoretically, it is possible to install both devices as stand-alone units towards which a lightpath has to be guided to enable the manipulation. In this case, the appropriate lightpath would occupy *two* pairs of switching ports, one when entering to be switched to such a device, and one afterwards to be switched onto the outgoing link. However, inspired by today's utilization, we assume that both regenerators and wavelength converters are located between (de-) multiplexer and optical switch, and thus can be plugged selectively at the lightpath to manipulate. So, a lightpath needs a single port pair in each traversed node, whether regeneration and/or conversion is applied or not.

1.2.4 Virtual topology

In a telecommunication network, a link generally represents a kind of connection between two nodes, enabling (direct) data transport between them. For instance, installed optical fibers or copper cables carry the data physically from one node to



(a) *Physical topology.* (b) *Lightpaths.* (c) *Virtual topology.*

Figure 1.12: *Example network with (a) physical topology, (b) some lightpaths, and (c) the associated virtual topology.*

another, a connection referred to as *physical* link. However, links are not restricted to reflect such hardware connections only.

Virtual links. Lightpaths in an optical transport network serve as direct optical transport lines without intermediate electronic layer access and can be interpreted as *virtual links* between the optically connected endnodes. A virtual link between two nodes exists whenever one or more lightpaths connect them, and the number of such lightpaths defines the capacity of the virtual link. On top of the node set, these virtual links form another network structure, called *virtual topology*. A small example in Figure 1.12 illustrates the relation between the physical topology, lightpaths, and the virtual topology.

Special topologies. Basically, an optical network with (some) optical switches holds optical connections composed of one or multiple concatenated lightpaths. For particular network realizations, two special topology variants occur. First, if only DXCs are used throughout the network, any lightpath spans a single physical link only. Then, virtual and physical topology coincide. Second, the other extreme occurs whenever exclusively single-hop optical connections are established (e.g., demanded as operational requirement). In this case, the virtual topology reflects exactly the traffic requirements between each pair of nodes. The most general scenario is therefore given if both prerequisites do not hold, i.e., in an optically switched network and multi-hop traffic connections.

Directions. By its nature, a bitstream is directed from the origin to the destination node. Similarly, each optical channel on a fiber is also operated as a directed transmission, and with WDM, all wavelength channels on a fiber have the same direction. From this point of view, optical transmissions are *unidirectional*. In practice, however, fibers are typically installed pairwise, carrying optical channels in both directions. When additionally equipped with WDM systems, equivalent devices are applied on both fibers. Any fiber pair thus offers the same connection capabilities in both directions. This allows to consider the transmission capacity on the physical links as being *bidirectional*.
Furthermore, also the connection requests are typically bidirectional, i.e., the connection between two nodes shall provide the same transmission bandwidth in both directions. In general, a connection may consist of multiple circuits, and bidirectionality means that the same number of optical connections is established in each direction. In this case, the virtual topology becomes bidirected.
Each optical connection is specified by its path in the physical topology on which the

associated bitstream is physically transmitted. Although bidirectional connections could theoretically reside on circuits routed independently for each direction, it is common practice to follow the same path in both directions, for the sake of simplifying the network management. In particular, lightpaths are usually routed oppositely on the same physical path. With that requirement, the optical connections (merged pairwise appropriately) become also bidirectional.
Summarizing, directivity affects three issues: the transmission capacities of physical links, the implementation of lightpaths in the physical topology, and the connection bandwidths of virtual links. Since the distinction of all the options related to uni- and bidirectional capacities, circuits, and connections would complicate further explanations and mainly just raise the formal effort, we take a simplified but realistic perspective in what follows: We assume all these items to be bidirectional as default and dwell on the distinction only if it implies essential differences in the current context. In addition, we do not anymore distinguish input and output ports at the switches, and refer with port numbers to bidirectional ports. Hence, an $n \times n$ switch (of any type) can handle n undirected circuits.

The data traffic is transmitted on the established optical connections traversing many optical devices whose frictionless operation is essential for successful data transports. Admittedly, hardware is not immune against malfunctions and disturbances, and connections are very sensitive against such failures. For provision of reliable services, operators have to take possible disruptions into account, planning their handling precautionary.

1.3 Survivability

Telecommunication networks are menaced by manifold potential failure sources: hardware malfunctions, operational mistakes by falsified signaling or human errors, physical destructions by construction works—with diggers cutting cables as a classical example—or natural disasters, terrorism attacks[3] , among many others. Each of these scenarios affects a network to a certain extent, from local trouble to severe demolitions. Nevertheless, as long as a failure does not disable too much hardware, network operators want to maintain at least those connections for which service guarantees have been given or which are vital for the network operation, as holds in particular for the backbone. To this end, they ask for designs of so-called *survivable* networks. In this section, we discuss survivability of telecommunication networks and its realization. Moreover, various known concepts are presented and compared. For further information on the topic, we refer to the discussions of Gerstel and Ramaswami [54, 55] and the book of Grover [61] which provide an excellent introduction with special emphasis on optical networking.

[3] Terrorism need not be targeted directly on a network to damage it. As sad example, the severe terror attack at September 11 in 2001 destroyed the World Trade Center in New York, and with it the endpoint of some Transatlantic connections to Europe, located in the cellars. For redundancy, each of the towers hosted a separate node. Their common failure was one of the main reasons for the Internet traffic breakdown soon after the attack.

1.3.1 Basic aspects

The *survivability* of a telecommunication network is its capability to 'survive intermittent failures', i.e., to restore and supply (at least a defined portion of) the data traffic in situations where some established connections are interrupted. Although this rough definition clearly needs to be specified in many aspects, it states the fundamental principle, also formulated as standard in the ITU Recommendation G.841 [159] (for SDH networks), and serves as orientation for a more precise characterization of survivable optical networks.

Network states. If a failure occurs, one or more established connections are interrupted, and one or more hardware devices break down. This turns the *normal operation state* of the network into a specific *failure state*. Network providers are clearly exerted to repair a damage as fast as possible in order to return soon to normal operation. Since this usually takes some time, particular failure state configurations are preplanned for the most relevant types of damage.

Failure types. Any failure yields the disruption of some circuits as end-to-end connections carrying data traffic. An appropriate treatment clearly depends on the failure extent. Operational mistakes are typically easy to repair by (automated) traffic management reactions, quickly restoring the normal operation state. Equipment outages require manual intervention of engineers to replace broken hardware, which takes much more time. As preventive measure, most devices are already constructed in a redundant way by the manufacturers: Parts important for the operation are assembled twice, such that an outage can be (immediately) compensated by delegation to the reserve part. This allows for repairs without affecting the traffic. In contrast to these (very) short-term affections, the outage of non-redundant hardware or multiple devices at once has a more serious effect, entailing the failure state to persist for a long(er) period during which the traffic has to be accommodated differently. Instead of taking all possible combinations of failing devices into account, operators usually consider outages only by means of entire physical links or nodes to simplify survivability considerations. So, whenever at least one non-redundant device on a location fails, the whole location is regarded out-of-work (which in fact is often the case during repair). In practice, the most frequent case is a *single link or single node failure* in the physical topology. Therefore, survivability planning includes particularly these situations. A link breakdown disrupts only the optical channels transmitted on it, while a node failure cuts the connections on all adjacent links at once. Moreover, traffic originating or terminating at a failing node cannot be recovered and remains inoperable until repair. However, single link and single node failures have only local consequences to take care for in the network design. The simultaneous occurrence of *multiple failures* can have major influence on the complete operation, e.g., by disconnecting some network parts completely. Fortunately, multiple failures are much more unlikely and thus are typically not considered in the planning process.

Recovery. Outage of devices can also be caused by maintenance works with a controlled shut-down. The traffic management usually prepares these situations by redirecting affected connections in advance. Actual failures occur abruptly and loss of traffic on the cut connections is unavoidable. If spare capacity is available, either

additionally installed or becoming free by disrupted connections, protected traffic (not originating at a failing node) can be redirected onto still operable *backup connections*, i.e., surrogate connections to take over traffic from failed connections until the malfunction has been fixed. Shifting traffic to such backup connections is called *recovery*.
Initiating a recovery requires first to identify the failing devices and to signal this information to any location that has to perform changes. After failure detection, any reconfiguration operation takes further time, e.g., to (re-)adjust switches, drop best-effort traffic to free required capacities, establish new circuits, or other mechanisms. The *recovery time* refers to the period from the failure incidence until finishing all initiated emergency operations and reaching the preplanned exceptional configuration. Providers clearly prefer a minimum outage duration to keep traffic loss as low as possible. Hence, the (theoretical) recovery time is an important measure for the quality of any survivability scheme.

Managing layer. In a layered network architecture, survivability can be implemented separately in any of the traffic layers. For instance, an IP over SDH over WDM network operator can choose to design the IP layer in a survivable manner independent of the used transport technology (stack), or plan the SDH connections carrying protected traffic in a redundant way within the SDH layer. Whenever a client layer applies survivability for its own, the used transport layer(s) inherit routing restrictions as well, in order to guarantee that a client connection and its designated backup connection(s) are not mapped onto routes in the transport layer which share resources. Moreover, a transport layer is not limited to serve for a single client layer or service, and merging multiple client layers can result in complicated restrictions. A simpler alternative is to implement the survivability mechanism directly within the transport layer, providing (optional) protection for any client layer/service. In view of the high amounts of data carried by the optical transport network, the network core is usually requested for its own fast and robust survivability mechanisms.

1.3.2 Known concepts

The literature offers a wide range of survivability mechanisms for various types of networks and architectures. A complete survey and comparison of all these schemes and their variants is beyond the scope of this thesis. Instead, we offer a brief overview on some basic concepts and compare those of their properties which are relevant for practical implementation in optical networks. A more comprehensive survey on survivability concepts, variants, and categorizations can be found in Grover [61] and Pióro and Medhi [140].

Comparison criteria. Basically, survivability concepts try to find a trade-off between (at least) two concurrent goals: efficient use of backup resources, and ease of network management in the application of an as fast as possible recovery mechanism. While the first goal can be simply benchmarked by the cost incurred for installing the required spare resources, the second goal is far more difficult to quantify. In fact, a fair comparison of two mechanisms can only be provided if both are established in parallel and run through the same failure situations. As such a procedure is usually

out of reach, we have to reside on hypothetical measures and simulation. For this, we use the recovery time as main criterion for the evaluation of concepts in terms of network management complexity.

Basic concepts. Survivability schemes for optical networks are mostly based on those developed for SDH which is the precedent core network technology, often used further in (parts of) current networks. A presentation of some main concepts therefore provides a good starting point for our scope. Within the framework of SDH network planning, Wessäly [166] discusses the most important variants together with their integration into solution methods.

Diversification. One of the most basic survivability concepts is *diversification*, introduced by Dahl and Stoer [36]. The key observation is the following: If the maximum fraction of a demand routed through any network node or link is restricted, then the maximum loss of this demand's total traffic in case of any single component failure is restricted as well. So, using a diversified routing for each demand prevents a complete disruption of all corresponding connections, maintaining (at least) a specified throughput percentage in any failure situation. To realize diversification, the normal operation routing model is simply extended by maximum demand flow bounds. No additional backup connections are provided to take over dropped connection's traffic, and thus no backup resources are needed at all. However, diversification is in general no zero-cost concept, since additional routing constraints must be satisfied which can obviate cheaper configurations possible for unprotected designs.
Diversification is a pure loss-limiting concept and does not arrange for recovery of disrupted connections. In telecommunications practice, diversification is therefore not really seen as a survivability scheme. However, it provides a framework for modeling survivability issues and can be extended to protection schemes, as will be shown in Section 2.3.

Reservation. If disrupted traffic is to be recovered in case of failures, provision for additional backup connections is unavoidable. The concept *reservation*, by Minoux [119], does not impose restrictions on the normal operation routing, but uses additional spare capacity to allow for rerouting of the traffic to survive in any failure case. By allowing for a rerouting of all connections, in particular also those unaffected by the current failure, the amount of required spare capacity can be kept low, since the backup routing can make best reuse of all still operable capacities. The drawback of this very resource-efficient concept is the recovery process which can escalate to reconfigure the whole network, including the shutdown of failed or dropped connections, and the complete establishment of all newly designated connections.

Path restoration. The concept of *path restoration*, invented by Wu [171], can be interpreted as a more practical concept between the former two extremes. Similar to the previous concept, there are no restrictions on the routing for normal operation, and spare capacity has to be installed for enabling backup connections. In contrast to reservation, the rerouting in failure situations is restricted to the failing connections, i.e., all unaffected connections have to be maintained. Clearly, any such solution represents also a feasible solution for the

concept reservation, and thus the total cost for path restoration (as measure for total resource consumption) cannot be lower than the cost of an optical solution for reservation. Additionally, it is optional whether still operable capacities occupied by failing connections in the normal operation state can be reused for the backup connections, an issue known as *stub release*. However, the main benefit of path restoration consists in a typically faster recovery, since action has only to be taken for those protected connections that actually fail.

Link restoration. While path restoration allows for an end-to-end rerouting of the affected connections, an even more restrictive variant is proposed as *link restoration* or *span restoration* by Grover [60]. Here, the rerouting of failing connections has also to maintain all non-affected route parts, providing backup alternatives only to circumvent the inoperable network components. For any connection disrupted by a link failure, the backup path has to follow exactly the same route as before between origin or destination and the respective endnode of the disabled link, while the (partly) backup connection bridges only the disabled gap between the nodes incident to the failure. Node failures are handled similarly, leaving the regular routing path unchanged up to the two neighbors of the failed node. This way, reconfigurations focus on the network part close to the failing component. Link restoration forms a special case of path restoration and thus yields potentially larger total cost (or capacity consumption), whereas recovery is fastened further due to the local treatment of failures.

Supporting features of SDH. All of the concepts described so far can potentially be applied to optical networks as well. However, two specific features of SDH contribute particularly to their resource-efficient realization: scalability of traffic flows, and ease of sharing capacities by electronic transmissions. Concerning scalability, SDH flows are organized by a hierarchy of transmission bandwidths, offering connections with different STM capacities. This hierarchy allows for a flexible subdivision of flows into multiple fractions of varying bandwidths which can be routed disjointly according to diversification, for instance. Optical networks in contrast hold indivisible lightpaths as routing units, prohibiting optical connection fragmentations (and even with grooming, flow splittings would be much less flexible). The more important difference indeed concerns the shared use of capacities which is hampered in transparent optical networks by the additional attributing of wavelengths. Spare capacity in an SDH network can be occupied by any two different backup paths as long as it is provided that not both of these paths will be required simultaneously. The same holds for transparent optical networks only under the additional restriction that both paths are operated on the same wavelength, and hence spare capacity can be shared by much less optical connections at once.

Survivability in optical networks. The high data volumes transported by optical technology increase the sensitivity against malfunctions. The breakdown of a single link in the physical topology, even if disrupting just a few lightpaths, can affect many client layer services and connections at once. This effect is even aggravated by a recovery that needs a complex reconfiguration. Path restoration, for instance, can require shutting down many existing lightpaths and setting up many new ones. Such a process takes some time during which only little traffic

is operated until the surrogate configuration is finally arranged. This observation holds generally for restoration schemes. These schemes take care about the presence of sufficient spare capacity to set up a (preplanned) failure state routing, but do not preestablish backup connections. As a consequence, optical network operators tend to implement mechanisms which offer high availability and short recovery times at the price of occupying more resources.
For optical networks, the favorite alternative to restoration is protection, meaning that all backup connections are preestablished and thus reconfiguration effort is kept as low as possible. There are two fundamentally different protection strategies. The first approach is to protect signals by setting up complete backup paths from origin to destination of a commodity. This strategy is called *(end-to-end) path protection*. Alternatively, backup paths can serve to only circumvent individual segments (a link or span) of working paths. Such strategies often restrict to protect single links, but can usually be simply extended to larger segments and thus to node failures as well, and are subsumed as *link* or *span protection*. As with link restoration, the advantage of these schemes is that the recovery switching is performed near the failure, with typically short switching times and no or little failure signaling effort. The drawback is that many different backup paths are required for protecting all working paths completely. In principle, all protection approaches are applicable for complete paths or segments of a path. In the following, we mainly concentrate on end-to-end path protection schemes which are for optical networks mostly used in practice.

Dedicated protection schemes. *Dedicated path protection* sets up two disjoint paths in the network for each protected connection. There are two specific schemes for dedicated path protection:

- *1+1 protection* duplicates each signal and transmits it on both paths. The receiver at the destination then selects the better signal. Dedicated path protection is very fast since in case of the better signal being disrupted, the receiver has only to switch to the other signal. However, this protection scheme is very capacity intensive due to doubling the traffic to be protected.
- In *1:1 protection*, two disjoint paths are set up with the same strategy as in 1+1 protection, but the signal is only transmitted on the working path. The backup path is offered to best-effort traffic. In case of a failure, the interrupted service is switched from the working path to the preempted backup path. The strategy to allocate protection capacity on request after a failure enables more efficient use of the network capacity, but it has a higher recovery time since the switching is not anymore a local decision at the endnode. The failure occurrence has to be detected and signaled in real-time to both commodity endnodes in order to change over to the backup path.

Usually, dedicated path protection is applied in today's SDH ring networks as 1+1 Automatic Protection Switching (APS). The routing of disjoint path pairs is very straightforward in ring networks, and often far more than 100% additional capacity is needed as spare capacity with dedicated path protection.
In meshed networks, dedicated path protection is more capacity efficient, since the routing can be more diverse. With respect to a selectable metric, such as number of hops or total fiber km-length, the paths are for instance routed on the shortest path and on the second shortest path which is disjoint from the shortest path. Note that

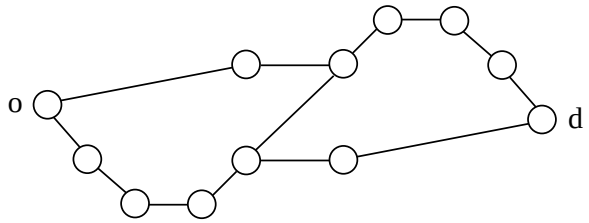


Figure 1.13: *Example network where a pair of disjoint paths between nodes o and d exists, but not if one is a shortest path w.r.t. the hop number.*

such a pair need not exist, see Figure 1.13. Therefore, Suurballe's algorithm and its variations (cf. Suurballe and Tarjan [156] and Bhandari [18]) are often applied to find a shortest cycle in the network that contains both origin and destination of the request. Such a shortest cycle comprises both the working and the backup path, the former typically selected as the shorter one. A case study on dedicated path protection based on several routing schemes in multi-fiber WDM mesh networks is reported in De Patre et al. [139].

Shared protection schemes. 1:1 protection is a special case of general M:N path mapping, where M protection paths are established to protect N different working paths of a commodity. *M:N shared protection* shares capacity in the sense that the spare resources are not dedicated for the recovery of a specific connection, but can be shared by the N connections for different failure scenarios.
A more general *shared protection* concept exists with a large number of strategies for sharing of backup resources between different commodities. Various specific schemes for shared protection in optical networks have been proposed and evaluated, e.g. in [112, 56, 80, 87]. The basic principle is that spare resources can be shared among different backup paths as long as the protected connections will not compete for the same resources at the same failure incidence, e.g., the associated working paths are disjoint in case only single failures are considered. In such a case, we say that spare capacity is *reserved* by a backup path, but not prededicated to it. In *shared path protection*, a fully disjoint backup path is preselected for each working path. If a failure occurs, a real-time signaling phase seizes and cross-connects the shared capacities to establish the requested backup paths. The need for real-time connection setups yields high recovery times, whereas sharing of spare capacities reduces the total amount of backup resources to install. Moreover, shared path protection can take advantage of a meshed topology to further reduce spare capacity consumption, since higher network connectivities in principle generate larger numbers of disjoint working paths which are able to share spare resources.

p-cycles. Finally, we briefly mention the special shared protection concept of *p-cycles*, at first encountered in Stamatelakis [153]. Instead of backup paths, this concept suggests the establishment of backup *cycles* with uniform wavelength, called p-cycles. Such a p-cycle protects all cycle links as well as all cycle chords at once and can be shared among multiple working lightpaths which are disjoint on these links. Adaptions of the original concept include node failures as well. Although the concept is theoretically extendible to protect entire lightpaths instead of individual links or segments, such additional restrictions are uncommon and would diminish both resource efficiency and recovery speed. p-cycles have originally be designed

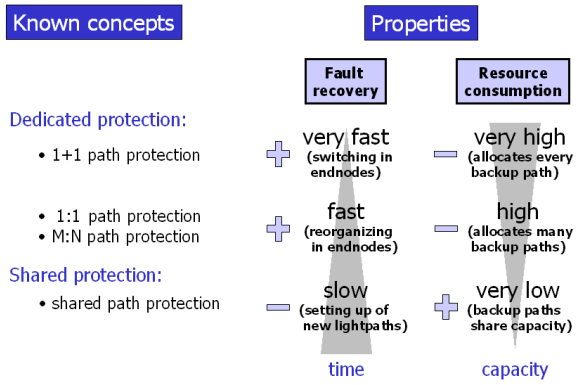


Figure 1.14: *Comparison of the discussed survivability concepts for end-to-end path protection with respect to (theoretical) recovery time and spare resource consumption.*

for link protection and are rarely considered otherwise. Since we focus on path protection schemes, we leave out this concept in the following comparison.

Concept comparison. Figure 1.14 gives an overview about the discussed survivability concepts for end-to-end path protection with a rough classification of their properties. Basically, it can be observed that the faster recovery a scheme provides, the more spare capacity it consumes. Although shared protection is currently widely studied in the literature in plenty of variants, such schemes have yet not entered the networks in practice. In fact, most network providers apply either 1+1 or 1:1 protection due to the fast recovery. A further reason is that shared protection schemes typically complicate the network management. Therefore, we propose in Section 2.3 an alternative scheme which is more easy to realize and offers a restricted sharing of spare capacities to reduce the total cost for robustness.

1.4 Evolution and architectures

Telecommunication networks are anything but static infrastructures. They form an ever-varying system that is affected by many outside influences, ranging from user and utilization fluctuations, integration of new services and operational concepts, a lively market environment, up to technological progress. This motivates providers to regularly adapt and modernize their networks, and hence it is interesting to see how optical networks evolved in practice and which are the final network realizations that we run across today or in the near future, taking emerging technology into account. The virtual topology of a network expresses the current connection pattern, which is regularly adjusted to changed traffic requirements. Moreover, there are issues belonging to mid- or long-term planning, where decisions are not that easy to revise, and thus represent characteristic features of network realizations. Two such features concern the used technology and the applied operation concepts. We call a specific combination of these two issues the *architecture* of a network. In this section, we highlight main architectures of optical networks in practice along their chronological migration. Further information on the evolution of optical networks can be found in Colle [33].

Fiber networks. The evolution of optical networks now lasts for more than three decades. It was initiated by the installation of first optical fibers in the early 1970s. In the following years, the new optical transmission superseded classical electronic transmission on copper cable connections bit by bit. Although the two technologies coexist until present, especially the increasing bandwidth demands in the backbones favored the use of fibers. The arising *fiber networks* were free of any copper cable connections. These networks yet do not get the attribute 'optical', because fiber systems are applied only as one-to-one substitution for electronic equipment, but exclusive properties of the optical medium are not utilized.

WDM networks. The first step to exploit optics is marked by the integration of WDM. The installation of WDM systems on fiber connections started in the early 1990s and created *point-to-point WDM networks*, also called *1st generation optical networks* or *opaque networks*. In these networks, (bundles of) wavelength channels are transmitted between adjacent nodes, whose electronic switching equipment requires a conversion back to electronic signals. Since light waves cannot traverse such nodes, these nodes (and networks) are called *opaque*. Notice that a WDM network need not apply WDM on any fiber, but can also contain fibers carrying a single optical channel, i.e., without installed WDM system.

Optical networks. During the next decade, the rapid progress in WDM technology regularly multiplied the transmission bandwidths on the links. Without immense investments for many further interface devices and cascades of latest DXCs, electronic switching could not master the throughput growth. To overcome the bottleneck of opaque nodes, manufacturers intensively researched on the capability to switch optical channels directly. With the begin of the new millennium, the first OXCs have been presented at commercial technology fairs. This new technology makes the nodes *transparent* for optical channels and yields *2nd generation optical networks* or—to express being current state-of-the-art—simply *optical networks*. These networks represent the first architecture with both links as well as nodes operated optically and hold lightpaths as characteristic feature. Only full signal regeneration and wavelength conversion are still performed by transponders, i.e., as o-e-o conversion. If no o-e-o conversion is allowed, most authors speak about so-called *all-optical networks*. In such networks, the optics forms a closed network layer which interfaces electronics only at the access points.

Hybrid architectures. So far, we have discussed some basic consistent architectures with specific dominating characteristics. Meanwhile, the ongoing network migration yields also various hybrid architectures. We just pick the kind of switching as illustrative example. Instead of a sole technology, an optical network can also mix up of opaque and transparent nodes, known as *optical networks with sparse wavelength conversion* (cf. Subramaniam et al. [155]). While most nodes apply optical switching, some are fully equipped with DXCs and provide full regeneration and wavelength conversion capabilities for all traversing optical connections. Further related architecture variants use optoelectronic nodes with both DXCs and OXCs (cf. Meddeb et al. [113]), or integrate both mechanisms in hybrid switching devices (cf. Cavendish et al. [30]). With a sufficiently detailed level of observation, nearly each network has a unique architecture. However, most of the specific networks tend to become over time similar to one of the basic architectures, which therefore can

be used as representative scenarios.

At present, we experience the advent of 2nd generation optical networks in practice and, considering the emerging research reports, all-optical networks come into reach. There is also a lot of research activity towards direct optical switching of packets (see for instance [39, 100, 138, 162, 165, 174]) or so-called *bursts* as collections of packets heading for the same destination (see [26, 81, 142, 164, 172]). Note that this does not mean IP over WDM, but introduces a new kind of packet-switching inside the optical layer. Since the protocol stack is reduced, the traffic management is simplified, and some equipment becomes obsolete. This makes packet- and burst-switching attractive for the operators. However, packet (or burst) routers need a buffering capability for the time it takes to read a header and to decide about the further way. In addition, packet-based traffic is based on best-effort transmissions and hence can require additional buffering in case a chosen link is congested. Optical buffering yet is not practicably procurable, and packet- or burst-switched optical networks are not expected to enter practice in the near future. Hence, the main subject of this thesis is the planning of circuit-switched optical networks.

1.5 Optical network planning

In view of the investments tied up by a network infrastructure, elaborate planning is required to avoid the waste of any resources. This, however, is a complicated task. Besides the many restrictions imposed by technological requisites and operational guidelines, the planners have also to bear in mind qualitative issues such as reliability and flexibility. Resulting in a problem of high complexity, the overall network planning is typically subdivided into partial tasks. For a hierarchically structured network transporting different kinds of traffic, the single levels and layers are typically considered separately. In this thesis, we focus on the planning of the optical layer as transport network in the backbone.

Basically, designing a network consists in determining an appropriate network configuration. We first discuss this central term and identify closely related issues involved. Next, we list further planning aspects and typical goals to strive for. So far, this introductory chapter provides an overview about design-relevant issues and their options. Some of the resulting design tasks are of particular importance for network operators in practice. The corresponding specifications are finally presented in a concluding overview.

1.5.1 Network configuration

The operation of an optical network relies on its proper configuration—a comprehensive term with various meanings. In this regard, we consider neither technical calibrations nor signaling induced device adjustments. For network planners, the term configuration subsumes the arrangement of hardware (in the physical topology), lightpaths (forming the virtual topology), and their composition to the requested end-to-end connections as an operable network.

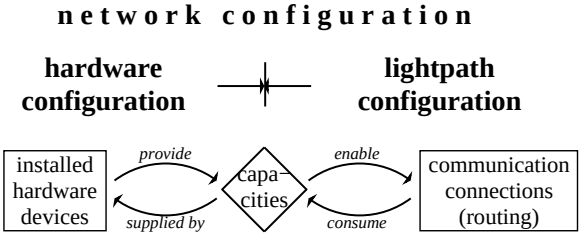


Figure 1.15: *Structure of the network configuration task.*

From this perspective, determining a feasible network configuration comprises the allocation of *hardware devices* such that sufficient *capacities* are provided to establish all needed *lightpaths* for the asked optical connections. This high-level description suggests a structuring as illustrated in Figure 1.15. All decisions concerning hardware belong to the hardware configuration, those concerning the connections belong to the lightpath configuration, and the capacities, provided on the one side and consumed on the other side, chain these issues. In the following, both subconfigurations are described in more detail.

Hardware configuration. The *hardware configuration* of the network refers to the arrangement of the hardware devices in the physical topology: transmitters, receivers, fibers, WDM systems, switches (OXCs, DXCs), regenerators, and converters. The number and location of transmitters and receivers is predetermined by the requested connections, and hence we neglect these devices in the planning. By their specific functions, the remaining devices offer different kinds of capacity: *transmission capacity* on the links and *switching*, *regeneration*, and *conversion capacity* at the nodes of the network.
On the links, the installation of fibers with WDM systems allows to operate a number of optical channels for data transport. Due to a uniform channel bitrate, the transmission capacity of a link can simply be expressed in terms of the number of available optical channels. Depending on whether the distinction of wavelengths is important or not, we account for the number of wavelength channels per wavelength or the total number of channels.
At the nodes, every installed switch provides a number of ports, each allowing to handle a single optical channel. For WDM, multiplexing and demultiplexing of transmissions is performed at the end of the fibers, i.e., before channels enter a switch, see Figure 1.16. This allows to access each channel independently for switching and, if multiple switches are installed at the same node, the total switching capacity accumulates from the single port numbers of the present switches.
Usually, transmission capacities are directly geared to the intended number of channels to transport along a link. The need of switching capacity at a node can also be determined in this *traffic-oriented* way, ensuring that the planned number of traversing optical channels can be switched. Alternatively, the *link-oriented* node dimensioning asks for installing sufficiently many ports to handle all optical channels even if the adjacent links were fully occupied. Such a node configuration is independent of the current traffic pattern and thus keeps the network more flexible for changing connection requirements, but typically requires higher switching capacities at the nodes.
A hardware configuration is completed by the placement of regenerators and wave-

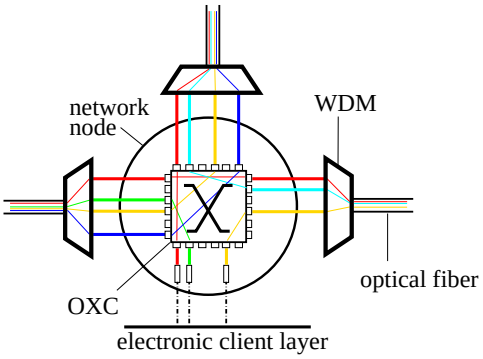


Figure 1.16: *Hardware arrangement at a node for switching (with an OXC).*

length converters, if available and needed. Being pluggable to individual channels, the needed numbers of these devices are directly given by the planned number of traversing channels that require these functionalities.

The placement of all hardware devices together determines the available capacity resources and is called the *dimensioning* of the physical topology. Note that all capacities are expressed in (equivalents of) the number of optical channels (or wavelength channels, if wavelengths are to be distinguished). The supplied capacities serve to establish lightpaths, as described next.

Lightpath configuration. Connections in optical networks are realized by means of (consecutive) lightpaths. A *lightpath configuration* consists of the arrangement of all lightpaths to establish and specifies completely the consumption of all physical capacities. The *routing* of the lightpaths determines their transmission paths in the physical topology. With WDM, a *wavelength assignment* has to be carried out additionally, i.e., assigning a wavelength of operation for each lightpath on each traversed link. Depending on the available technologies, the limited length of lightpaths as well as wavelength conversion capabilities have also to be taken into account for a proper configuration.
We distinguish two kinds of routings according to the following path choice properties. A routing is called *non-bifurcated* if all connections between the same endnodes use the same path, otherwise it is *bifurcated*. Note that bifurcation does not mean to split optical connections by fractions, e.g., subdivide a lightpath into multiple portions routed independently. The lightpath routing is always carried out in entire optical channels. Some services explicitly rely in addition on non-bifurcated routing, for instance high-quality video streaming where transmission on different paths could change the prescribed data order, resulting in replay interruptions. Whenever such a service occupies multiple optical channels in parallel, non-bifurcation requirements carry over to the corresponding lightpaths.
The lightpath configuration of an optical network is schematized as virtual topology, see Section 1.2.4. As direct optical transport line, each lightpath corresponds to a virtual link representing the logical connection of the endnodes.

Network configuration. The *network configuration* comprehends both the hardware and the lightpath configuration. These partial configurations are coupled by the capacities, as shown in Figure 1.15. The capacities provided by the hardware

configuration in the physical topology have to be large enough to realize the lightpath configuration. In case of multi-hop traffic, a network configuration also involves the composition of end-to-end connections by lightpaths, i.e., a routing of the connections on the virtual topology.

Demand and commodities. A network is primary designed to satisfy the traffic *demand*. In backbones, the traffic occurs in an aggregated form, subsuming many data streams originating from various client layers and services. Hence, it is natural to record the demand for planning in this aggregation, too.
We assume traffic demands to be given as a set of *commodities*, each specified by the associated origin and destination node and the requested number of optical connections to establish. Further requirements are imposed by survivability needs, settings for the targeted configuration (e.g., single- or multi-hop traffic, non-bifurcation needs, etc.), and the underlying technology (e.g., opaque or transparent and associated technical restrictions).

Main architectures. Regarding the relation of physical topology, virtual topology, and traffic demands, three main architectures are to be distinguished. Transparent optical networks with multi-hop traffic form the most general case where the virtual topology need not coincide with the physical topology or the traffic demands (interpreting the demand matrix as node adjacency matrix). The two other architectures show special relations. Single-hop traffic in transparent networks yields end-to-end lightpaths, and so the virtual topology equals the traffic demands. In opaque networks, any lightpath spans a single fiber link, and thus the virtual topology simplifies to (a part of) the physical topology (neglecting any multiplicities).

Once established, a properly configured network can serve the requested demands. Over time, changing traffic requirements and technological progress regularly force the network operators to adapt their network configuration and to modernize their hardware. In this regard, network planners have to cope with various questions and design tasks, discussed next.

1.5.2 Network design

The design of a telecommunication network is not a singular task which ends with the realization of the planned infrastructure, but an ever-repeating process with varying requisites in a dynamic environment. Driven by technological progress, operational and organizational alterations, innovative concepts, and last but not least fluctuating traffic, the planner regularly has to adapt the network to serve current connection requests and to guarantee best service quality. This yields several types of typical planning tasks that have to be solved in regular time spans. Short-term planning, for instance, includes connection reconfigurations according to changing traffic patterns which result from corrected or new forecasts. Such operational decisions are easily alterable due to software-controlled equipment. Being harder to revise, the installation of new devices is a matter of mid- or long-term strategic planning, in particular adding new fibers or extending the physical topology by new links or nodes. In the following, we discuss some particular aspects of planning a network.

Compatibility. In particular for decisions with a long-term effect, careful planning must be applied. For example, the choice of a device manufacturer often prohibits simultaneous use of competitor devices because of incompatibility. Since the available hardware is part of the input for network design, such decisions can be integrated with in the planning, but at the price of a high complexity increase. The better alternative is to select some scenarios and to compare the resulting networks. In this way, the planning can support the decision maker, but outflank problem complications.

Network upgrade. Similarly, changing an existing physical topology is also a strategic matter. We speak about *upgrade planning* if a network is already at hand and should be appropriately extended and reconfigured, while setting up a network from scratch is called *greenfield planning*. In both cases, each non-existing link or node can potentially be inserted into the physical topology, with an optional setup cost. However, it is common practice to base on a physical topology with reasonably preselected nodes and links for hardware installations.

Dimensioning. The capacity dimensioning task has both strategic as well as operational character. Besides the fundamental requirement to satisfy current connection requests, a more strategic aspect concerns the flexibility of a network. In the best case, the dimensioning is carried out such that minor future adaptions can be realized with no or low effort. Clearly, an optimal preparation of envisioned evolution scenarios would require a multi-period planning—a much more complex problem which just starts to enter research investigations. As practicable alternative, network design can also achieve more flexibility by following simple principles, like link-oriented node dimensioning. Moreover, designing a network with high survivability level does not only help to overcome failure situations fastly, but inserts additional spare capacities which open elbowrooms for lightpath reconfigurations.
In case of upgrade planning, another issue concerns the mobility of preinstalled equipment. While fibers are obviously bound to their place, all other devices are theoretically movable from one location to another. Router interface cards or regenerators can often be easily unplugged and reused on another location, whereas more complex hardware such as WDM systems or switches cause more reallocation effort due to necessary recalibrations. However, network operators usually disregard (expensive) transportations of damageable hardware that so far has worked satisfactory and favor its further use on the spot, also promoted by the fact that the hardware is often easily expandable to a higher stage at low cost.

Traffic engineering. Given a proper hardware installation, establishing optical connections is a purely operational task since switches are software-controlled and thus reconfigurable. The task to determine a new circuit pattern on a fixed physical network is called *traffic engineering*. Note that this includes a possible *lightpath reconfiguration* in multi-hop transparent networks. We distinguish two basic types of traffic engineering. In the *static* case, the circuit configuration is redesigned for the entire connection demand given as input. Such a circuit (and lightpath) reconfiguration is carried out in regular intervals to adapt the network to changing traffic requirements. Between these major restructurings, existing connections might become useless and new connections are requested over time. Without replanning the whole circuit configuration, removing and adding of single or few connections is

done by *dynamic* traffic engineering.

Traffic forecasts. Since future traffic obviously cannot be foreseen, the specified demands are usually based on a *traffic forecast*. Consequently, the service capability of a proposed network also depends on the quality of the forecasts as foundation for the planning. As major influence, this should be kept in mind when a revision of the planning is deduced from the practical suitability of its outcome. In practice, the provision of reliably good forecasts may be very difficult. Providers usually rest on extrapolation of historical traffic patterns and are regularly fooled by unexpected changes. In particular, so-called killer applications with unforeseen popularity can suddenly generate immense additional traffic and thereby cause network overflows, which is naturally unavoidable.

Objective. Designing an optical network will always head for an *objective*, depending on the beholder and its interests. Network users avail the facilities and thus focus on the offered services, their quality and price. Network providers supply the infrastructure and have a stake in the investments and their return. In between, network operators try to find the best way of using the available infrastructure to realize the most efficient and asked services. Each perspective yields a different network design goal: Users aim at best application support, providers at lowest cost, and operators at best benefit from a given infrastructure. The common point of these perspectives is to find the best trade-off between service qualities and their prices. However, it is an option to decide which issues are fixed as requirements and which remain variable to be optimized. A possibility is to define an investment budget for network extensions by which a maximum throughput (based on a given pattern) shall be obtained. The most common objective indeed is to minimize the equipment costs for the realization of some given traffic requirements.

1.5.3 Preliminary specifications

The previous parts of the introduction have pointed up many alternative faces of technology, operation, and planning of optical networks. Although each issue allowing for choices corresponds to a fair question for network design, some decisions are usually determined prior to the planning, and some minor aspects are disregarded for the sake of tractability. So, this concluding part is designated to give a brief overview of the most important preliminary specifications and definitive settings for those network design problems that are to be studied in this thesis.

Basically, the design of optical networks consists in the task to determine a network configuration that serves the traffic demand. To concretize this task, some fundamental preliminaries are specified as follows:

- **Network layer:** The scope of the configuration task lies on the optical layer as transport network.
- **Compatibility:** All considered hardware devices are system-compatible, i.e., we expect all available devices to work together, except for prohibited combinations of fibers and WDM systems. We do not account for functionality-independent incompatibilities, for instance caused by different device manufacturers (which should be appropriately preselected).

- **Physical topology:** The physical topology of the network is proposed as fixed input, including new nodes or links to consider with a setup cost. Adding further nodes or links is hence disregarded within the optimization.
- **Bitrate:** All traffic connections in the network operate on the same bitrate, which allows to express both demands and routings in terms of a number of optical connections or lightpaths, respectively.
- **Demand:** The traffic demand is static, deterministic, and (entirely) given as input. Dynamic demands can theoretically be simulated to some extent as subsequent lightpath reconfigurations on a fixed hardware configuration, but this approach does not include all aspects of online studies. The related issues of call admission and dynamic provisioning of connections are studied in Poensgen [141], for instance.
- **Survivability:** Generally, an optical network is to be protected against all single link or node failures. To incorporate allocated as well as best-effort traffic, we allow to designate a protection level for each connection individually. Unprotected connections can be dropped in case of a failure.

Some aspects allow for optional alternatives which can usually be integrated by slight changes to the models. To avoid huddling plenty of variations together in an exhaustive presentation, we pick some default settings as follows:

- **Existing network:** Basically, we consider the more general case of upgrade planning, with the possibility to specify preinstalled equipment that can be reused on the spot. Greenfield planning occurs as special case with no preinstalled hardware at all.
- **Directions:** We assume capacities, traffic demands, and optical connections to be bidirectional by default. As a consequence, we represent both the physical and the virtual topology by undirected multi-graphs.
- **Routing:** Each demand can be routed bifurcated, i.e., multiple connections between the same nodes need not use the same paths in the physical topology.
- **Objective:** The design goal is to minimize the total network cost for new equipment to install.

Variations of these prerequisites will be indicated explicitly and exemplary discussed as excursions.

Next, fundamental guidelines for the network design are prescribed by the used architecture and the traffic management. We distinguish three scenarios:

- opaque networks, exclusively applying electronic switching such that each lightpath spans a single link only,
- transparent networks with single-hop traffic, where each optical connection is realized by a single lightpath, and
- transparent networks with multi-hop traffic, where end-to-end circuits can decompose into multiple subsequent lightpaths.

Designing a network comprises both the hardware dimensioning as well as the lightpath configuration, including wavelength assignment in the transparent cases. Traffic engineering is not considered as a separate task, but is contained as a subproblem (with a completely fixed hardware configuration).

By these specifications, we obtain a family of specific optical network design problems designated as practically relevant by our industrial cooperation partners Telekom Austria and T-Systems International. To support the planners, we develop suitable solution methods for these problems by means of mathematical optimization, providing both good designs as well as a quality guarantee for their appraisement.

Chapter 2

Modeling optical network design

A mathematical optimization approach to the design of optical networks requires to transform the real-world problem description into the mathematical 'world'. The goal is to find appropriate models that fit to the job. Abstraction plays a key role in this process: Mathematics is best if applied on some well-defined objects with specified rules for their handling. To meet this perspective, an analysis of all related aspects has to be carried out in order to filter out issues of minor importance or influence, leaving the focus on the problem essence for further investigation. The preceding introduction has already prepared such a transformation by specification of the major issues.

In this chapter, we investigate the mathematical modeling of optical network design. The derived models form the base for solution methods. A comparison to particular related models in the literature is postponed to the following chapter, where we discuss these approaches in combination with algorithmic techniques. For a general introduction into modeling of network design, we refer to the excellent books of Pióro and Medhi [140], also providing a comprehensive collection of formulations for various specific settings, and of Grover [61] with emphasis on survivability models.

We begin with a description of our modeling framework as common base for all considered optical network architectures. Such a 'premodeling' step is not a necessary burden, but proves helpful to identify those issues that reflect the main properties and interdependencies subject to the pursued objective.
Next, we formulate integer linear programming models for optical network design. We presented a first simplified model for transparent optical networks in Zymolka et al. [178]. Here, we give more detailed formulations for various network architectures. The exemplary adaptions for alternative or additional aspects reveal the flexibility of our framework in combination with our models.
Finally, we integrate survivability into the models. We have developed a new concept particularly designed for meshed optical networks, balancing spare capacity savings and ease of management. A basic variant was introduced and evaluated in Koster et al. [98, 97], followed by an enhanced version in Wessäly et al. [168] and Gruber et al. [62].

2.1 Modeling framework

In this section, we lay the ground for transforming real-world optical network design into parameterized mathematical models suited for optimization methods. Following the task structuring which is illustrated in Figure 1.15 on page 32, we present a basic view on the network configuration abstracting from minor details and identifying an adequate set of concise objects relevant for the network design. We first propose a device-oriented hardware model which is appropriately simple for mathematical compilation and, at the same time, flexible enough to incorporate many planning issues. Next, we consider the capacities which are on the one side supplied by the installed hardware and on the other side consumed by the connections to establish. Then we investigate the routing, describing the main properties of (lightpath) connections and their respective modeling. Completing the problem transformation, we finally turn to the objective and discuss hardware cost models. Altogether, this view provides the framework for our modeling of optical network design in the subsequent sections.

2.1.1 Hardware

In network design, modeling the physical hardware is typically a difficult matter. On the one hand, planners prefer a highly detailed model that incorporates as many aspects as possible in order to get the most realistic figure of the network. On the other hand, any simplification makes planning tasks more tractable, and the special power of mathematical approaches often bases on the abstraction from subordinate details, getting a problem more handy for solution algorithms.
Our hardware model tries to satisfy both needs in the following trade-off. Guided by the practical perspective, we apply a modular description for optical networks as composition of many single devices, but restrict on few abstract sorts of hardware *modules* to reduce the detail level: fibers, WDM systems, switches, converters, and regenerators. These module sorts subsume all devices either by abstracting the variant (e.g., DXC and OXC as switch) or particular combinations (e.g., optical fiber together with amplifiers). Transmitters and receivers are not explicitly incorporated, because they only occur at the beginning and the end of the circuits to establish for carrying traffic, hence their number is fixed for a static traffic requirement.
Any module sort has a couple of attributes representing those characteristic properties that are relevant for the planning. Modules of the same sort may vary in the attribute values, representing different module types for that sort. The auxiliary use of artificial module types allows for easy integration of some aspects that otherwise require cumbersome modeling. Some illustrative applications are exemplified below.

Module sorts. In the following, we give a short description of the five module sorts involved in our hardware configuration model and their main attributes:

Fibers. Fibers represent the long-distance carrier for optical transmissions in the network and are installed on physical links. As module sort, a *fiber* combines an optical fiber with the corresponding signal refreshment equipment, i.e., a fiber is ready-to-use for transmissions between the connected nodes.

In general, each link can contain multiple fibers of any type which can additionally carry WDM systems. The fiber types differ in their signal propagation qualities which influence the signal degradation during the transmission and the applicability of WDM systems.

WDM systems. WDM allows for the efficient use of fibers by enabling simultaneous transmissions on different wavelengths. The module sort *WDM system* combines multiplexer and demultiplexer units which perform the merging of single wavelength channels to a combined signal for transmission and their separation afterwards. Although each of these units is physically located at a network node, WDM systems represent transmission equipment and are, as an integrated unit, installed on fibers (and hence on the physical links as well).
Each WDM system type is characterized by the supported spectrum, i.e., the number of optical channels and their particular wavelengths that can be combined. Its application may premise particular fiber properties. Modern WDM systems can combine many optical channels with very dense wavelength division (DWDM). During transmission on a fiber, the signals must not deviate too much for a correct demultiplexing. To model these requirements, the possible fiber types are specified for each WDM system type.
Moreover, fibers can also be operated without applying WDM. In this case, the transmission consists of a single optical channel (of any wavelength). However, it simplifies the modeling if we avoid to explicitly distinguish whether or not WDM is applied on a fiber. For this, we could introduce an auxiliary WDM system operating a single optical channel (without de-/multiplexing) which represents the 'no WDM' case. This allows to assume that any installed fiber carries exactly one WDM system.

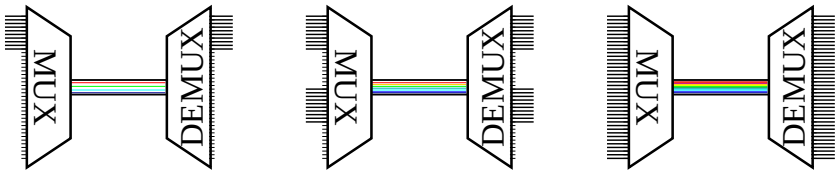
Switches. Switching of optical channels can be performed by several hardware devices, such as DXCs, OXCs, or their variants, as described in Section 1.1.3. The module sort *switch* indeed abstracts from the particular technology and models just the common functionality: Each module provides a number of ports for switching optical channels. If using a DXC, the switch includes also the interface devices for the required o-e and e-o conversions, i.e., two couples of a transmitter and a receiver for each port to enable switching of a bidirectional optical connection. Clearly, switches are always installed in the nodes of the physical topology, and each node can contain multiple modules.
We remind that switching an optical channel is independent of its wavelength, and all switches are fully configurable, i.e., any (input) port can be mapped on any (output) port. So, different switch types vary only in the number of offered ports (and their price).

Converters. Wavelength conversion is only applied in the nodes of the physical topology. By a conceptual interpretation, we do not (explicitly) distinguish whether the conversion is performed optically or by a transponder, i.e., as an o-e-o conversion. Abstracting from the applied technique, we assume that the module sort *converter* operates a single optical channel without range limitation, i.e., any wavelength can be converted into any other. Hence, converter types differ only in their prices.

Regenerators. Regeneration does not refer to amplification, but to full (or 3R) signal refreshment and includes a possible wavelength conversion. As module sort, a *regenerator* can handle a single optical channel of any wavelength. So, also regenerator types differ only in their price.
Typically, regenerators are placed at the network nodes. An exception occurs in networks with links that are too long for direct transmission. A typical example are wide-area networks covering large geographical regions, such as national networks in the USA. Such links can only be operated by intermediate application of regenerators. An explicit integration of on-link regenerations can simply be avoided by insertion of auxiliary nodes at the regeneration points. To ease the modeling, we postpone the description of regeneration points and their consistent integration to further aspects discussed in Section 2.2.5 and assume in the following that all network links allow for direct transmission.

Additionally, all sorts of modules have some common attributes pertaining to installation capabilities. Commercial availability and installation space restrictions can impose an upper bound on both the number of each module type and the total number of modules of a sort at each location. Such bounds are also obligatory for upgrade planning to express that a limited number of modules is initially given for free reuse on the spot.
The interpretation of the physical network as a module construction kit provides a sufficiently accurate network figure for a realistic cost determination, in particular for upgrade planning. Moreover, such a unitized hardware description offers additional modeling flexibility: The auxiliary use of module types is a helpful tool to integrate further planning issues, as already demonstrated for fiber operation without WDM. To provide just one further example for the many possibilities, we describe the integration of upgradeable WDM systems into the framework.

Example: WDM system upgrades. Manufacturers sometimes offer basic devices with a restricted functionality which can be upgraded without exchanging the entire unit. So, WDM systems are built of a rack that contains a de-/multiplexer unit and some slots for fiber interfaces by which the wavelength channels to combine are inducted. Not all slots need to be used initially. For instance, a WDM system can have 40 slots but uses only 10 of them in the basic version. Upgrading the system is then possible by adding further interfaces, see Figure 2.1. This way, some of the already installed hardware can further be used.
An integration of such WDM system upgrades into our model has to ensure that an upgrade system can only be installed on top of an appropriate basic system (without further use of the latter on another fiber). For this, we introduce an auxiliary fiber



(a) *Base WDM system (10 wavelengths).* (b) *WDM system upgrade (20 wavelengths).* (c) *Full WDM system (40 wavelengths).*

Figure 2.1: *Example for WDM system upgrades.*

type which represents those fibers that already carry an extendible WDM base system. Both the WDM base system and its upgrade system are only installable on this auxiliary fiber type whose number on a link is limited to the number of preinstalled base system combinations on that link. Moreover, the WDM upgrade system gets the parameters of the upgraded system. Whenever a WDM upgrade system is selected for installation, the corresponding base system cannot be further used due to the fiber type number limitation and the property that each fiber carries exactly one WDM system. This way, a proper accounting of the supplied capacities is guaranteed.

2.1.2 Capacities

The installed hardware as physical part of an optical network offers the required functionalities to establish connections. These functionalities are quantified in terms of capacities to express the amount of traffic a module can handle. According to different functionalities, there are several types of capacity for transmission, switching, and transformation of connections.

Transmission capacity on the physical links is provided by fibers and WDM systems. While the fiber serves as a signal carrier, the applied WDM system determines the number of usable optical channels and their operation wavelengths. Hence, transmission capacity is expressed in numbers of wavelength channels. The individual capacities of multiple WDM systems (and fibers) on a link sum up to the total link capacity.

Switching capacity at the network nodes is supplied by switches. The modules offer a number of ports, each being able to handle a single optical channel independent of its wavelength. Therefore, switching capacity can be expressed as the number of (bidirectional) ports or, by the one-to-one correspondence, also as the number of optical connections to dispatch. In case of multiple switches at the same node, their capacities accumulate to the total node switching capacity.

Switching capacity at the nodes can be dimensioned either traffic-oriented or link-oriented, as introduced in Section 1.5. Since the latter yields more flexible network designs which can simpler be adapted to changing traffic patterns, we focus on link-orientation as default and postpone model adaptions for traffic-oriented node dimensioning to an excursion at the end.

Conversion capacity, as well provided by the nodes, describes the possibility to exchange the wavelength of operation for lightpaths. For this, wavelength converters have to be installed. Each converter handles a single optical channel and is able to convert any wavelength to any other. Hence, the number of wavelength converters at a node directly determines the conversion capacity in terms of manipulatable optical channels.

Regeneration capacity is similar to conversion capacity, since a regenerator is also applied to a single optical channel independently of its wavelength. Hence, the regeneration capacity of a node in terms of the number of regenerable lightpaths corresponds to the number of installed regenerators.

Note that all capacities are counted in (equivalents of) entire units of optical/wavelength channels, where the more specific distinction of wavelength channels occurs only on the links. This observation allows to simplify some further explanations: Whenever we use the term *capacity* without further specification, we refer to a number of optical channels, while *wavelength capacity* reflects a number of wavelength channels of a particular wavelength. Clearly, the sum of the wavelength capacities over all wavelengths on a link equals its capacity.

2.1.3 Connections

Connections requested in the network are established as end-to-end circuits which correspond to paths in the physical topology. Basically, each circuit consumes a wavelength channel on each traversed link and a switching port in each traversed node, including its endnodes. The use of different technologies indeed makes it necessary to refine this simplified connection model according to the specific circuit types.

Opaque networks. In opaque networks, circuits represent connections which use optics only as a point-to-point carrier. Since each transmission is o-e-o-converted in each node, neither the wavelengths of the used link channels nor a limited overall length are to be taken into account. So, establishing connections reduces to routing paths such that the capacities at links and nodes are not exceeded.

Transparent networks. In contrast, transparency in network nodes brings up additional features of connections. Here, the circuits are composed of lightpaths which are subject to wavelength continuity unless wavelength converters are applied. In addition, the length limitation for lightpaths carries over to circuits unless regenerators are used for establishing multi-hop connections.
Since a lightpath must specify the wavelength of operation on each link, the routing has to be carried out with respect to the wavelength capacities in the physical topology, and the task is accompanied by the determination of a conflict-free wavelength assignment. Moreover, each exchange of the wavelength along a lightpath occupies a wavelength converter at the corresponding node, and thus consumes conversion capacity.
In case of multi-hop connections, the concatenation of lightpaths along a circuit needs a regenerator whenever a lightpath is terminated and a new one is set up. Hence, multi-hop connections also consume regeneration capacity in the corresponding nodes. Allocating these nodes on a multi-hop connection requires an accurate measurement of lightpath lengths.

Lightpath length measurement. Concerning length limitations, any physical equipment can in principle influence the signal quality in several regards and thus decrease the maximum transmission distance. Although theoretically possible, an accurate compilation of all these interdependent effects and integration of the resulting parameters into the models yields a significant increase in size and complexity. Instead, we apply a simplified length measure mechanism typically used by the network planners. In practice, it is customary to limit connection lengths always in terms of fiber kilometers. The link lengths then do not refer anymore to

physical fiber lengths, but are set in order to reflect the extent of signal degradation along lightpaths, for instance adjusted by relation factors for different fiber types. Moreover, node disturbances can be integrated by adding an appropriate value to the adjacent link's lengths. This method is sufficiently flexible to incorporate many aspects related to length-restricted transmissions, while keeping the measurement easy.
For using such a measurement, the physical topology has to be slightly adapted. To take different length restrictions for various equipment combinations into account, we can replace a physical link by multiple parallel links each with specific length representing the transmission properties of the associated installable equipment. This way, the path of any connection in the refined topology specifies also the traversed equipment and thus allows for an accurate length measurement corresponding to the suffered signal degradation.
Moreover, the length restriction model can also be extended to take multiple competing limitations into account, e.g., to distinguish signal quality loss by attenuation or by dispersion. In this case, each degradation measure defines a separate length value for each link, and the topology modifications have to be carried out w.r.t. each of these measures in a straightforward way. However, although our modeling offers this possibility, we remark that the network design task complexity grows substantially with any extension of the physical topology, and thus the details to incorporate should be chosen carefully to avoid model overloading. To limit notation overhead, we restrict ourselves to a single length measure per fiber in the sequel.

2.1.4 Cost

From a provider-oriented perspective, a main goal of designing an optical network consists in minimizing the total network cost. There are various cost sources, such as capital expenditures (so-called CAPEX) subsuming building costs, e.g., for digging conduits for the fibers or preparing rooms for the node equipment, hardware costs for buying and installing new modules, or operating expenses (so-called OPEX), e.g., for maintenance and configuration services, among others. The planning indeed can only take those costs into account which depend on the decisions to make and are either already known or can be forecasted (at least approximately).

Cost model. We use a cost model that is as flexible as the component model for the hardware. In fact, we associate all costs with network and hardware components, each with an arbitrary cost function. So, it is up to the planner which costs are to be involved. For instance, interpreting the design as a decision about which hardware to buy, it is reasonable to take only current market prices for the components into account. Alternatively, the provider could as well choose to disregard purchase costs which are paid only once, and to focus rather on regularly incurred costs for network operating and maintenance. In practice, however, planning a network incorporates as many information as available, which is typically a mix of the examples above, e.g., by combining depreciations and operational costs.
Our cost model distinguishes two main types of costs: those associated with network components, and those associated with hardware components. The first category refers to setting up nodes and links in the physical topology. Since the topology

is fixed for the design, new links and nodes have to be inserted priorly, but their usage can be bound to an incurred setup cost. The cost for providing capacities in the network is modeled by the hardware module costs. Each module unit has a total cost for its use. In general, the cost value can depend on the installation location. Fiber costs, for instance, are obviously varying with the length of a link to be installed on, while node equipment typically incurs the same cost at any node.
In cooperation with our industrial partners, we have selected the following cost functions which reflect network building costs and are composed as follows:

Fiber cost. A fiber module represents a combination of an optical fiber together with the required amplifiers, placed at regular distances, subdividing the fiber into segments. The segment length is limited by the maximum transmission distance without amplification. Hence, the total cost of a fiber type is the sum of three parts: a cost per physical link kilometer for the optical fiber itself, a cost per each full segment for the amplifiers, and a fixed installation cost (independent of the link length).

WDM system cost. Although installed on links, WDM systems represent hardware that is located at the ends of links and thus have no link-length dependent cost. Instead, different module types vary in their capacities. Thus, we combine the WDM system type cost of a base cost and a cost per wavelength interface. We always consider fully equipped systems, i.e., interface costs are not usage-dependent.

Switch cost. Similar to WDM systems, switch costs are typically independent of the installation location, but may vary with the supported capacity. So, a switch type cost is composed of a type base cost and an again usage-independent cost per port.

Converter and regenerator cost. Since all types of wavelength converters and regenerators offer the same functionality, no parameter-dependent costs occur, and we define only a unit cost for any of these modules.

Clearly, all of the considered cost factors can also vary for different module types and installation locations. However, we assume the cost functions to be evaluated a priori, yielding a single value for the total cost of each module. Already existing equipment is always assumed to be available for free. Hence, the network design objective only sums up the defined total cost for any newly installed module.

The choice of suitable cost models completes the modeling framework. At this point, we have identified all relevant objects and properties that have to be taken into account in the design of optical networks. So, we are now ready to formalize the task as a mathematical optimization problem.

2.2 Mathematical models

The preceding framework provides the basis for the formulation of integer linear programming models for optical network design. We begin this section with a lucid

introduction of the required formal definitions and notations. The models are then presented separately for different network scenarios, ordered by increasing complexity of the planning task. Finally, we discuss some selected formulation alternatives and model adaptions to those options that differ from our chosen defaults or reflect so far non-integrated aspects. Throughout this section, we leave out survivability which is investigated separately afterwards.

2.2.1 Parameters and notation

The basic structure of an optical transport network is the physical topology which maps the geographical layout. It contains all physical locations for hardware placement as nodes and their point-to-point connections as physical links, together typically represented by a multi-graph. By default, we consider bidirectional capacities, demands, and circuits, which allows to rest on undirected multi-graphs, cf. Section 1.2.4.

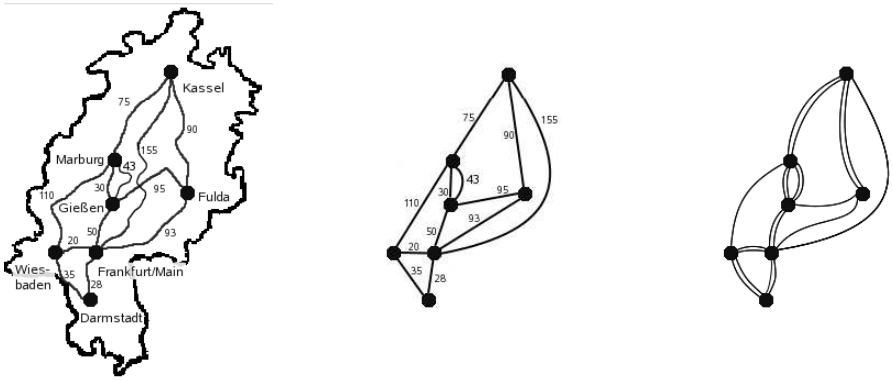
Definition 2.1 (Physical topology) *The **physical topology** of an optical network consists of a finite set N of **nodes** and a finite multi-set T of **physical links** represented by an undirected multi-graph $\mathcal{T} = (N, T)$. To become usable, each link $t \in T$ has a link setup cost $C_t \in \mathbb{R}_+$, and similarly each node $n \in N$ defines a node setup cost $C_n \in \mathbb{R}_+$.*

If the physical topology of an optical network is disconnected, the multi-graph components are independent of each other and can be considered individually. Therefore we can w.l.o.g. assume the physical topology $\mathcal{T}$ to be connected.

As physical topology, the multi-graph $\mathcal{T}$ does not distinguish fibers with different signal propagation qualities as long as these fibers are commonly installed. Parallel physical links occur only in case two nodes are connected by separately conducted connections (which is of importance for survivability investigations, cf. Section 2.3). To explicitly account for varying length limitations of the fibers, the physical topology is transformed as described in Section 2.1.3 into the so-called *supply network* by subdivision of the physical links such that a separate supply link serves for the installation of all fiber types with the same transmission properties. This way, each supply link can get an individual length for the transmission distance measurement. Figure 2.2 shows an example for a real-world network, its physical topology, and the associated supply network.

Definition 2.2 (Supply network) *The undirected **supply network** $\mathcal{N} = (N, L)$ is defined as the multi-graph obtained from the physical topology $\mathcal{T} = (N, T)$ by replacing each physical link $t \in T$ with finitely many parallel **supply links** L_t representing different equipment properties, where $L = \bigcup_{t \in T} L_t$. Each link $\ell \in L$ has a length $\omega_\ell \in \mathbb{R}_+$ assigned to it (expressed in virtual kilometers (km)). With respect to these lengths, the maximum total distance for optical transmissions is the **optical reach** $\Omega \in \mathbb{R}_+ \setminus \{0\}$.*

Both the physical topology and the supply network build the foundation for the installation of modules and the establishment of (lightpath) connections. Another



(a) *Real-world network.* (b) *Physical topology.* (c) *Supply network.*

Figure 2.2: *Example of a small Hessian network with (a) its real-world shape, (b) the physical topology, and (c) the supply network for two property-different fiber types, one of which is installable only on links of not more than 90km. Notice that Marburg and Gießen are connected by two parallel, but independent physical links.*

such fundamental resource is the set of all operable wavelengths that can be provided by WDM systems.

Definition 2.3 (Wavelength spectrum) *The finite set Λ denotes the **wavelength spectrum** of all wavelengths that can be operated in the optical network.*

Next, we turn to the hardware configuration of the network, composed of the modules which are introduced sort-wise. We begin with the link technology, where fibers serve as a basic transmission carrier.

Definition 2.4 (Fibers) *The finite set $\mathcal{F}$ represents all available **fiber types**. For each supply link $\ell \in L$, the subset $\mathcal{F}_\ell \subset \mathcal{F}$ contains all fiber types that are installable on link ℓ. For each supply link $\ell \in L$, each installable fiber type $f \in \mathcal{F}_\ell$ has as attributes*

- *a cost $C_\ell^f \in \mathbb{R}_+$ for new installation,*
- *the number $e_\ell^f \in \mathbb{Z}_+$ of already existing fibers, and*
- *the maximum number (bound) $b_\ell^f \in \mathbb{N}$ of units allowed to use.*

Moreover, the maximum number of fibers installable on supply link $\ell \in L$ is denoted by $b_\ell^{\mathcal{F}} \in \mathbb{Z}_+$, whereas $b_t^{\mathcal{F}} \in \mathbb{Z}_+$ limits the total number of fibers on a physical link $t \in T$ (over all associated supply links in L_t).

WDM systems represent the second constituent of link technology. In contrast to fibers which have to be installed on particular supply links, WDM systems can be compatible with different fibers from several supply links $\ell \in L_t$. Hence, we allocate WDM systems on the level of physical links.

Definition 2.5 (WDM systems) *The finite set $\mathcal{W}$ subsumes all available **WDM system types**. As attributes, each WDM system type $w \in \mathcal{W}$ defines*

- *the set $\mathcal{F}_w \subset \mathcal{F}$ of fiber types on which it can be installed,*
- *the set $\Lambda_w \subset \Lambda$ of supported wavelengths*

and thus provides $|\Lambda_w|$ optical channels. Furthermore, each WDM system type $w \in \mathcal{W}$ specifies for each physical link $t \in T$

- *the cost $C_t^w \in \mathbb{R}_+$ for being newly installed,*
- *the number $e_t^w \in \mathbb{Z}_+$ of existing units, as well as*
- *the maximum number (bound) $b_t^w \in \mathbb{N}$ of such modules allowed to use.*

For notational simplifications, we introduce for each WDM system type $w \in \mathcal{W}$ the additional parameters

- *$k^{w\lambda} \in \{0,1\}$ as indicator whether wavelength $\lambda \in \Lambda$ is provided ($k^{w\lambda} = 1$) or not, and*
- *$k^w = |\Lambda_w| = \sum_{\lambda\in\Lambda} k^{w\lambda}$ for the optical channel capacity.*

Since the total number of WDM systems on a physical link is implicitly limited by the total number of (compatible) fibers, we do not define an additional bound.

Considering the node technology, we first introduce the switches.

Definition 2.6 (Switches) *The finite set $\mathcal{S}$ represents all available **switch types**. As attributes, each switch type $s \in \mathcal{S}$ defines the number $k^s \in \mathbb{N}$ of offered switching ports and further for each node $n \in N$*

- *the installation cost $C_n^s \in \mathbb{R}_+$,*
- *the number $e_n^s \in \mathbb{Z}_+$ of already existing modules, and*
- *the maximum number $b_n^s \in \mathbb{N}$ of usable units of that type.*

In addition, $b_n^{\mathcal{S}} \in \mathbb{N}$ limits the total number of all switches at each node $n \in N$.

Further node equipment is built of wavelength converters and regenerators, two module sorts which we differ conceptually and which we therefore introduce separately.

Definition 2.7 (Converters) *The finite set $\mathcal{C}$ defines all **(wavelength) converter types**. The attributes for each converter type $c \in \mathcal{C}$ define for any node $n \in N$*

- *the installation cost $C_n^c \in \mathbb{R}_+$,*
- *the number $e_n^c \in \mathbb{Z}_+$ of already existing modules, and*
- *the maximum number (bound) $b_n^c \in \mathbb{N}$ of such modules.*

Moreover, $b_n^{\mathcal{C}} \in \mathbb{N}$ limits the total number of all converters allowed to use at node $n \in N$.

We remark that the total number bound for converters is only introduced for a unified view on all module sorts and for the sake of completeness. In practice, operators typically use a cheapest device as often as required throughout the network. In such a case, the bound on the total number of converters coincides with the number bound on the only type. The same observation holds for the regenerators.

Definition 2.8 (Regenerators) *The finite set $\mathcal{R}$ defines all **regenerator types**. For each node $n \in N$, each regenerator type $r \in \mathcal{R}$ has as attributes*

- *the installation cost $C_n^r \in \mathbb{R}_+$,*
- *the number $e_n^r \in \mathbb{Z}_+$ of already existing modules, and*

- *the maximum number (bound)* $b_n^r \in \mathbb{N}$ *of such modules.*

The number $b_n^{\mathcal{R}} \in \mathbb{N}$ *limits the total number of all regenerators usable at node* $n \in N$*.*

Having defined all parameters for the hardware configuration, we switch over to the lightpath configuration. As a basic term, we first introduce the routing path of optical transmissions.

Definition 2.9 (Routing path) *A* ***routing path*** *is an alternating sequence of nodes and links in the supply network* $\mathcal{N}$ *defined by* $p = (n_0, \ell_1, n_1, \ell_2, n_2, \ldots, \ell_h, n_h)$*, where* $n_i \in N$ *for all* $i = 0, \ldots, h$*, and* $\ell_i \in L$ *with* $\ell_i = n_{i-1}n_i$ *for all* $i = 1, \ldots, h$*. We call* $o_p = n_0$ *the* ***origin****,* $d_p = n_h$ *the* ***destination****, and the value* h *the* ***(physical) hop-length*** *of the routing path. It depends on the context whether* p *is interpreted as directed or undirected path, the latter making origin and destination exchangeable. W.l.o.g., all routing paths correspond to simple, non-closed paths, i.e.,* $n_i \neq n_j$ *for all* $0 \leq i < j \leq h$*, and thus define uniquely determined sets of traversed links and nodes. For a routing path* p*, we denote by* $L(p) = \{\ell_1, \ldots, \ell_h\} \subset L$ *the set of all supply links, by* $N(p) = \{n_1, \ldots n_{h-1}\}$ *the set of inner nodes, and by* $N[p] = \{n_0, \ldots, n_h\} \subset N$ *the set of all nodes in the path.*

In optical networks, lightpaths form the basic connection type. A lightpath is characterized by its routing path together with an assignment of an operation wavelength on each traversed link.

Definition 2.10 (Lightpath) *A routing path* p *together with an assignment* $\alpha_p : L(p) \to \Lambda$ *of an operation wavelength* $\alpha_p(\ell)$ *to each link* $\ell \in L_p$ *of* p *defines a* ***lightpath*** (p, α_p)*.*

Whenever the associated wavelength assignment does not matter, we refer to a lightpath also by its routing path p, i.e., without specified wavelengths.
Lightpaths then compose to optical connections, which are the end-to-end circuits in optical networks.

Definition 2.11 (Optical connection) *An* ***optical connection*** *is a sequence* $(p_1, \alpha_{p_1}), \ldots, (p_k, \alpha_{p_k}$ *of lightpaths such that the destination of* p_i *is the origin of* p_{i+1} *for all* $i = 1, \ldots, k-1$*. We call* k *the* ***virtual hop-length*** *of the optical connection. For* $k = 1$*, the optical connection is called* ***single-hop****, otherwise* ***multi-hop****.*
The routing path of an optical connection composes of the concatenated routing paths of the corresponding lightpaths. W.l.o.g., we assume all optical connection routing paths to be as well simple, non-closed paths.

We finish with the formal statement of the connection requirements.

Definition 2.12 (Commodities) *The (multi-) set* $\mathcal{Q}$ *subsumes all* ***commodities****, also called* ***demands****, which represent the connection requirements. Each commodity* $q \in \mathcal{Q}$ *is specified by*

- *the endnodes to connect, split in an* ***origin*** $o_q \in N$ *and a* ***destination*** $d_q \in N$*,* $o_q \neq d_q$*, and by*
- *the* ***demand value*** $v_q \in \mathbb{N}$ *as number of optical connections to establish between these nodes.*

As we consider undirected commodities, the direction indicated by origin and destination is selected arbitrarily for the sake of modeling. Moreover, multiple commodities between the same pair of origin and destination are allowed.

2.2.2 Opaque networks

As characteristic feature, opaque optical networks perform electronic switching in the nodes where transmissions always undergo an o-e-o conversion. So, the optical medium serves just as point-to-point carrier between fiber-connected nodes, and all lightpaths span only a single link. This implies two consequences: First, a wavelength of operation can be chosen independently for any traversed link of each optical connection, and second, the total length of optical connections is not limited by the maximum optical transmission distance. In particular, the first observation indicates that a feasible wavelength assignment can be simply determined for any routing as long as the link capacities in terms of total channel numbers are not exceeded by the numbers of transmitted connections. So, the wavelength assignment can be released from the core task and settled in a postprocessing step. With this exclusion, we can abstract from specific wavelengths and take only the number of optical channels into account.

Regarding the unlimited circuit length, we remind on the assumption that each physical link allows for optical transmissions (on at least one fiber type). Due to the opacity of nodes, link lengths then do not matter anymore, and splitting links to reflect different fiber propagation properties is unnecessary. Hence, the supply network is equivalent to the physical topology, i.e., $T = L$ and $L_t = \{ \ell \}$ for all $t \in T$. Nevertheless, we formulate the constraints already in the general case, i.e., on the supply network, in order to allow reuse for other architectures.

Capacity variables. The hardware configuration and the lightpath configuration are coupled by capacities as central term. By the use of auxiliary variables for the capacities, we can separately model the hardware dimensioning and the connection routing. So, we introduce the following capacity variables:

$y_\ell \in \mathbb{Z}_+$	denotes the number of channels provided on link $\ell \in L$;
$y_n \in \mathbb{Z}_+$	denotes the number of ports provided at node $n \in N$.

The codomain of these variables is a discrete set of values that correspond to capacities supplied by feasible hardware installations. Although the variables can be easily eliminated throughout all models, an explicit notation of capacities proves to be helpful in formulating and solving the problem.

Location setup. As setup costs are involved, it is also important to allow for not setting up a location, even though there might be existing equipment. For this, we can consult the capacity variables, indicating utilization of links and nodes by positive values. Since it is only important to distinguish between zero and non-zero capacities, we use the following binary variables on the physical topology:

$x_n \in \{0, 1\}$	indicates whether node $n \in N$ is set up ($x_n = 1$) or not, and
$x_t \in \{0, 1\}$	indicates whether physical link $t \in T$ is set up ($x_t = 1$) or not.

With M sufficiently large (e.g., the total demand sum plus the largest capacity of a switch or WDM system, respectively), the correct setting of these variables is enforced by:

$$y_n \leq M x_n \qquad \forall\, n \in N \tag{2.1a}$$

$$\sum_{\ell \in L_t} y_\ell \leq M x_t \qquad \forall\, t \in T \tag{2.1b}$$

$$x_n, x_t \in \{0,1\} \qquad \forall\, n \in N,\ t \in T \tag{2.1c}$$

$$y_n, y_\ell \in \mathbb{Z}_+ \qquad \forall\, n \in N,\ \ell \in L \tag{2.1d}$$

The node capacities directly determine the node setup in constraints (2.1a), since supply nodes and physical nodes coincide. For setting up physical links, constraints (2.1b) state the corresponding conditions for any number of associated supply links. Finally, binary indicators and integer capacities are guaranteed by constraints (2.1c) and (2.1d).

Hardware configuration model. Let us next consider the hardware configuration. As wavelength converters and regenerators are needless in opaque networks, we deal only with switches at the nodes and combinations of fibers and WDM systems on the links. Taking existing equipment into account, we have to *install* new modules where required, and to assign a fiber to each WDM system. The existing equipment indeed need not necessarily be used further, e.g., when an preinstalled small WDM system is replaced by a new larger one to be carried on the same fiber. Therefore, we have also to decide how many units of each module type are to *use* at each location. So, we introduce the following variables:

$x_n^s \in \mathbb{Z}_+$	denotes the number of *new* switches (DXCs) of type $s \in \mathcal{S}$ to be installed at node $n \in N$;
$z_n^s \in \mathbb{Z}_+$	denotes the total number of switches (DXCs) of type $s \in \mathcal{S}$ to be used at node $n \in N$;
$x_\ell^f \in \mathbb{Z}_+$	denotes the number of *new* fibers of type $f \in \mathcal{F}$ to be installed on link $\ell \in L$;
$x_t^w \in \mathbb{Z}_+$	denotes the number of *new* WDM systems of type $w \in \mathcal{W}$ to be installed on physical link $t \in T$;
$z_\ell^{fw} \in \mathbb{Z}_+$	denotes the total number of combinations composed of a fibers of type $f \in \mathcal{F}$ carrying a WDM system of type $w \in \mathcal{W}$ to be used on link $\ell \in L$.

Observe that the variables x_n^s, x_ℓ^f, and x_t^w just set new installations without defining the particular combinations of modules to use. These combinations are separately defined by the variables z_n^s and z_ℓ^{fw}, the latter also fixing the assignments of fibers and WDM systems for particular supply links. This way, we are able to cover all possible configurations composed of existing and newly installed modules, e.g., further use of a fiber with replaced WDM system, which would require more effort when alternatively using installation variables referring directly to specific equipment combinations. Clearly, the assignment variables z_ℓ^{fw} are only required for feasible combinations of WDM systems $w \in \mathcal{W}$ and associated fibers $f \in \mathcal{F}_w$. For notational convenience, we complete the assignment variable set for all combinations and exclude impossible assignments by fixing $z_\ell^{fw} = 0$ for all $w \in \mathcal{W}, f \notin \mathcal{F}_w$.

Node dimensioning. With the introduced variables, we can formulate hardware dimensioning constraints as part of an integer linear program. The auxiliary capacity variables allow also to address node and link configurations separately. We begin with the nodes for which the hardware dimensioning model reads:

$$\sum_{s\in\mathcal{S}} k^s z_n^s = y_n \qquad \forall\, n \in N \tag{2.2a}$$

$$z_n^s \le e_n^s + x_n^s \qquad \forall\, n \in N,\ s \in \mathcal{S} \tag{2.2b}$$

$$z_n^s \le b_n^s \qquad \forall\, n \in N,\ s \in \mathcal{S} \tag{2.2c}$$

$$\sum_{s\in\mathcal{S}} z_n^s \le b_n^{\mathcal{S}} \qquad \forall\, n \in N \tag{2.2d}$$

$$x_n^s,\ z_n^s \in \mathbb{Z}_+ \qquad \forall\, n \in N,\ s \in \mathcal{S} \tag{2.2e}$$

For each node, constraint (2.2a) accounts for the switching capacity y_n which is accumulated from the used switches. Type-wise, constraints (2.2b) ensure their supply by existing and newly installed modules. The numbers of usable switches for each type and for all modules in total are bounded by constraints (2.2c) and (2.2d), respectively. Finally, conditions (2.2e) care for solution integrality. Note that conversion and regeneration need not be modeled.

Link dimensioning. Next, we turn to the links where the transmission capacities are composed in a more complicated way:

$$\sum_{w\in\mathcal{W}} \sum_{f\in\mathcal{F}_w} k^w z_\ell^{fw} = y_\ell \qquad \forall\, \ell \in L \tag{2.3a}$$

$$\sum_{w\in\mathcal{W}} z_\ell^{fw} \le e_\ell^f + x_\ell^f \qquad \forall\, \ell \in L,\ f \in \mathcal{F}_\ell \tag{2.3b}$$

$$\sum_{\ell\in L_t} \sum_{f\in\mathcal{F}} z_\ell^{fw} \le e_t^w + x_t^w \qquad \forall\, t \in T,\ w \in \mathcal{W} \tag{2.3c}$$

$$\sum_{w\in\mathcal{W}} z_\ell^{fw} \le b_\ell^f \qquad \forall\, \ell \in L,\ f \in \mathcal{F}_\ell \tag{2.3d}$$

$$\sum_{\ell\in L_t} \sum_{f\in\mathcal{F}} z_\ell^{fw} \le b_t^w \qquad \forall\, t \in T,\ w \in \mathcal{W} \tag{2.3e}$$

$$\sum_{w\in\mathcal{W}} \sum_{f\in\mathcal{F}_w} z_\ell^{fw} \le b_\ell^{\mathcal{F}} \qquad \forall\, \ell \in L \tag{2.3f}$$

$$\sum_{\ell\in L_t} \sum_{w\in\mathcal{W}} \sum_{f\in\mathcal{F}_w} z_\ell^{fw} \le b_t^{\mathcal{F}} \qquad \forall\, t \in T \tag{2.3g}$$

$$z_\ell^{fw} = 0 \qquad \forall\, \ell \in L,\ w \in \mathcal{W},\ f \notin \mathcal{F}_w \tag{2.3h}$$

$$x_\ell^f,\ x_t^w,\ z_\ell^{fw} \in \mathbb{Z}_+ \qquad \forall\, \ell \in L,\ f \in \mathcal{F},\ t \in T,\ w \in \mathcal{W} \tag{2.3i}$$

For each supply link $\ell \in L$, the variables z_ℓ^{fw} express the selection of fiber and WDM system pairs to use and, at the same time, their assignments to compatible combinations whose channel numbers sum up to the total transmission capacity, as described by constraint (2.3a). Provision of the selected numbers of fibers and WDM systems is location-wise guaranteed by constraints (2.3b) and (2.3c), respectively, implying

the installation of new units whenever the existing modules do not suffice. Furthermore, the availability of each module type is restricted by the number bounds stated in constraints (2.3e) and (2.3d). Eventually, the total number of installable fibers and thus link systems is limited by constraint (2.3f) for each supply link separately and in constraint (2.3g) for each physical link in total. Prohibition of incompatible combinations of fibers and WDM systems is achieved in constraints (2.3h) by simply zero fixing of the corresponding variables (or, alternatively, removing them from the model). The integrality conditions (2.3i) complete the link configuration model.

Capacity relation. With link-oriented dimensioning of switching capacities, the two presented configuration models are linked as follows. For each selection of feasible link capacities y_ℓ, the capacity requirement at each node is determined by the sum of the capacities of all incident links, i.e.,

$$\sum_{\ell \in L(n)} y_\ell \leq y_n \qquad \forall\, n \in N\,. \tag{2.4}$$

This way, the nodes are dimensioned such that any set of connections not exceeding the link capacities can be switched. In particular, the counting on the left hand side does not care about the fact that a connection bypassing a node uses an optical channel on two incident links, but occupies only a single (bidirectional) port in the intermediate node. Hence, the resulting switching capacity at each node would suffice even if all incident links were fully loaded with connections terminating in the node. As this is rarely the case, the nodes are typically oversized by (2.4). To curtail this effect (at the price of neglecting some extreme traffic changes), a reasonable alternative is to use the current traffic requirements as an indicator for the number of connections that will terminate at each node. In this case, the node dimensioning can be carried out by

$$\sum_{\ell \in L(n)} y_\ell + \sum_{\substack{q \in \mathcal{Q}:\\ n \in \{o_q, d_q\}}} v_q \leq 2y_n \qquad \forall\, n \in N\,. \tag{2.5}$$

Here, each incident link channel, assumed as part of a bypassed connection, is just counted half, expressed equivalently by doubling the right hand side. The second term on the left hand side corrects this counting by the actual number of emanating connections. By (2.5), the node capacities (can) become smaller than from (2.4), but are still sufficient to establish most lightpath reconfigurations according to the link capacities, and so we use (2.5) in this thesis. For the actual design, indeed, both alternatives allow to exclude the node capacities from the routing restrictions. As long as the lightpath connections meet the link capacities, their establishment will not exceed the switching capacities as well.
The combination of all constraints resulting from the configuration models (2.2) and (2.3) for all nodes and links, respectively, together with the linking constraints (2.5) for all nodes comprises all network configurations at disposition. Moreover, the formulation transfers any hardware configuration to the associated set of supplied link capacities. The corresponding variables then hold the bounds for the capacity consumption of optical connections to establish.

Lightpath configuration model. The routing of connections in telecommunication networks is typically modeled as a multicommodity flow. Since flows are

typically oriented, it is convenient to direct the commodities. As already indicated by the definitions in Section 2.2.1, we arbitrarily separate each commodity's endnodes into an origin and a destination node and orientate the connections from origin to destination. The connections are then routed as directed flows, although representing bidirected optical connections and thus undirected routing paths which consume capacity in both directions.
For the modeling of network flows, two alternative approaches are known: using *link flow variables* or *path variables*. In general, a link flow model is more compact, while path variables allow for an easy formulation of further routing path restrictions such as the limited length of lightpaths in transparent optical networks. Optical connections in opaque networks allow for both variants, and we first describe the link flow formulation.

Link flow model. We introduce the following variables for directed link flows:

$f^q_{\ell n_1} \in \mathbb{Z}_+$ denotes the number of optical connections of commodity $q \in \mathcal{Q}$ routed on link $\ell \in L$ with $\ell = n_1 n_2$ from n_1 to n_2.

Note that the more familiar notation with n_1, n_2 as indices does not apply here since there can be multiple supply links connecting a node pair, and thus the selected one has to be specified explicitly with additional indication of the flow direction.
The routing model for optical connections then reads:

$$\sum_{\substack{\ell \in L(n_1):\\ \ell = n_1 n_2}} f^q_{\ell n_1} - \sum_{\substack{\ell \in L(n_1):\\ \ell = n_2 n_1}} f^q_{\ell n_2} = \begin{cases} v_q & , n_1 = o_q \,, \\ -v_q & , n_1 = d_q \,, \\ 0 & , \text{otherwise} \end{cases} \qquad \forall\, q \in \mathcal{Q},\ n_1 \in N \tag{2.6a}$$

$$\sum_{q \in \mathcal{Q}} (f^q_{\ell n_1} + f^q_{\ell n_2}) \le y_\ell \qquad \forall\, \ell \in L,\ \ell = n_1 n_2 \tag{2.6b}$$

$$f^q_{\ell n} \in \mathbb{Z}_+ \qquad \forall\, q \in \mathcal{Q},\ n \in N,\ \ell \in L(n) \tag{2.6c}$$

For each commodity $q \in \mathcal{Q}$, the flow conservation constraints (2.6a) ensure that a flow of v_q optical connections originates from o_q, terminates at d_q, and is correctly relayed in all other intermediate nodes. The total flow of all commodities together must not exceed the link capacities independently of the direction, which is expressed by constraints (2.6b). Conditions (2.6c) guarantee flow integrality as required to represent optical connections.
Any non-negative integer solution of (2.6) describes an integer flow for each commodity. It is well known that any such integer flow decomposes into simple paths and directed cycles (see, for instance, Ahuja et al. [2]). The flow on any directed cycle can be reduced to zero without violating (2.6). The network cost cannot increase this way. A cycle-free flow then decomposes into simple paths which correspond to the routed optical connections. This decomposition is not unique, i.e., there can be different path sets that sum up to the same link flow, and the planner has the opportunity to choose a path decomposition that suits best according to second order goals.

Path flow model. An alternative way to represent commodity routings is proposed by use of path variables. Since these variables directly correspond to the optical connections, we can restrict the routing to simple paths. For this, we intro-

duce the following sets:

$\mathcal{P}_{od}$	is the set of all simple paths from $o \in N$ to $d \in N$ in $\mathcal{N}$;
$\mathcal{P} = \bigcup_{\substack{o,d \in N: \\ o \neq d}} \mathcal{P}_{od}$	is the union of all simple paths in $\mathcal{N}$.

The path variables are defined as follows:

$f_p^q \in \mathbb{Z}_+$	denotes for commodity $q \in \mathcal{Q}$ the number of optical connections routed on path $p \in \mathcal{P}_{o_q d_q}$.

We remark that the variable index q is not required as long as no two commodities on the same node pair are involved. Here, we allow for the more general case of having parallel commodities which are to be handled individually, e.g., due to different restrictions regarding bifurcation, survivability, or others.
The routing model with path variables then reads:

$$\sum_{p \in \mathcal{P}_{o_q d_q}} f_p^q = v_q \qquad \forall\, q \in \mathcal{Q} \tag{2.7a}$$

$$\sum_{q \in \mathcal{Q}} \sum_{\substack{p \in \mathcal{P}_{od}: \\ \ell \in L(p)}} f_p^q \leq y_\ell \qquad \forall\, \ell \in L \tag{2.7b}$$

$$f_p^q \in \mathbb{Z}_+ \qquad \forall\, q \in \mathcal{Q},\ p \in \mathcal{P} \tag{2.7c}$$

The routing for each commodity is determined by the selection of the requested number of paths as optical connections in constraints (2.7a). By constraints (2.7b), the capacity of each link limits the number of selectable paths across that link for all commodities in common. Finally, conditions (2.7c) provide flow integrality.
Joining all paths selected for a commodity creates a commodity flow as before. This flow can contain cycles, removable as well, and could be decomposable into a different set of simple paths.

Objective. The optimization task is completed by the objective, which reads

$$\sum_{n \in N} C_n x_n + \sum_{t \in T} C_t x_t + \sum_{n \in N} \sum_{s \in \mathcal{S}} C_n^s x_n^s + \sum_{\ell \in L} \sum_{f \in \mathcal{F}} C_\ell^f x_\ell^f + \sum_{t \in T} \sum_{w \in \mathcal{W}} C_t^w x_t^w \tag{2.8}$$

and has to be minimized.

Model summary. The compilation of the discussed constraints results in complete integer program formulations for the design of opaque optical networks. Depending on the routing model, we distinguish two major variants:

$\mathbf{OND^{l}_{opq}}$ resp. $\mathbf{OND^{p}_{opq}}$

involving

- minimization of (2.8) subject to
- the setup constraints (2.1),
- the node configuration model (2.2) and
- the link configuration model (2.3) with
- the capacity linking constraints (2.5), and
- the **l**ink flow model (2.6) **resp.** the **p**ath flow model (2.7).

Showing that each solution of the formulations represents an operable network according to the practice-oriented description is an easy, but cumbersome exercise and thus left out. Moreover, the integer linear programs $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$ and $\mathrm{OND}^{\mathrm{l}}_{\mathrm{opq}}$ obviously model the same problem, as has already been argued for the alternative flow formulations which state the only difference.

Problem complexity. The flow formulation provides also the key for a brief complexity classification of the problem which contains the well-known integer multicommodity flow problem as subtask:

Lemma 2.13 *The opaque optical network design problem is $\mathcal{NP}$-hard.*

Proof. Originally, Karp [85] has proven the $\mathcal{NP}$-completeness for a simple version of integer multicommodity flow (with disjoint paths only), and Even et al. [46] carried over the result to general integer multicommodity flows, which holds even if only two commodities are considered. In the general case, for a given graph $\mathcal{N} = (N, L)$ with edge capacities $k_\ell \in \mathbb{Z}_+$ and a set $\mathcal{Q}$ of commodities, the task is to decide whether there is an integer flow of the requested size v_q for each commodity $q \in \mathcal{Q}$ such that the union of all flows does not exceed the edge capacities.
By the following transformations, we show that such a decision problem occurs as a subtask in opaque optical network design (and thus reduces to it). For each supply link, we define a specific fiber and WDM system type as $\mathcal{F} = \{\, f_\ell \mid \ell \in L \,\}$, $\mathcal{W} = \{\, w_\ell \mid \ell \in L \,\}$ with $\mathcal{F}_{w_\ell} = \{\, f_\ell \,\}$ and $k^{w_\ell} = k_\ell$. Their exclusive use on the associated supply link is ensured by setting the following parameters. For each physical link $t \in T$, define $b_t^{\mathcal{F}} = |L_t|$ as well as $e_t^{w_\ell} = b_t^{w_\ell} = 1$ if $\ell \in L_t$ and $e_t^w = b_t^w = 0$ otherwise. For each link $\ell \in L$, set $e_\ell^{f_\ell} = b_\ell^{f_\ell} = b_\ell^{\mathcal{F}} = 1$ and $e_\ell^f = b_\ell^f = 0$ for all $f \neq f_\ell$. In addition, the link systems are determined by fixing $z_\ell^{f_\ell w_\ell} = 1$ and $z_\ell^{f w} = 0$ whenever $f \neq f_\ell$ or $w \neq w_\ell$. For the nodes, we set $\mathcal{S} = \{\, s \,\}$ with $k^s = \sum_{\ell \in L} k_\ell$ and $e_n^s = b_n^s = b_n^{\mathcal{S}} = 1$ for all $n \in N$, i.e., use a large enough switch to handle all possible traffic. Finally, we set the variables $x_n = x_t = 1$ for all $n \in N, t \in T$, and $x_t^w = x_\ell^f = x_n^s = 0$ for all types and locations.
As a consequence, all constraints in $\mathrm{OND}^{\mathrm{l}}_{\mathrm{opq}}$ are satisfied except for (2.6a) and (2.6b) which together with the integrality of the link flow variables $f^q_{\ell n_1}$ formulate the original integer multicommodity flow problem. This way, any integer multicommodity flow instance can be reduced to a corresponding instance of opaque optical network design, proving the claim. ■

Formulation comparison. Comparing both formulations in size reveals that the size of $\mathrm{OND}^{\mathrm{l}}_{\mathrm{opq}}$ is polynomial in the input data size, whereas $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$ typically contains an exponential number of path variables. In fact, for any node pair, a meshed network provides a huge number of different paths for connections, each generating a separate path variable. A polynomial number of paths is only given on trivial network structures, such as rings where each node pair can only be connected by two different paths as long as no parallel links are present. In this light, $\mathrm{OND}^{\mathrm{l}}_{\mathrm{opq}}$ provides the more compact integer linear program.
Concerning the sets of feasible solutions, equivalence of the two formulations holds up to routing variations by inserting (or removing) directed cycles at no cost and different flow decompositions into simple routing paths. If both solution spaces are restricted to cycle-free routings for all commodities, it is easy to verify that any

solution of $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$ corresponds to exactly one solution of $\mathrm{OND}^{\mathrm{l}}_{\mathrm{opq}}$, and this mapping is surjective. The other way round, any solution of $\mathrm{OND}^{\mathrm{l}}_{\mathrm{opq}}$ can have several routing path decompositions, each corresponding to exactly one solution of $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$ which is a surjective mapping as well. By these observations, given a commodity routing as a link flow can be interpreted as an aggregation of a set of path flows for which the path decomposition information was lost. This information is dispensable in the opaque scenario, but not in the transparent case where paths have to satisfy further restrictions.

2.2.3 Transparent networks with single-hop traffic

In transparent optical networks, we have to deal with lightpaths which differ from connections in opaque networks in two fundamental properties: wavelength continuity and length restriction. As a consequence, it is not possible anymore to exclude the wavelength assignment subtask from the core problem, and we have to integrate it into the integer linear program formulation. In addition, transmissions are optically forwarded in the nodes, and so we have to account for their total distance over multiple fibers. In particular, different fiber propagation properties become relevant, i.e., each physical link $t \in T$ can in fact decompose into a non-trivial set of supply links L_t with $|L_t| \geq 1$ in general, according to Definition 2.2. In this section, we focus on single-hop traffic and refine the previous problem formulation to transparent optical network design.

Hardware model adaptions. On the hardware side, most constraints for the installation of already integrated modules are not affected by the distinction of wavelengths. In fact, we inherit wide parts of the former models, including the switch configuration model (2.2) and nearly all constraints from the link configuration, namely (2.3b)–(2.3i). The link dimensioning indeed has to account for wavelength capacities which are modeled by the following variables:

$y_\ell^\lambda \in \mathbb{Z}_+$	denotes the number of wavelength channels with wavelength $\lambda \in \Lambda$ on link $\ell \in L$

As a consequence, we have to adapt all constraints dealing with link capacities. So, transmission capacity at a link $\ell \in L$ is now given wavelength-wise by

$$\sum_{w \in \mathcal{W}} \sum_{f \in \mathcal{F}_w} k^{w\lambda} z_\ell^{fw} = y_\ell^\lambda \qquad \forall\, \ell \in L,\ \lambda \in \Lambda \tag{2.9}$$

which substitutes (2.3a) in the former link configuration model. For the link-oriented node dimensioning, the wavelength capacities must be summed up to channel capacities by

$$\sum_{\ell \in L(n)} \sum_{\lambda \in \Lambda} y_\ell^\lambda + \sum_{\substack{q \in \mathcal{Q}: \\ n \in \{o_q, d_q\}}} v_q \leq 2 y_n \qquad \forall\, n \in N \tag{2.10}$$

replacing (2.5). Finally, the link setup constraints (2.1b) change to

$$\sum_{\ell \in L_t} \sum_{\lambda \in \Lambda} y_\ell^\lambda \leq M x_t \qquad \forall\, t \in T\,. \tag{2.11}$$

Conversion capacity. With the distinction of wavelengths, the capability for their conversion comes into play, too. The nodes can be equipped with wavelength converters to provide conversion capacities (whereas regenerators are not allowed for single-hop traffic). Again, auxiliary variables for conversion capacities are helpful for separate modeling of hardware and routing restrictions. So, we introduce the following variables:

$x_n^c \in \mathbb{Z}_+$	denotes the number of *new* wavelength converters of type $c \in \mathcal{C}$ to install at node $n \in N$;
$z_n^c \in \mathbb{Z}_+$	denotes the total number of wavelength converters of type $c \in \mathcal{C}$ to use at node $n \in N$;
$y_n^{\mathcal{C}} \in \mathbb{Z}_+$	denotes the conversion capacity of node $n \in N$.

Hereupon, the hardware configuration model is extended by the wavelength converter model as follows:

$$\sum_{c \in \mathcal{C}} z_n^c = y_n^{\mathcal{C}} \qquad \forall\, n \in N \tag{2.12a}$$

$$z_n^c \le e_n^c + x_n^c \qquad \forall\, n \in N,\ c \in \mathcal{C} \tag{2.12b}$$

$$z_n^c \le b_n^c \qquad \forall\, n \in N,\ c \in \mathcal{C} \tag{2.12c}$$

$$\sum_{c \in \mathcal{C}} z_n^c \le b_n^{\mathcal{C}} \qquad \forall\, n \in N \tag{2.12d}$$

$$z_n^c,\ x_n^c,\ y_n^{\mathcal{C}} \in \mathbb{Z}_+ \qquad \forall\, n \in N,\ c \in \mathcal{C} \tag{2.12e}$$

At each node $n \in N$, constraints (2.12a) account for the conversion capacity offered by the used modules. Moreover, constraints (2.12b) take care of necessary new installations of each converter type at each node, and the bounds on type numbers are expressed by constraints (2.12c), whereas constraints (2.12d) limits the total number of converters at each node. The integrality conditions (2.12e) complete this model. Since applying conversions presumes the existence of switching capacity, the converters need not be considered explicitly for the node setups which are driven by utilization of switches. Further changes of the hardware part are not needed for adaption to transparent optical networks with single-hop traffic.

Lightpath configuration model. In the routing part, indeed, lightpaths are now able to span multiple links, which makes optical connections more difficult to model. Since path length restrictions become important, we cannot anymore rely on link flow variables by which individual routing connections need not be uniquely determined. The routing of lightpaths therefore has to be modeled by path variables. In turn, this provides also advantages: Using the knowledge of the link lengths, the lightpath reach can implicitly be handled by removing too long paths from the respective path sets. This way, intricate conditions are avoided, and the number of path variables is reduced accessorily. We only have to define the adapted path sets as follows:

$\widetilde{\mathcal{P}}_{od} \subset \mathcal{P}_{od}$	is the set of all simple paths p in $\mathcal{N}$ from $o \in N$ to $d \in N$ which are feasible lightpath routes, i.e., have a total length of $\sum_{\ell \in L(p)} \omega_\ell \le \Omega$;

$\widetilde{\mathcal{P}} = \bigcup_{\substack{o,d\in N:\\ o\neq d}} \widetilde{\mathcal{P}}_{od}$ is the union of all such simple paths in $\mathcal{N}$ which do not exceed the maximum transmission length.

Besides their routing paths, the lightpaths additionally require a conflict-free wavelength assignment. To incorporate this issue, one possibility is to refine the path sets to holding complete lightpaths with preassigned wavelengths. Due to the immense variability of assigning wavelengths link-wise, this would create an intractable magnitude of such sets. The better alternative is to introduce separate wavelength assignment variables, accompanied by further variables for counting the implied conversions:

$a^\lambda_{p\ell} \in \mathbb{Z}_+$ denotes the number of times wavelength $\lambda \in \Lambda$ is assigned to lightpaths routed along $p \in \widetilde{\mathcal{P}}$ on link $\ell \in L(p)$;

$a^\lambda_{pn} \in \mathbb{Z}_+$ denotes the number of lightpaths routed along $p \in \widetilde{\mathcal{P}}$ whose wavelength is converted from $\lambda \in \Lambda$ to another wavelength in intermediate node $n \in N(p)$.

Using these variables, the lightpath configurations can be described by a common model for routing and wavelength assignment, which reads:

$$\sum_{p\in\widetilde{\mathcal{P}}_{o_q d_q}} f^q_p = v_q \qquad \forall\, q \in \mathcal{Q} \tag{2.13a}$$

$$\sum_{\lambda\in\Lambda} a^\lambda_{p\ell} = \sum_{\substack{q\in\mathcal{Q}:\\ p\in\widetilde{\mathcal{P}}_{o_q d_q}}} f^q_p \qquad \forall\, p \in \widetilde{\mathcal{P}},\ \ell \in L(p) \tag{2.13b}$$

$$\sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ \ell\in L(p)}} a^\lambda_{p\ell} \le y^\lambda_\ell \qquad \forall\, \ell \in L,\ \lambda \in \Lambda \tag{2.13c}$$

$$a^\lambda_{p\ell_i} - a^\lambda_{p\ell_{i+1}} \le a^\lambda_{pn_i} \qquad \forall\, \lambda \in \Lambda,\ p \in \widetilde{\mathcal{P}},\ 1 \le i \le h-1, \text{ where } p = (n_0, \ell_1, n_1, \ell_2, n_2, \ldots, \ell_h, n_h) \tag{2.13d}$$

$$\sum_{\lambda\in\Lambda} \sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ n\in N(p)}} a^\lambda_{pn} \le y^{\mathcal{C}}_n \qquad \forall\, n \in N \tag{2.13e}$$

$$f^q_p,\ a^\lambda_{p\ell},\ a^\lambda_{pn} \in \mathbb{Z}_+ \qquad \forall\, q \in \mathcal{Q},\ p \in \widetilde{\mathcal{P}},\ \lambda \in \Lambda,\ \ell \in L,\ n \in N \tag{2.13f}$$

The demand constraints (2.13a) ensure that all commodities are satisfied using lightpaths routed on the available paths of restricted length. For the selected number of lightpaths along any given route $p \in \widetilde{\mathcal{P}}$, the wavelength assignment is determined link-wise in constraints (2.13b), providing the respective number of wavelengths on each link. Constraints (2.13c) state that all traversing lightpaths together cannot exceed the supplied wavelength capacities on any link.

The converter counting requires to look individually on every path p and every non-end node n_i. For modeling purposes only, direct the path in an arbitrary way and let ℓ_i and ℓ_{i+1} be two consecutive links along p with common node n_i. If $a^\lambda_{p\ell_i} = a^\lambda_{p\ell_{i+1}}$ for some wavelength λ, then none of the lightpaths along p needs a converter. If $a^\lambda_{p\ell_i} \neq a^\lambda_{p\ell_{i+1}}$, then the difference determines exactly the number of lightpaths that change from wavelength λ to another wavelength (in case $a^\lambda_{p\ell_i} > a^\lambda_{p\ell_{i+1}}$) or from another wavelength to λ (in case $a^\lambda_{p\ell_i} < a^\lambda_{p\ell_{i+1}}$). Summing up over all such differences over all wavelengths in node n_i clearly yields always zero, while the actual

number of needed converters in a node is given by accumulating only the positive differences for each wavelength. For each inner node of a path, these differences are determined in constraints (2.13d) and hold by variables a^{λ}_{pn}. For each node, their sum over all paths traversing the node and over all wavelengths gives the total number of converters needed at the node and is limited by the provided conversion capacity in constraints (2.13e). Finally, fractional routings, assignments, and converter placements are prohibited by conditions (2.13f).

Objective. The integration of converter costs causes an extension of the objective function to

$$\sum_{n\in N} C_n x_n + \sum_{t\in T} C_t x_t + \sum_{n\in N}\left(\sum_{s\in\mathcal{S}} C^s_n x^s_n + \sum_{c\in\mathcal{C}} C^c_n x^c_n\right) + \sum_{\ell\in L}\sum_{f\in\mathcal{F}} C^f_\ell x^f_\ell + \sum_{t\in T}\sum_{w\in\mathcal{W}} C^w_t x^w_t \tag{2.14}$$

which again has to be minimized.

Model summary. As a result, the complete formulation for transparent optical network design with single-hop routing is assembled to the model:

$\mathbf{OND_{tsh}}$

involving

- minimization of (2.14) subject to
- the setup constraints (2.1) with (2.1b) replaced by (2.11),
- the node configuration model (2.2) extended with
- the wavelength converter model (2.12),
- the link configuration model (2.3) with (2.3a) replaced by (2.9),
- the capacity linking constraints (2.10), and
- the routing and wavelength assignment model (2.13).

Solutions. Given a feasible solution of $\mathrm{OND_{tsh}}$, the associated design of the transparent optical network is completely specified except for the particular wavelength assignment of individual lightpaths. In case multiple lightpaths follow the same route, the assignment variables $a^{\lambda}_{p\ell}$ block sufficiently many wavelengths on each traversed link, but do not dedicate particular wavelengths to each particular lightpath. To accomplish the task, the allocation has to be carried out such that the number of converters indicated by constraints (2.13e) is not exceeded. A polynomial time method to resolve the cumulative wavelength assignments for parallel lightpaths into individual lightpath assignments will be sketched in Chapter 4. This way, any wavelength allocation obtained as solution of $\mathrm{OND_{tsh}}$ can be transformed to a feasible lightpath configuration, though the result need not be uniquely determined. In the other direction, any given operable network design can obviously be transformed into a feasible solution of $\mathrm{OND_{tsh}}$, which verifies the correctness of the model.

Problem complexity. The program $\mathrm{OND_{tsh}}$ contains the integer multicommodity flow problem as a subtask, too, which can be easily observed for the special case

$|\Lambda| = 1$ and Ω arbitrarily large: Without different wavelengths and restricted path lengths, the model $\mathrm{OND}_{\mathrm{tsh}}$ reduces to $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$. Hence, Lemma 2.13 implies that the transparent optical network design problem with single-hop routing is $\mathcal{NP}$-hard as well. Furthermore, it is not hard to see that the wavelength assignment subtask of $\mathrm{OND}_{\mathrm{tsh}}$ corresponds to a coloring-type problem, bringing another difficult subtask into play. A discussion in more detail of this issue will be carried out in Chapter 4.

2.2.4 Transparent networks with multi-hop traffic

A single-hop routing of the traffic is sometimes desired by network operators. Nevertheless, the solution space of possible routings is restricted by such a precondition. In addition, it is not applicable to node pairs for which even the shortest path exceeds the maximum transmission distance. Both handicaps are overcome by allowing for multi-hop lightpath connections. Then, a circuit can be composed of multiple concatenated lightpaths, with regenerators to be placed in between. Fortunately, the models for transparent optical networks with single-hop or multi-hop routing are quite similar.

Hardware model adaptions. Concerning hardware, the most important model adaptions have already been carried out with the changeover from opaque to transparent networks. In fact, all corresponding constraints from $\mathrm{OND}_{\mathrm{tsh}}$ can be reused unchanged. The hardware configuration is just supplemented by regenerator modules which are modeled very similar to converters, including auxiliary regeneration capacity variables. So, we make also use of a similar set of variables:

$x_n^r \in \mathbb{Z}_+$	denotes the number of *new* regenerators of type $r \in \mathcal{R}$ to install at node $n \in N$;
$z_n^r \in \mathbb{Z}_+$	denotes the total number of regenerators of type $r \in \mathcal{R}$ to use at node $n \in N$;
$y_n^{\mathcal{R}} \in \mathbb{Z}_+$	denotes the regeneration capacity of node $n \in N$.

As a further extension of the hardware configuration part, the regenerator model reads:

$$\sum_{r \in \mathcal{R}} z_n^r = y_n^{\mathcal{R}} \qquad \forall\, n \in N \tag{2.15a}$$

$$z_n^r \leq e_n^r + x_n^r \qquad \forall\, n \in N,\ r \in \mathcal{R} \tag{2.15b}$$

$$z_n^r \leq b_n^r \qquad \forall\, n \in N,\ r \in \mathcal{R} \tag{2.15c}$$

$$\sum_{r \in \mathcal{R}} z_n^r \leq b_n^{\mathcal{R}} \qquad \forall\, n \in N \tag{2.15d}$$

$$x_n^r,\ z_n^r,\ y_n^{\mathcal{R}} \in \mathbb{Z}_+ \qquad \forall\, n \in N,\ r \in \mathcal{R} \tag{2.15e}$$

Replacing converter types with regenerator types, these constraints have exactly the same form as those in the wavelength converter model (2.12) and are thus explained the same way. The setup of nodes is also determined without explicitly considering the regenerators, because their use, like for wavelength converters, presumes the existence of switching capacity.

Lightpath configuration. For the routing, the previous single-hop formulation already contains all variables to model the establishment of lightpaths including the associated wavelength assignment, but allows for each commodity only to select among end-to-end lightpaths. Releasing this restriction already enables multi-hop traffic, and thus we need no additional types of variables. It suffices to extend the set of path variables to incorporate other than direct end-to-end connections:

$f_p^q \in \mathbb{Z}_+$ now denotes the number of lightpaths routed on $p \in \widetilde{\mathcal{P}}$ which are used by commodity $q \in \mathcal{Q}$.

Moreover, we inherit the wavelength assignment and converter constraints (2.13c), (2.13d), (2.13e), and (2.13f) which are independent of whether a lightpath is an end-to-end connection or not. The remaining model has to be adapted as follows:

$$\sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ o_p=n}} f_p^q - \sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ d_p=n}} f_p^q = \begin{cases} v_q & , n = o_q\,, \\ -v_q & , n = d_q\,, \\ 0 & , \text{otherwise} \end{cases} \qquad \forall\, q \in \mathcal{Q},\ n \in N \tag{2.16a}$$

$$\sum_{\lambda\in\Lambda} a_{p\,\ell}^{\lambda} = \sum_{q\in\mathcal{Q}} f_p^q \qquad \forall\, p \in \widetilde{\mathcal{P}},\ \ell \in L(p) \tag{2.16b}$$

$$\sum_{\substack{q\in\mathcal{Q}:\\ n\neq d_q}} \sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ d_p=n}} f_p^q \leq y_n^{\mathcal{R}} \qquad \forall\, n \in N \tag{2.16c}$$

Constraints (2.16a) model the multi-hop routing for any commodity by flow conservation for the selected lightpaths, i.e., as (directed) link flow routings in the virtual topology induced by the lightpaths. Note that this in fact formulates a two-layer routing problem. Constraints (2.16b) equal the former constraints (2.13b) except for that all commodities occur in the sum on the right hand side. Additionally, the regeneration capacities have to be taken into account for each node in constraints (2.16c). It states that all lightpaths terminating in a node that is not the dedicated commodity's destination require a regenerator.

Objective. The objective is extended further to incorporate regeneration cost by

$$\begin{aligned} &\sum_{n\in N} C_n x_n + \sum_{t\in T} C_t x_t + \sum_{n\in N} \left(\sum_{s\in\mathcal{S}} C_n^s x_n^s + \sum_{c\in\mathcal{C}} C_n^c x_n^c + \sum_{r\in\mathcal{R}} C_n^r x_n^r \right) \\ &+ \sum_{\ell\in L} \sum_{f\in\mathcal{F}} C_\ell^f x_\ell^f + \sum_{t\in T} \sum_{w\in\mathcal{W}} C_t^w x_t^w \end{aligned} \tag{2.17}$$

to be minimized.

Model summary. So, for transparent optical network design with multi-hop routing, we finally end up with the comprehensive model:

$\mathbf{OND_{tmh}}$

involving

- minimization of (2.17) subject to
- the setup constraints (2.1) with (2.1b) replaced by (2.11),
- the node configuration model (2.2) extended with
- the wavelength converter model (2.12) and the
- the regenerator model (2.15),
- the link configuration model (2.3) with (2.3a) replaced by (2.9),
- the capacity linking constraints (2.10), and
- the adapted routing and wavelength assignment model (2.16) in addition to (2.13c), (2.13d), (2.13e), and (2.13f).

Solutions. Concerning the correspondence of feasible solutions of $\mathrm{OND}_{\mathrm{tmh}}$ and operable network designs in this setting, two non-uniqueness issues combine now: the non-unique wavelength assignment for multiple lightpaths on the same route as before, and the non-unique decomposition of the routing (as flow on lightpaths) into particular circuits as end-to-end connections, a consequence of the two-level routing. In fact, the second issue is equivalent to the non-uniqueness of link flow decompositions into simple paths as discussed for the models $\mathrm{OND}^{\mathrm{l}}_{\mathrm{opq}}$ and $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$, but resides on the virtual topology here instead of the physical topology before. Hence, the correctness of the formulation follows in the same way as before, combining the observations to obtain a mapping of feasible model solutions to operable network designs, and vice versa.

Problem complexity. By restriction to the very same special case as for $\mathrm{OND}_{\mathrm{tsh}}$, choosing $|\Lambda| = 1$ and Ω arbitrarily large, $\mathrm{OND}_{\mathrm{tmh}}$ also reduces to $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$. Unsurprisingly, the most comprehensive formulation $\mathrm{OND}_{\mathrm{tmh}}$ therefore models an $\mathcal{NP}$-hard problem as well, and as obvious extension of $\mathrm{OND}_{\mathrm{tsh}}$, the previously mentioned coloring subtask carries over to $\mathrm{OND}_{\mathrm{tmh}}$, too. It is likely that the additional routing level increases the problem's complexity further.

2.2.5 Further aspects

So far, the network design models for all three scenarios have been derived in a default setting which is binding throughout this thesis. However, planners in practice might prefer distinct specification for some options according to their particular network environments. For instance, if any of the bidirectionality assumptions for capacities, demands, or routings does not hold, we can switch to directed multi-graphs and lightpaths. The corresponding model adaptions indeed are straightforward and thus not stressed explicitly.

Instead, we show that our framework and the developed models are flexible to integrate further aspects and details. Investigating all possible variations and their mutual effects is far out of reach. So, we independently discuss some selected issues which have been excluded before or are considered as practically relevant alterna-

tives. In particular, we study the integration of regeneration points, the possibility to reallocate preinstalled modules, the traffic-oriented dimensioning of switching capacities, and the modeling of non-bifurcation restrictions.

2.2.5.1 Regeneration points

In Section 2.1.1, the insertion of regeneration points has been suggested to integrate the installation of regenerators *on* links which are too long for a direct optical transmission. This idea has the advantage to avoid an explicit distinction whether these modules are placed at nodes or on links. Basically, the key to incorporate regeneration points consists of an extension of the modeling framework.

Framework extension. Consider an arbitrary physical link whose length forbids direct optical transmission for some fiber types. As usual, we generate the supply network by replacing the physical link with a couple of parallel supply links, one for each group of fiber types with identical signal degradation properties. Next, pick an arbitrary supply link whose associated fiber types require on-link regeneration(s). Due to the expenses incurred by the additional hardware, the operators prefer a minimum number of regenerations to be performed. This predetermines possible locations to place the regeneration points, which are also often suggested by geographical conditions or—in case of existing fibers—can be already established. However, we assume that a particular location for each regeneration point can be or is fixed a priori. Then, we subdivide the corresponding supply link accordingly and insert an additional supply node to mark each of the regeneration points.
At a regeneration point, all optical channels transmitted on any fiber have to be fully refreshed by use of a regenerator, determining the regeneration capacity in a traffic-oriented way as in regular nodes. Performing these 3R regenerations requires to demultiplex combined signals, so the network manipulation is also conform with the transmission equipment model, i.e., the prerequisite that WDM systems are installed on supply links. The node functionalities indeed change for regeneration points since the optical channels need not really be switched. For this, we introduce an auxiliary switch type only available at the regeneration nodes for representing the 'no-switching' functionality, similar to the 'no-WDM' system on fibers. The auxiliary switch type has zero cost and arbitrarily large switching capacity.
This way, regeneration points can be handled the same way as 'regular' nodes for hardware dimensioning and connection routing. The proper setting of the supply link lengths takes care for the occupation of regenerators whenever optical channels are routed this way. Note that such a correct length measurement for traversing lightpaths prevents simpler modeling by only adapting WDM system costs and virtual link lengths without subdivision into subsequent links. The use of regeneration points, as desired, maintains the prerequisites that regenerators are always installed at supply nodes as well as that each supply link enables direct transmission, and we avoid complications by having to consider on-link placement as special case.

In principle, transparent networks with single-hop traffic do not allow for any regeneration. Hence, too long fiber links become unusable, and regeneration points need not be inserted. Anyway, it is also possible to keep those fibers and to consider regeneration points as exceptional issue, but we omit a discussion of particular

adaptions for this very special case and refer for the rest only to the models $\mathrm{OND}^{\mathrm{l}}_{\mathrm{opq}}$, $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$, and $\mathrm{OND}_{\mathrm{tmh}}$.

Sketch of model adaptions. To pinpoint all model adaptions, some repetitive formulations would be required to incorporate minor changes. Thus, we restrict to briefly sketch the major consequences in an informal way.
An important issue concerns Definition 2.2 on page 47 where the supply nodes N do not coincide anymore with the physical nodes $N' \subsetneq N$. Fortunately, all constraints for physical nodes remain unchanged, and we focus only on regeneration points in the following. Due to the framework extension, most constraints carry over literally, in particular the lightpath configuration models (2.6), (2.7), and (2.16), as all these constraints are already formulated on the level of supply links independent on their relation to physical links. Clearly, if not still present, the regeneration capacity dimensioning (2.15) and consumption constraints (2.16c) have to be added, with appropriate adaption of the objective (2.8) to incorporate regenerator cost. Other model parts can even be simplified. The auxiliary switch properties allow to fix some variables referring to regeneration points, for instance those in the node dimensioning model (2.2), or for converters in model (2.12) since conversion capabilities are included at regeneration points. Thereby, the link-oriented capacity relations (2.5) respectively (2.10) become redundant for regeneration points.
Next, we turn to links with the dimensioning model (2.3) and setup constraints (2.1b) respectively (2.11). The insertion of regeneration points yields physical links $t \in T$ with an associated set L_t of supply links which need not all be parallel anymore, but also form chains of subsequent links. This has no impact on those model parts which do not depend on the correspondence of physical and supply links. Since already corresponding to individual supply links, this includes the transmission capacity accounting in (2.3a) respectively (2.9), the variables z_ℓ^{fw} representing the assignment of WDM systems to the carrying fibers with incompatibility conditions (2.3h), as well as the fiber number limitations (2.3b), (2.3d), and (2.3f). So, affected are only setup constraints and the remaining dimensioning constraints modeling WDM system utilization as well as the total fiber number on physical links. Regarding setups, the constraints require just to restrict the sum on all links in a single segment, or can remain unchanged except for a multiplication of M to adapt on summing up capacities on subsequent links in the chains. Regarding module limitations, the same multiplication can be applied to the parameters b_t^w and $b_t^{\mathcal{F}}$ to account for subsequently installed WDM systems and fibers. Alternatively, these module number limits can be demanded for each regeneration segment in the chain, i.e., for each parallel supply link subset. Then, constraints (2.3e) and (2.3g) have to be formulated for each of these subsets individually, summing up only the associated supply links on the left hand side.

2.2.5.2 Movable modules

Upgrade planning involves a reuse of already existing equipment in the physical network, and we assume as default that all preinstalled devices are only disposable on their current spot. An alternative option is to allow reallocation for some module types. In this case, a planner disposes of a pool of devices reusable wherever needed, and new modules must be purchased only if the existing ones do not suffice.

Model adaption for device pools. The necessary adaption of the models for integration of module pools is based on an elegant variable redefinition which we explain exemplary for wavelength converters. In constraints (2.12b), the non-negative integer variables x_n^c account for the number of converters of type $c \in \mathcal{C}$ which have to be newly installed at node $n \in N$. By relaxing the non-negativity of $c \in \mathcal{C}$ and turning the inequalities (2.12b) into equations, the same variables hold the difference between the existing and the required number of such modules. Thereby, a negative value indicates the number of devices that are removed from a location and can be employed somewhere else. For each type individually, the total sum of these values over all locations gives the resulting pool balance.
Formally, we turn the variables $x_n^c \in \mathbb{Z}$ into general integers and replace (2.12b) by

$$z_n^c = e_n^c + x_n^c \qquad \forall\, n \in N,\ c \in \mathcal{C}. \tag{2.18a}$$

To indicate by how many new devices the pool has to be filled up, we introduce a location-independent counting variable $x^c \in \mathbb{Z}_+$ for each converter sort $c \in \mathcal{C}$. Their proper setting is guaranteed by adding the constraints

$$\sum_{n \in N} x_n^c \leq x^c \qquad \forall\, c \in \mathcal{C}. \tag{2.18b}$$

Movable devices managed by a pool have typically a location-independent cost, i.e., $C_n^c = C^c$ for all $n \in N$. For the correct cost calculation in the objective (2.17) or (2.14), we replace the corresponding term

$$\sum_{n \in N} \sum_{c \in \mathcal{C}} C_n^c x_n^c \qquad \text{by} \qquad \sum_{c \in \mathcal{C}} C^c x^c.$$

Note that these adaptions can easily be restricted to a subset of the converter types, leaving all other types unmovable as before.

Exactly the same reformulation can be done to integrate device pools for any other module sorts (except for fibers, of course), using a model similar to (2.18) instead of the constraints (2.2b), (2.3c), or (2.15b), respectively, and the corresponding adaption of the appropriate objective cost terms. A related matter is whether the various module type bounds at individual locations are kept or not. Moreover, it is also possible to involve transport cost for device reallocations, but these would require further model extensions which we omit here.

2.2.5.3 Traffic-oriented node dimensioning

In the developed models, dimensioning of switching capacity at the nodes has been carried out link-oriented, i.e., the capacity requirements at a node are measured according to the incident link capabilities, but independent of the traffic that has actually to be handled at the node. This way, the network gains more flexibility in regard of changing traffic patterns, at the price of higher switching capacities at the nodes and, thus, typically higher total cost. In the following, we present the alternative traffic-oriented dimensioning of switching capacities.

Model adaptions. To turn link-oriented into traffic-oriented dimensioning of switching capacities, we remove the corresponding constraints (2.5) respectively

(2.10) from the formulations and extend instead each model's routing part by a new set of constraints as follows. For opaque networks, we add to the link flow formulation (2.6) the constraints

$$\sum_{q\in\mathcal{Q}} \sum_{\substack{\ell\in L(n_1):\\ \ell=n_2n_1}} f^q_{\ell n_2} + \sum_{\substack{q\in\mathcal{Q}:\\ n_1=o_q}} \sum_{\ell\in L(n_1)} f^q_{\ell n_1} \le y_{n_1} \qquad \forall\, n_1 \in N.$$

For each node $n_1 \in N$, the number of all incoming optical channels, including any terminating and transit traffic, plus the number of connections originating at n_1 gives the total number of connections to be switched, as expressed by the left hand side. This determines the minimum switching capacity at node n_1 that has to be supplied for the current lightpath configuration. In the path flow formulation (2.7), a simpler counting principle can be applied. Here, we insert the constraints

$$\sum_{q\in\mathcal{Q}} \sum_{\substack{p\in\mathcal{P}_{o_qd_q}:\\ n\in N[p]}} f^q_p \le y_n \qquad \forall\, n \in N \tag{2.19}$$

where all used paths traversing a node sum up to the total number of connections to be switched at this node.
Both models for transparent networks use a path formulation as well and are enlarged by constraints very similar to (2.19), only different in the selectable paths for the commodities. In the single-hop case, the routing model (2.13) is extended by

$$\sum_{q\in\mathcal{Q}} \sum_{\substack{p\in\widetilde{\mathcal{P}}_{o_qd_q}:\\ n\in N[p]}} f^q_p \le y_n \qquad \forall\, n \in N,$$

whereas the multi-hop routing (2.16) gets the additional constraints

$$\sum_{q\in\mathcal{Q}} \sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ n\in N[p]}} f^q_p \le y_n \qquad \forall\, n \in N.$$

2.2.5.4 Non-bifurcated routing

As default, we have assumed that all commodities allow for bifurcated routing. However, non-bifurcation is important for some services, and thus we briefly show that an appropriate adaption of the models is easy to achieve. For each group of parallel circuits independently, it can be decided whether the corresponding routing has to be non-bifurcated or not. We split the commodity into multiple parallel commodities, one for each subset that has to be routed non-bifurcated, and an additional one containing all arbitrarily routable connections, if present. Therefore, we presume that non-bifurcation restrictions hold for entire commodities in the sequel. We refer to all commodities requesting a non-bifurcated routing as *non-bifurcated commodities* in the following, subsumed in the subset $\bar{\mathcal{Q}} \subset \mathcal{Q}$.

Model adaptions. Due to the auxiliary capacity variables, the hardware configuration and the lightpath configuration have been modeled separately. Therefore, restricting routing possibilities does not affect any hardware configuration constraints.

In the routing models, we have to ensure for each non-bifurcated commodity that all its connections take the same route. To this end, we apply a simple trick: All flow variables for any non-bifurcated commodity are turned into binary variables which select the routing path for all lightpaths concertedly. In addition, each of these variables gets the corresponding demand value as coefficient to express the represented number of connections.
Formally, we redefine for each commodity $q \in \bar{\mathcal{Q}}$ all variables $f^q_{\ell n_1} \in \{0,1\}$ on all supply links (bidirectionally) respectively $f^q_p \in \{0,1\}$ on all actual paths. Then, we replace these variables in the constraints as follows. In opaque networks with link flows, each occurrence of $f^q_{\ell n_1}$ for a commodity $q \in \bar{\mathcal{Q}}$ is substituted by $v_q f^q_{\ell n_1}$ in constraints (2.6a) and (2.6b). Similarly, any term f^q_p for $q \in \bar{\mathcal{Q}}$ is turned to $v_q f^q_p$ in constraints (2.7a) and (2.7b) of the path flow formulation for opaque networks, in constraints (2.13a) and (2.13b) for transparent networks with single-hop traffic, and throughout the routing model (2.16) for transparent networks with multi-hop traffic. As an example, constraints (2.7b) afterwards read

$$\sum_{\substack{p\in\mathcal{P}:\\ \ell\in p}} \left(\sum_{q\in\mathcal{Q}\setminus\bar{\mathcal{Q}}} f^q_p + \sum_{q\in\bar{\mathcal{Q}}} v_q f^q_p \right) \leq y_\ell \qquad \forall\, \ell \in L.$$

In some cases, the resulting constraints eventually can be simplified by division with v_q, as for instance in (2.6a), (2.7a), (2.13a), and (2.16a).

Summary. The integration of individual non-bifurcation requirements for commodities as well as the other variations and extensions presented above make clear that both the underlying framework as well as the derived models form a flexible base for the mathematical support of the network planner in practice and allow to include many relevant details. So, we finish the discussion of minor aspects and turn to the survivability of networks which is of crucial importance in practice.

2.3 Realizing survivability

To make a telecommunication network survivable, specified connections have to be protected against possible failure situations. In case of a malfunction, the normal operation of the network proceeds to an emergency state, where some hardware modules do not work anymore, and some established connections are disrupted. The preparation of appropriate reactions by the network management requires to take those situations into account already in the network design process. For this, we present the survivability concept Demand-wise Shared Protection (DSP) which has been developed for meshed optical networks. DSP provides a new trade-off between resource efficiency, recovery speed, and ease of application.

In this section, we first specify the reliability framework and formalize survivability requirements. Next, we explain the concept in three variants, beginning with the basic idea which is step-wise generalized. Thereby, we discuss conceptual properties and show how DSP (and its variants) can be integrated into the optical network design models. Finally, we compare DSP to closely related known concepts, discussed in Section 1.3.

2.3.1 Notation

The operation of an optical network can be disturbed by a variety of possible failures, affecting the configuration to different extents. As introduced in Chapter 1, we focus on the most important failure situations, the breakdown of single nodes or single links in the physical topology. Such situations, including the normal operation of the network, are interpreted as distinct operation states, each specified by those components which are currently considered out of order.

Definition 2.14 (Operating states) *An **operating state** is defined by a subset $\sigma_{\mathcal{T}} \subset N \cup T$ containing the failing physical links and nodes. The special state $\sigma_{\mathcal{T}} = \emptyset$ is called the **normal operation state**, while any other state is referred to as a **failure state**. Moreover, we denote by $S^1_{\mathcal{T}} = \{\, \sigma_{\mathcal{T}} \subset N \cup T \mid \, |\sigma_{\mathcal{T}}| \leq 1 \,\}$ the set of all operating states in which at most one single physical link or node fails.*

Operating states are defined by means of the physical topology, reflecting that physically colocated hardware is assumed to fail concertedly. However, both the hardware configuration and the lightpath configuration are based on the supply network, and so we have to map the failure states to this topology. In the normal operating state, the supply network clearly remains unchanged. For any failure state $\sigma_{\mathcal{T}} \in S^1_{\mathcal{T}}$, we consider the corresponding set $\sigma \subset N \cup L$ of failing supply network elements, defined as

$$\sigma := \begin{cases} L_t & , \sigma_{\mathcal{T}} = \{\, t \,\} \subset T \,, \\ \{\, n \,\} \cup L(n) & , \sigma_{\mathcal{T}} = \{\, n \,\} \subset N \,, \end{cases}$$

and the corresponding *residual supply network* $\mathcal{N}^\sigma$ defined by

$$\mathcal{N}^\sigma := \begin{cases} (N, L \setminus L_t) & , \sigma = L_t,\ t \in T \,, \\ (N \setminus \{\, n \,\}, L \setminus L(n)) & , \sigma = \{\, n \,\} \cup L(n),\ n \in N \,. \end{cases}$$

So, a failure of a physical link $t \in T$ disables all supply links in L_t, and in case a node $n \in N$ fails, the surviving supply network is obtained by removing this node n which turns all incident links $L(n)$ inoperable as well. For notational convenience, we subsume all operating states σ corresponding to $\sigma_{\mathcal{T}} \in S^1_{\mathcal{T}}$ in the set S^1 as single failure states for the supply network. The transition to a residual supply network in case of a failure disrupts all connections which have traversed at least one of the removed links or nodes.

Regarding the connections, recall that we allow for protected traffic as well as for best-effort traffic. This distinction has to be incorporated into the demand definition. So, we additionally specify the part of each commodity that is to be protected against all relevant failures.

Definition 2.15 (Commodity protection) *For any commodity $q \in \mathcal{Q}$, the additional attribute $v^*_q \in \mathbb{Z}_+$ with $0 \leq v^*_q \leq v_q$ defines the number of optical connections to be protected against any single node or link failure in the physical topology of the optical network, except for failures of the commodity endnodes $o_q, d_q \in N$.*

We remark that the commodity protection requirements can also be refined by requesting an individual number of connections to maintain for each particular failure

state $\sigma \in S^1$. In practice, however, the guaranteed service quality of connections rarely depends on what fails, except for endnode breakdown, of course. Thus, the surviving connection number for a commodity is typically uniform for all failure states.

Obviously, the survival of a particular connection cannot be guaranteed, but in case of disruption, a surrogate circuit unaffected by the failure can be provided to the protected traffic. Having detected the failure, the traffic to survive is shifted to the so-called *backup connection*. It depends on the survivability scheme whether backup connections are preestablished or set up dynamically. In the former case, these connections can also be used by best-effort traffic which can be dropped upon request.

The provision of backup connections presumes a sufficient connectivity of the physical network which is defined by means of path disjointness.

Definition 2.16 (Disjoint paths) *Let $\mathcal{T} = (N, T)$ be an undirected physical topology and $n_1, n_2 \in N$, $n_1 \neq n_2$. Two paths p_1, p_2 in $\mathcal{T}$ from n_1 to n_2 are called **link-disjoint** if $T(p_1) \cap T(p_2) = \emptyset$, i.e., no link $t \in T$ is contained in both of them, and **node-disjoint** or simply **disjoint** if they are link-disjoint and do not share any node $n \in N \setminus \{n_1, n_2\}$ except for their endnodes, i.e., $N(p_1) \cap N(p_2) = \emptyset$ holds for the inner node sets.*

With these terms, we can now introduce the connectivity of a node pair.

Definition 2.17 (Connectivity) *Let $\mathcal{T} = (N, T)$ be an undirected physical topology. For a node pair $n_1, n_2 \in N$ with $n_1 \neq n_2$, the **link-connectivity** in $\mathcal{T}$ is defined as the maximum number of mutually link-disjoint paths connecting n_1 and n_2. Similarly, the **node-connectivity** or simply **connectivity** for n_1, n_2 in $\mathcal{T}$ is defined by the maximum number of mutually node-disjoint paths between n_1 and n_2 and denoted by $\kappa_{n_1 n_2} \in \mathbb{Z}_+$. We also say that a node pair with connectivity κ is κ-connected.*

Menger's theorem [118] and its variations (see, e.g., Schrijver [150]) state min-max relations between connectivities and cut cardinalities in graphs, which are helpful for determination of connectivity values. So, for instance, we know that the connectivity of a pair of non-adjacent nodes equals the minimum size of a node cut separating these nodes. With this, one can easily verify whether a given set of mutually disjoint paths is already of maximum cardinality or not.
Figure 2.3 illustrates path disjointness and connectivity by an exemplary network. We have defined both link- and node-connectivity for the sake of completeness. A restriction of survivability to link failures is sometimes investigated and motivated by the fact that node equipment is often built highly redundant and thus not involved in failure precautions. However, we consider the more general case of survivability against link and node failures. Therefore, we always refer to node-connectivity or simply connectivity in the sequel.

Making use of disjoint backup connections for a commodity in case of any failure requires the endnodes to be at least 2-connected. Today, optical backbone networks often have a meshed topology providing higher connectivity for many node pairs.

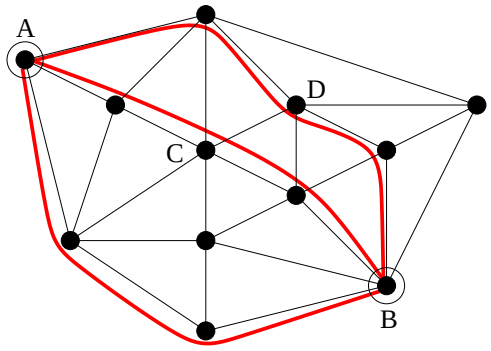


Figure 2.3: *An exemplary network with nodes A and B being 3-connected, proven by the three pairwise disjoint paths. Nodes C and D are 4-connected and have a link-connectivity of 5.*

For example, the meshed network in Figure 2.3 contains in total 28 node pairs with connectivity 3, further 29 node pairs with connectivity 4, and 9 node pairs with connectivity 5. Such topological properties allow for plenty of alternative routings, but most known survivability concepts do not make use of more than disjointness relations for pairs of paths. We have developed an approach exploiting the network connectivity to save spare capacities. For comparisons, we refer often to 1+1 path protection since that concept is widely applied in practice today.

2.3.2 Demand-wise shared protection (DSP)

Demand-wise Shared Protection (DSP) is a survivability concept that has been developed for meshed optical networks. Remind that the terms *demand* and *commodity* are used synonymously. DSP comes in three variants, a basic version denoted as bDSP with adjustable parameters that are to be predefined, a version denoted as pDSP which integrates explicit determination of these parameters into the optimization process, and the general concept DSP that abstracts from these parameters. After explanation of each variant, we show how the scheme can be integrated into the previously described optical network design models.
Some preliminary remarks are necessary to simplify the following descriptions. Whenever we discuss operating states for a commodity, we do not include the breakdown of its origin or destination. Such a failure disables the commodity and thus redundantizes its further consideration. Moreover, we consider connection routings always on the level of the physical topology, i.e., implicitly retransform each connection in the supply network to the corresponding connection in the physical topology. The supply network is of minor relevance for conceptual survivability issues, since connectivity, disjointness, and related terms are defined on the physical topology. For modeling, the corresponding mapping of failing components introduced in Section 2.3.1 is used implicitly without further explanation. The integer linear program modifications restrict on the routing models for the lightpath configuration, while all other model parts remain unchanged. Finally, we remind that the establishment of backup connections does not require installation of additional transmitters and receivers at terminal nodes (cf. Section 2.1.1). Since the data streams are switched in these nodes, too, traffic of a failing connection can be redirected onto a backup

connection already in optical form. Hence, the cost for all transmitters and receivers is determined by the number of requested working connections and does not vary with the number of additionally established backup connections.

2.3.2.1 The basic concept (bDSP)

As the name 'demand-wise shared protection' indicates, spare capacity is shared by the connections belonging to a demand, but not among different demands. Each demand gets spare capacity dedicated to it, and backup connections are always provided as preestablished end-to-end connections. Such connections can be used to accommodate best-effort traffic, if present. Otherwise, provision of backup connections that are only required in failure situations is to be limited in order to reduce total capacity requirements. For this, bDSP exploits the network connectivity by routing diversification.

Diversification. Diversification aims at spreading the normal operation routing of a demand over several different paths. In optical networks, this spreading has to be done in integer amounts as lightpaths cannot be split. For a commodity $q \in \mathcal{Q}$ with a demand value of v_q connections, a *diversification parameter* $0 < \delta_q \leq 1$ is set to define an upper bound $\lfloor \delta_q v_q \rfloor$ on the fraction of the demand allowed to flow through a single link or node (except for origin and destination, of course). Then, any single component failure affects at most $\lfloor \delta_q v_q \rfloor$ connections of the demand, while at least $v_q - \lfloor \delta_q v_q \rfloor$ connections are ensured to survive.
To guarantee existence of such a diversified routing, the parameter δ_q cannot take any value in $(0, 1]$, since the network has to provide sufficient connectivity for the commodity's endnodes. Using Menger's theorem, one observes that a diversification parameter of δ_q presumes $\lceil v_q / \lfloor \delta_q v_q \rfloor \rceil$ disjoint paths to exist between o_q and d_q in order to accommodate the entire demand. Thus, the commodity's connectivity $\kappa_q := \kappa_{o_q d_q}$ poses a natural lower bound of $\frac{1}{\kappa_q} \leq \delta_q$ on selectable diversification parameters δ_q. In fact, we show (in the next proof) that a feasible (integer) routing exists whenever it holds that $\frac{1}{\kappa_q} \leq \delta_q \leq 1$ and δ_q is set such that $\delta_q v_q$ is integer.

Key idea. A straightforward application of diversification can only limit the maximum loss incurred by any link or node breakdown. In general, providing protection against failures for a demand needs to establish additional backup connections. For this, the key idea of bDSP is to consider working and backup connections concertedly and to diversify their joint routing best possible, i.e., such that any single failure affects as few as possible connections, leaving sufficiently many left to carry (or take over) the protected traffic. The total number of connections required this way depends on the connectivity to exploit and can be predetermined as follows.
Consider a commodity q with a demand of v_q connections of which $v_q^* \leq v_q$ have to be protected, $v_q, v_q^* \in \mathbb{Z}_+$. Let origin o_q and destination d_q of q be κ_q-connected, $\kappa_q \geq 2$. Then, we route a total of

$$\widetilde{v}_q := \max \left\{ v_q, \left\lceil \frac{\kappa_q}{\kappa_q - 1} v_q^* \right\rceil \right\} \tag{2.20}$$

connections, including both working and backup connections, with a diversification parameter of

$$\widetilde{\delta}_q := \frac{\widetilde{v}_q - v_q^*}{\widetilde{v}_q} \,. \tag{2.21}$$

The following lemma proves existence and suitability of such routings.

Lemma 2.18 *For a commodity q with connectivity of κ_q and a demand of v_q connections of which $0 \leq v_q^* \leq v_q$ are to be protected against any single link or node failure (except for o_q, d_q), there exist routings with in total $\widetilde{v}_q$ connections and diversification parameter $\widetilde{\delta}_q$ that satisfy all traffic and survivability requirements, i.e., provide at least v_q connections in normal operation and at least v_q^* connections in each failure state from S^1.*

Proof. At first, we show that the given network connectivity allows for such a routing. By $\widetilde{v}_q \geq \frac{\kappa_q}{\kappa_q - 1} v_q^*$ according to (2.20), we get

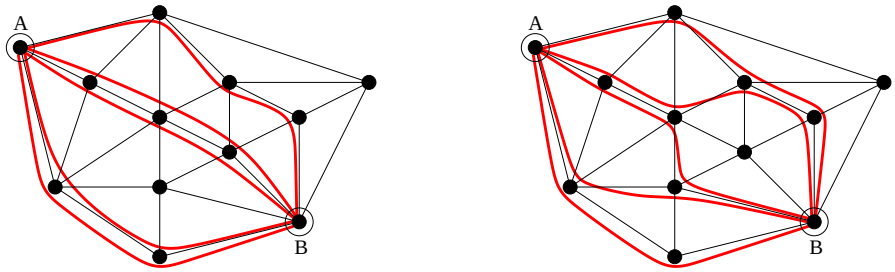
$$\widetilde{\delta}_q \widetilde{v}_q \overset{(2.21)}{=} \widetilde{v}_q - v_q^* \geq \widetilde{v}_q - \frac{\kappa_q - 1}{\kappa_q} \widetilde{v}_q = \widetilde{v}_q - \left(1 - \frac{1}{\kappa_q}\right) \widetilde{v}_q = \frac{1}{\kappa_q} \widetilde{v}_q$$

and hence $\widetilde{\delta}_q \geq \frac{1}{\kappa_q} > 0$. Note also that $\widetilde{\delta}_q \widetilde{v}_q$ is integer by (2.21). In addition, we obviously have $\widetilde{\delta}_q = \frac{\widetilde{v}_q - v_q^*}{\widetilde{v}_q} \leq \frac{\widetilde{v}_q}{\widetilde{v}_q} = 1$. So, $\widetilde{\delta}_q$ is a feasible diversification parameter.
The existence of corresponding routings is now easy to verify. For this, we restrict on routings composed of mutually disjoint paths only (though this is not a necessary condition). As discussed before, a routing of $\widetilde{v}_q$ connections with at most $\widetilde{\delta}_q \widetilde{v}_q$ following the same path requires no more disjoint paths than

$$\left\lceil \frac{\widetilde{v}_q}{\lfloor \widetilde{\delta}_q \widetilde{v}_q \rfloor} \right\rceil = \left\lceil \frac{\widetilde{v}_q}{\widetilde{\delta}_q \widetilde{v}_q} \right\rceil = \left\lceil \frac{1}{\widetilde{\delta}_q} \right\rceil \leq \left\lceil \frac{1}{\frac{1}{\kappa_q}} \right\rceil = \kappa_q$$

as the network provides for q by prerequisite and Definition 2.17.
Regarding satisfaction of the stated requirements, (2.20) directly implies $\widetilde{v}_q \geq v_q$, and thus the routing contains sufficiently many connections for normal operation. Moreover, the survivability requirements are satisfied as well. Due to the applied diversification, any single link or node failure affects at most $\lceil \widetilde{\delta}_q \widetilde{v}_q \rceil$ connections, and thus at least $\widetilde{v}_q - \lceil \widetilde{\delta}_q \widetilde{v}_q \rceil = \widetilde{v}_q - \widetilde{\delta}_q \widetilde{v}_q \overset{(2.21)}{=} v_q^*$ connections survive, as requested. ■

The proof represents the formal explanation of the key idea for the concept. With connectivity κ_q available for a commodity q, a routing could in fact use κ_q disjoint paths to establish all working and backup connections. Then, any single failure disrupts at most the connections along one of these paths. For minimizing the number of connections disrupted by an arbitrary failure, the $\widetilde{v}_q$ connections are best distributed uniformly on the disjoint paths. In the mean, each path then carries $\frac{1}{\kappa_q} \widetilde{v}_q$ connections, and any failure is survived by at least $\frac{\kappa_q - 1}{\kappa_q} \widetilde{v}_q$ connections, demanded to be at least v_q^*. This inequality is finally transformed to the second term in (2.20), where the rounding takes care for distributing indivisible connections instead of the (possibly fractional) mean values. As already indicated in the proof, we remark that a routing on disjoint paths is only assumed for theoretical derivation of the values $\widetilde{v}_q$



(a) bDSP *routing on disjoint paths.*

(b) bDSP *routing on non-disjoint paths.*

Figure 2.4: *A network with exemplary* bDSP *routings for* $v_{AB} = 5$ *and* $v^*_{AB} = 3$.

and δ_q which characterize the required diversification. The set of possible routings satisfying the corresponding restrictions does in general not restrict on exclusive use of mutually disjoint routing paths.
Summarizing, bDSP takes advantage of the individual network connectivity for a demand and diversifies the routing of working and backup lightpaths together. Thereby, it reduces the potential impact of any failure and thus decreases the total number of required connections. The following examples demonstrate this effect and show also a feasible bDSP routing without any pair of disjoint routing paths.

Examples. Consider the network in Figure 2.4(a) where a demand from A to B asks for $v_q = 5$ connections of which $v^*_q = 3$ have to be protected. The network provides a connectivity of $\kappa_q = 3$ between the origin and destination. In this case, $1+1$ path protection would generate 8 connections. For bDSP, we get $\widetilde{v}_q = \max\{5, \lceil \frac{1}{1-\frac{1}{3}} 3 \rceil\} = \max\{5, 5\} = 5$, and $\widetilde{\delta}_q = \frac{5-3}{5} = 0.4$. A possible bDSP routing could use $\lceil \frac{5}{2} \rceil = 3$ node-disjoint paths as shown in Figure 2.4(a). This routing illustrates also the principle idea of bDSP, encoded in the second term of the maximum in (2.20): By distributing traffic in a uniform way on κ_q mutually disjoint paths which exist by prerequisite, at most a portion of $\frac{1}{\kappa_q}$ fails at the same time, and the remaining traffic of $(1 - \frac{1}{\kappa_q})\widetilde{v}_q$ must be at least v^*_q. However, bDSP routings are not restricted to disjoint paths. Figure 2.4(b) shows an alternative routing for the very same parameters.
For the same node pair with demand value $v^*_q = 4$, $1+1$ path protection installs in total 9 lightpaths. bDSP asks for totally $\widetilde{v}_q = 6$ connections and sets the diversification parameter to $\widetilde{\delta}_q = 1/3$. Hence, not more than two connections are allowed to traverse any link or node different from A, B. Both routings in Figure 2.4 can be easily extended to the increased commodity by adding a connection appropriately.

Minimum connection number. The introduced bDSP routings in fact establish a minimum total number of connections, as proven by the following lemma.

Lemma 2.19 *For a commodity q with connectivity $\kappa_q \geq 2$, any connection routing providing v_q working connections in normal operation and at least v^*_q connections in any single link or node failure state, $0 \leq v^*_q \leq v_q$, has to establish at least a total number of $\widetilde{v}_q$ connections according to* (2.20).

Proof. Assume such a routing with strictly less than $\widetilde{v}_q$ connections exists, and let $v \in \mathbb{N}$ be the number of connections in this routing. Due to the number of working connections, $v \geq v_q$. Hence, the assumption holds only if v is strictly smaller than the second term in the maximum in (2.20), and since v is integer, we get

$$v < \frac{\kappa_q}{\kappa_q - 1} v_q^* \,. \tag{2.22}$$

As prerequisite, the commodity has connectivity κ_q. By Menger's theorem, there exists a generalized o_q, d_q-cut $\Gamma \subset T \cup N \setminus \{o_q, d_q\}$ in $\mathcal{T}$ of size $|\Gamma| = \kappa_q$. The assumed routing sends in total at least v connections through the cut Γ. Hence, at least one cut element $\gamma \in \Gamma$ is traversed by at least v/κ_q connections (we abstain from a rounding up, if possible). Hence, if this link or node γ fails, at least v/κ_q connections are disrupted. So, the surviving number of connections is at most

$$v - \frac{v}{\kappa_q} = \frac{\kappa_q - 1}{\kappa_q} v \overset{(2.22)}{<} \frac{\kappa_q - 1}{\kappa_q} \frac{\kappa_q}{\kappa_q - 1} v_q^* = v_q^*,$$

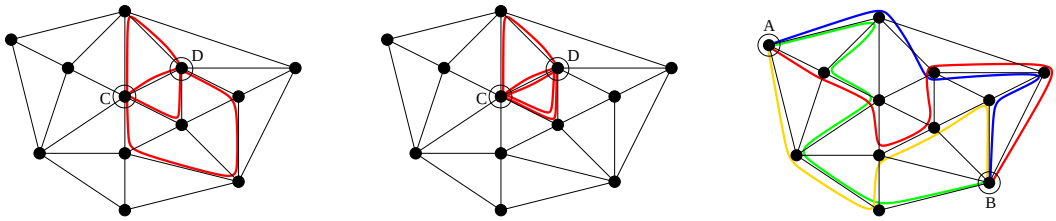
contradicting the second prerequisite. ■

Hence, the described bDSP scheme in fact calls for a minimum total number of connections.

Connectivity as parameter. So far, we implicitly assumed that the available connectivity is to be fully exploited for each commodity. This way, $\widetilde{v}_q$ is minimized. However, it might be reasonable to rely on less connectivity κ_q' with $2 \leq \kappa_q' \leq \kappa_q$ in order to enable shorter connections. Consider Figure 2.5 where nodes C and D are 4-connected, and a corresponding commodity q requests $v_q = 4$ connections of which $v_q^* = 3$ are to be protected. With $\kappa_q' = 4$, we need only $\widetilde{v}_q = 4$ connections in total, but we have to use disjoint paths, as shown in Figure 2.5(a). At least one connection will have to span at least four links in any alternative routing. In contrast, premising a lower connectivity of $\kappa_q' = 3$ in (2.20) yields $\widetilde{v}_q = 5$ connections to route, but allows for the routing shown in Figure 2.5(b) with a lower total hop number.
In fact, the physical topology of the network prescribes a maximum connectivity κ_q for each commodity $q \in \mathcal{Q}$, but the concept does not restrict to use this value. Instead, the connectivity to exploit can be seen as parameter κ_q' selectable from the set $\{2, 3, \ldots, \kappa_q\}$ individually for each commodity. This way, bDSP is parameterized.
Note that setting $\kappa_q' = 2$ does not reduce the concept to 1+1 protection. For A, B with $v_q = v_q^* = 2$ and $\kappa_q' = 2$, we obtain $\widetilde{v}_q = 4$ with $\widetilde{\delta}_q = 0.5$, and Figure 2.5(c) shows a pathological bDSP routing for this case. Since any pair of connections shares a link, this routing cannot be decomposed into a pair of cycles as would be established by 1+1 protection.
The following lemma states a basic property when varying the connectivity to exploit.

Lemma 2.20 *With increasing connectivity parameter κ_q', the corresponding number $\widetilde{v}_q$ of connections to establish decreases monotonically.*

Proof. It suffices to show the claimed monotony for the second term in (2.20).
For increasing κ_q', one knows that $\frac{1}{\kappa_q' - 1}$ decreases monotonically, and so does the



(a) *Connectivity 4.* (b) *Connectivity 3.* (c) *Connectivity 2.*

Figure 2.5: *b*DSP *routings with different connectivity parameters.*

constantly extended series $1 + \frac{1}{\kappa'_q - 1} = \frac{\kappa'_q}{\kappa'_q - 1}$. Since $v^*_q \geq 0$, this carries over to $\frac{\kappa'_q}{\kappa'_q - 1} v^*_q$. With the monotony of the rounding up operation, i.e., $a \geq b \Rightarrow \lceil a \rceil \geq \lceil b \rceil$, the claim follows now immediately. ■

Since $\kappa'_q \geq 2$, the proof shows also that $\widetilde{v}_q \leq \max\{v_q, 2v^*_q\}$ holds in any case.

2-DSP and max-DSP. In the basic variant bDSP, the connectivity parameters are to be fixed prior designing the network. Two particular settings are highlighted as special cases. If each commodity exploits the maximum connectivity provided by the topology, we refer to the scheme as max-DSP. The opposite extreme is to set the minimal value $\kappa'_q = 2$ for each $q \in \mathcal{Q}$, called 2-DSP. Any other selection of connectivity parameters generates an individual bDSP scheme.

Model integration of bDSP. For integrating bDSP into the previously presented models for optical network design, we do not restrict on a particular scheme. We only assume to have (feasible) predefined connectivities to exploit for each commodity. The required total number $\widetilde{v}_q$ of connections and the corresponding diversification parameter $\widetilde{\delta}_q$ are then precomputed according to (2.20) and (2.21).
The necessary model adaptions are presented separately for each proposed integer linear programming formulation. In common for all models, we have to replace the original number v_q of requested connections by the new demand values $\widetilde{v}_q$ in all the constraints (2.6a), (2.7a), (2.13a), and (2.16a). Furthermore, the models need only be extended to include the appropriate diversification restrictions, forming simple bounds for the traffic flows. The straightforward formulation of the additional conditions makes it easy to verify that the proposed modifications ensure proper routings according to bDSP, and thus formal proofs are left out (which holds for the other DSP variants, too).

Opaque networks with link flows. In $\mathrm{OND}^{\mathrm{l}}_{\mathrm{opq}}$, we add the following diversification flow bounds on the commodity routings:

$$\sum_{\substack{\ell \in L(n_1): \\ \ell = n_1 n_2}} f^q_{\ell n_1} \leq \widetilde{\delta}_q \widetilde{v}_q \qquad \forall\, q \in \mathcal{Q},\ n_1 \in N \setminus \{o_q, d_q\} \tag{2.23a}$$

$$\sum_{\ell \in L_t} f^q_{\ell o_q} \leq \widetilde{\delta}_q \widetilde{v}_q \qquad \forall\, q \in \mathcal{Q} :\ o_q d_q = t \in T \tag{2.23b}$$

For each commodity $q \in \mathcal{Q}$, constraints (2.23a) restrict the number of connections routable through each physical node (except for origin and destination of the commodity) to the value prescribed by diversification. The flow throughput is deter-

mined by the outflow at each node, which equals the inflow by the flow conservation constraints (2.6a). Since the throughput limits are uniform for physical links and nodes, any node restriction implies that no feasible routing exceeds the flow bounds on all incident links as well. Hence, physical link flow restrictions are redundant except for direct links between origin and destination of a commodity, since these links are not incident to a flow restricted node. Constraints (2.23a) formulate the routing restrictions on such direct links, where flow need only be considered in direction from origin to destination (the opposite flow value can always be fixed to zero).
We briefly remark that the redundancy of all other physical link throughput restrictions holds only for the case of protection against all single link or node failures. Other non-trivial selections of relevant operating states, like taking only link failures into account, invalidate the utilized implication. For such cases, the formulation has to be replaced or extended by explicit flow limitations (accounting for flow in both directions) for physical links, too. This observation clearly carries over to the following adaptations of other models as well.

Opaque networks with path flows. For $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$, the diversification constraints read:

$$\sum_{\substack{p\in\mathcal{P}_{o_q d_q}:\\ n\in N(p)}} f^q_p \le \widetilde{\delta}_q \widetilde{v}_q \qquad \forall\, q\in\mathcal{Q},\ n\in N\setminus\{o_q,d_q\} \tag{2.24a}$$

$$\sum_{\substack{p\in\mathcal{P}_{o_q d_q}:\\ |L(p)|=1}} f^q_p \le \widetilde{\delta}_q \widetilde{v}_q \qquad \forall\, q\in\mathcal{Q}:\ o_q d_q = t\in T \tag{2.24b}$$

This model extension is quite similar to (2.23) except for determining flow throughput now by counting of the appropriate traversing connections expressed with use of path variables. Remind that in the opaque model, the connections for any commodity q are represented by end-to-end routing paths, and thus each node $n\in N\setminus\{o_q,d_q\}$ occurs always as an inner node in the paths for q, which explains the sum condition in constraints (2.24a). The direct link flow limitation in constraints (2.24b) subsumes the remaining routing paths without inner nodes for each commodity, i.e., those connections which use a single (direct) link only.

Transparent networks with single-hop traffic. The additional constraints for $\mathrm{OND}_{\mathrm{tsh}}$ are very similar to those for $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$ in (2.24):

$$\sum_{\substack{p\in\widetilde{\mathcal{P}}_{o_q d_q}:\\ n\in N(p)}} f^q_p \le \widetilde{\delta}_q \widetilde{v}_q \qquad \forall\, q\in\mathcal{Q},\ n\in N\setminus\{o_q,d_q\} \tag{2.25a}$$

$$\sum_{\substack{p\in\widetilde{\mathcal{P}}_{o_q d_q}:\\ |L(p)|=1}} f^q_p \le \widetilde{\delta}_q \widetilde{v}_q \qquad \forall\, q\in\mathcal{Q}:\ o_q d_q = t\in T \tag{2.25b}$$

In the transparent single-hop case, all connections correspond to end-to-end lightpaths. So, the opaque path flow model extension can be reused just with the appropriate restriction of selectable routing paths.

Transparent networks with multi-hop traffic. For $\mathrm{OND}_{\mathrm{tmh}}$, a node different from origin or destination of a commodity need not anymore be an inner node

of the selectable paths. The diversification constraints therefore read now:

$$\sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ n\in N[p]\setminus\{o_p\}}} f_p^q \le \widetilde{\delta}_q\widetilde{v}_q \qquad \forall\, q\in\mathcal{Q},\ n\in N\setminus\{o_q,d_q\} \tag{2.26a}$$

$$\sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ L(p)\subset L_t}} f_p^q \le \widetilde{\delta}_q\widetilde{v}_q \qquad \forall\, q\in\mathcal{Q}:\ o_qd_q=t\in T \tag{2.26b}$$

In constraints (2.26a), a node's throughput for a commodity q is again measured by the inflow, composed of traversing and terminating lightpaths. For this, associated lightpaths p beginning at a node $o_p = n \in N\setminus\{o_q,d_q\}$ are excluded to avoid double accounting. The additional limitation of direct link flows can reuse constraints (2.25b) restricted to the subset $p\in\widetilde{\mathcal{P}}_{o_qd_q}\subset\widetilde{\mathcal{P}}$ of end-to-end lightpaths for a commodity q. For the alternative of having uniformly $\widetilde{\mathcal{P}}$ as path set in the model extension, constraints (2.26b) show an appropriately adapted sum condition. Since single link lightpaths need not correspond to end-to-end connections anymore, the restriction to lightpaths on a direct link for a commodity must be formulated straightforward.

2.3.2.2 The parameter optimizing concept (pDSP)

In the basic bDSP variant, an individual connectivity parameter is preselected for each demand. The examples in Figure 2.5 have shown that different routing possibilities result from different connectivity values, each with an individual connection number and diversification parameter. Indeed, it is not clear a priori which of these alternatives provides the most preferable routings from a total cost perspective.

Example. Consider again the commodity example illustrated in Figure 2.5. The routing in Figure 2.5(b) (exploiting connectivity three) has less hops in total than the routing in Figure 2.5(a) (exploiting the maximum connectivity of four) and seems more favorable when using the total hop number as (simple) capacity consumption and thus routing cost indicator. However, the advantage changes if any link and node (except for C and D) offers a free capacity of exactly one. In this case, the maximally diversified routing in Figure 2.5(a) has zero cost, whereas any routing exploiting a connectivity of three (establishing five connections) must use at least one link or node twice (since not more than four node-disjoint paths exist) and thus incurs additional cost for expanding the corresponding capacity.

Parameter integration. As a straightforward bDSP generalization, pDSP explicitly integrates the individual connectivity parameter selection for each commodity into the network design optimization. With these additional decisions, a suitable parameterization of all commodity routings can be selected in order to fine-tune total capacity requirements and thus reduce the total design cost.
The simplest way for an integration of the connectivity parameter determination is to formulate the connectivity selection explicitly by additional variables for each demand. Unfortunately, the total number $\widetilde{v}_q$ of required connections according to (2.20) does not depend linearly on the explored connectivity. As example, for a commodity with values $v_q = 5$ and $v_q^* = 4$, the corresponding values of $\widetilde{v}_q$ for con-

nectivities 2, 3, 4, 5 are 8, 6, 6, 5, respectively. Therefore, we have in general to use a separate binary variable for each possible value of $\kappa'_q \in \{2, \dots, \kappa_q\}$ for each commodity with $\kappa_q > 2$. Instances with many commodities between highly connected endnodes can this way obtain a substantial number of additional binary variables. For instance, greenfield planning taking any potential link into account has to consider a complete physical topology, and each node pair has a connectivity of $|N| - 2$.

Model integration of pDSP. The explicit formulation of the connectivity parameter selection is realized in exactly the same way in the models for all network architectures. We therefore describe just the required adaptions in common. These adaptions apply to the already extended models involving bDSP as described in the preceeding section.
Consider an arbitrary commodity $q \in \mathcal{Q}$ with origin-destination connectivity $\kappa_q = \kappa_{o_q d_q} > 2$. For the parameter value selection, we introduce the following variables:

$u_q^i \in \{0,1\}$ denotes whether connectivity $i \in \{2, \dots, \kappa_q\}$ for commodity $q \in \mathcal{Q}$ is exploited ($u_q^i = 1$) or not.

Moreover, we define the corresponding total number of connections to route for each potential connectivity value $i \in \{2, \dots, \kappa_q\}$ as individual parameter

$$\widetilde{v}_q^i := \max\left\{v_q, \left\lceil \frac{i-1}{i} v_q^* \right\rceil\right\}.$$

In any model, we add the constraints

$$\sum_{i=2}^{\kappa_q} u_q^i = 1 \qquad \forall\, q \in \mathcal{Q} \tag{2.27}$$

to ensure that exactly one connectivity value is selected. For each commodity $q \in \mathcal{Q}$, the number of connections to route is then given by

$$\sum_{i=2}^{\kappa_q} \widetilde{v}_q^i u_q^i$$

which replaces the former parameter $\widetilde{v}_q$ throughout all constraints except for the diversification extensions (2.23)–(2.26). In these constraints, we instead make use of (2.21) and replace the right hand side $\widetilde{\delta}_q \widetilde{v}_q$ by the equivalent term

$$\sum_{i=2}^{\kappa_q} \widetilde{v}_q^i u_q^i \; - v_q^*$$

without explicit inclusion of the diversification parameter (which depends on the connection number and thus as well on the selected connectivity).

Solutions. With the described extension, each solution of the models specifies the used connectivity by the variables u_q^i individually for each commodity. Any such solution is clearly also feasible for the basic bDSP scheme with exactly the same connectivities prespecified (i.e., setting the connectivity parameters κ'_q as indicated

by the variables $u_q^i = 1$ for all commodities $q \in \mathcal{Q}$).
In fact, pDSP reflects a concept generalization which integrates all individual bDSP schemes at once and enables to select a particular scheme which suits best for the instance. However, the common feature of both variants, bDSP and pDSP, is the explicit handling of the explored connectivities, either as parameters or by variable representations. A further conceptual generalization allows to abstract from these entities.

2.3.2.3 The general concept (DSP)

Consider an arbitrary commodity q demanding for v_q connections of which v_q^* are to be protected. Given a proposed routing for this commodity, it is easy to verify whether all requests are satisfied: simply by counting the number of connections in total, which must be *at least* v_q, and for each physical link or node endangered to fail, counting those connections that do *not* traverse this element, which must be *at least* v_q^*. Notice that such a verification involves neither the connectivity of the commodity nor the minimum total number of connections needed for its exploitation, as requested by both bDSP and pDSP, when having fixed the connectivity to use. In fact, providing more connections than necessary is not a handicap as long as no additional capacity extensions are needed for their establishment. This observation shows that predetermination and request of a particular total connection number for a commodity (derived from a particular connectivity) can be released, which lays the ground for a further conceptual generalization.

Key idea of DSP. The key idea of the general concept DSP is to repostulate the routing and survivability requirements as follows. For each commodity $q \in \mathcal{Q}$, ensure that

$$\left.\begin{array}{l} \text{1. at least(!) } v_q \text{ connections are established in total, and} \\ \text{2. at least(!) } v_q^* \text{ of the established connections survive in any considered failure state } \sigma \in S^1. \end{array}\right\} \tag{2.28}$$

Each commodity routing according to these two conditions provides (at least) the requested protection. Note that the total number of connections for a commodity is not specified this way. Instead, the failure states are considered explicitly. Nevertheless, both conditions can be easily translated into linear inequalities for the routing models, as shown next before discussing further properties of DSP.

Model integration of DSP. The general concept DSP consists of integrating the repostulated requirements (2.28) directly into the optical network design models. Since these conditions abstract from explicit consideration of connectivities, the formulation does not occur as further extension of the previous model adaptions for bDSP and pDSP, but is based on the original models presented in Section 2.2. The modifications again affect only the routing parts and are presented individually for each architecture and model alternative.

Opaque networks with link flows. In $\text{OND}^{\text{l}}_{\text{opq}}$, the flow conservation constraints (2.6a) are substituted by:

$$\sum_{\substack{\ell\in L(n_1):\\ \ell=n_1n_2}} f^q_{\ell n_1} - \sum_{\substack{\ell\in L(n_1):\\ \ell=n_2n_1}} f^q_{\ell n_2} \quad \begin{cases} \geq \quad v_q \ , n_1 = o_q \, , \\ \leq -v_q \ , n_1 = d_q \, , \\ = \quad 0 \ , \text{otherwise} \end{cases} \qquad \forall\, q \in \mathcal{Q},\ n_1 \in N \tag{2.29a}$$

This encodes the first requirement in (2.28). For the second, we have to add the constraints

$$\sum_{\ell\in L(o_q)} f^q_{\ell o_q} - \sum_{\substack{\ell\in L(n):\\ \ell=n_1n}} f^q_{\ell n_1} \geq v^*_q \qquad \forall\, q \in \mathcal{Q},\ \{n\} = \sigma \in S^1,\ n \neq o_q, d_q \tag{2.29b}$$

$$\sum_{\ell\in L(o_q)} f^q_{\ell o_q} - \sum_{\ell\in L_t} f^q_{\ell o_q} \geq v^*_q \qquad \forall\, q \in \mathcal{Q} :\ o_qd_q = t \in T, L_t = \sigma \in S^1 \tag{2.29c}$$

limiting the commodity's throughputs similarly to (2.23). The first term on the left hand sides counts the actual total number of connections routed for a commodity. Reducing this flow amount by the flow through a (physical) node, measured again as inflow in constraints (2.29b), must at least provide the number of connections that has to survive in case the node fails. As for the basic scheme, the node conditions imply the same property for all incident links as well. The missing conditions for flow on direct links between origin and destination of commodities are provided in constraints (2.29c) in the same way.

Opaque networks with path flows. The model extension for $\text{OND}^{\text{p}}_{\text{opq}}$ is as follows. Constraints (2.7a) formulate the normal operation flow and are simply turned into inequalities

$$\sum_{p\in\mathcal{P}_{o_qd_q}} f^q_p \geq v_q \qquad \forall\, q \in \mathcal{Q} \tag{2.30a}$$

according to the first condition in (2.28), whereas the second condition requires a similar extension as above by

$$\sum_{p\in\mathcal{P}_{o_qd_q}} f^q_p - \sum_{\substack{p\in\mathcal{P}_{o_qd_q}:\\ n\in N(p)}} f^q_p \geq v^*_q \qquad \forall\, q \in \mathcal{Q},\ \{n\} = \sigma \in S^1,\ n \neq o_q, d_q \tag{2.30b}$$

$$\sum_{p\in\mathcal{P}_{o_qd_q}} f^q_p - \sum_{\substack{p\in\mathcal{P}_{o_qd_q}:\\ |L(p)|=1}} f^q_p \geq v^*_q \qquad \forall\, q \in \mathcal{Q} :\ o_qd_q = t \in T, L_t = \sigma \in S^1 \tag{2.30c}$$

using path variables instead of flow variables. We remark that the left hand side of some inequalities can be simplified, in case of (2.30b) to $\sum_{p\in\mathcal{P}_{o_qd_q}:n\notin N(p)} f^q_p$ for instance, while we keep the shown form for the sake of uniformity.

Transparent networks with single-hop traffic. For OND_{tsh}, the DSP model adaption is very similar to (2.30), since both formulations differ only in the sets of selectable paths. Here, constraints (2.13a) are replaced by the inequalities

$$\sum_{p\in\widetilde{\mathcal{P}}_{o_qd_q}} f^q_p \geq v_q \qquad \forall\, q \in \mathcal{Q} \tag{2.31a}$$

and the corresponding extension for failure states reads:

$$\sum_{p\in\widetilde{\mathcal{P}}_{o_q d_q}} f_p^q - \sum_{\substack{p\in\widetilde{\mathcal{P}}_{o_q d_q}:\\ n\in N(p)}} f_p^q \geq v_q^* \qquad \forall\, q\in\mathcal{Q},\ \{n\}=\sigma\in S^1,\ n\neq o_q, d_q \tag{2.31b}$$

$$\sum_{p\in\widetilde{\mathcal{P}}_{o_q d_q}} f_p^q - \sum_{\substack{p\in\widetilde{\mathcal{P}}_{o_q d_q}:\\ |L(p)|=1}} f_p^q \geq v_q^* \qquad \forall\, q\in\mathcal{Q}:\ o_q d_q = t\in T, L_t=\sigma\in S^1 \tag{2.31c}$$

Transparent networks with multi-hop traffic. The adaptions for $\mathrm{OND}_{\mathrm{tmh}}$ combine those for link and paths flows in (2.29) and (2.30). The flow conservation constraints (2.16a) must be reformulated for an unspecified total number of connections to route as:

$$\sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ o_p=n}} f_p^q - \sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ d_p=n}} f_p^q \quad \begin{cases} \geq \ \ v_q\ , n=o_q\,, \\ \leq -v_q\ , n=d_q\,, \\ =\ \ 0\ , \text{otherwise} \end{cases} \qquad \forall\, q\in\mathcal{Q},\ n\in N \tag{2.32a}$$

For the second condition in (2.28), notice that the connections are not restricted on end-to-end paths in this model. Hence, the model extension reads now:

$$\sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ o_p=o_q}} f_p^q - \sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ n\in N[p]\setminus\{o_p\}}} f_p^q \geq v_q^* \qquad \forall\, q\in\mathcal{Q},\ \{n\}=\sigma\in S^1,\ n\neq o_q, d_q \tag{2.32b}$$

$$\sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ o_p=o_q}} f_p^q - \sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ L(p)\subset L_t}} f_p^q \geq v_q^* \qquad \forall\, q\in\mathcal{Q}:\ o_q d_q = t\in T, L_t=\sigma\in S^1 \tag{2.32c}$$

The total number of connections established for a commodity is expressed in the first term on the left hand sides, summing up the flow on all paths originating at the commodity's source. The constraints guarantee at least the demanded number of connections to survive in case all connections traversing a single node or link are disrupted by a failure. The throughput measurement for nodes in constraints (2.32b) and direct origin-destination links in constraints (2.32c) is modeled as already discussed with the extension (2.26).

DSP **properties.** Altogether, DSP is realized by turning some flow equalities into inequalities and by a particular expression of the protection requirements, which now directly demand for sufficient surviving connections instead of (conversely) limiting node or link throughputs. Thereby, DSP releases the dictates of prescribed connection numbers from bDSP and pDSP, but does not invalidate any bDSP or pDSP routing (for any feasible value of κ_q'), as all these routings satisfy both conditions from (2.28) as well. In fact, the generalization consists of allowing for further routing alternatives. This does not only correspond to adding redundant connections (as long as these do not incur additional costs), but includes also routings that cannot be reduced to connectivity-induced routings with a minimum number of connections, as shown by the following example.

Example. In the exemplary network used before, consider a commodity q between nodes A and B with $v_q = v_q^* = 4$ and $\kappa_q = 3$. Figure 2.6 shows a possible DSP routing with in total seven connections that satisfies both conditions in (2.28). Hence,

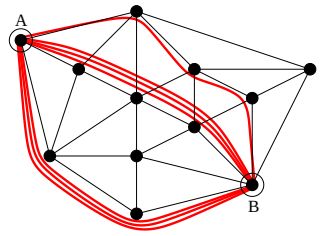


Figure 2.6: *A* DSP *routing for a commodity* q *between nodes* A *and* B *with* $v_q = v_q^* = 4$ *that does not correspond to a* pDSP *routing for any connectivity.*

this routing represents a feasible DSP solution for q. In particular, note that none of these connections is redundant, i.e., can be left out without violating the second condition in (2.28).
Applying pDSP allows to select the explored connectivity $\kappa_q' \in \{2, 3\}$. With $\kappa_q' = 3$, pDSP would ask for a total of $\widetilde{v}_q = 6$ connections. Due to the observed non-redundancy, the depicted DSP routing indeed cannot be reduced to such a solution only by removing of connections. Furthermore, choosing $\kappa_q' = 2$ yields $\widetilde{v}_q = 8$. Here, an arbitrary connection could be added to the routing in Figure 2.6, making it feasible for pDSP parameterized this way. In fact, such an extension to pDSP based on a lower connectivity than explored by DSP is always possible.
Nevertheless, this routing expansion possibility does not mean that any DSP routing can be transformed into one for pDSP at same cost. In our example, assume to consider upgrade planning where only the capacities occupied by the routing illustrated in Figure 2.6 are available for free. Then any extended routing for pDSP with $\kappa_q' = 2$ increases total cost. In this case, the DSP routing provides the only zero cost solution, whereas pDSP has a strictly positive optimum objective value. This scenario demonstrates that the general concept DSP in fact expands the set of feasible routings and cannot be reduced to an equivalent pDSP solution in any case.

DSP **connection number bounds.** The example illustrates the increased variability of DSP routings in comparison to those for bDSP and pDSP and shows that DSP is not restricted to establish a number of connections from the set $\{\widetilde{v}_q^2, \ldots, \widetilde{v}_q^{\kappa_q}\}$. Besides flow conservation equalities, the exclusive use of lower limits for concerted establishment of connections in particular sets, as described in the introduction, leaves room for such alternative routings. This does especially not imply a natural upper bound on the total number of connections routed by DSP for a commodity, as long as no additional cost for capacity consumption is incurred. Nevertheless, except for the single-hop case with end-to-end connection length limitations, it is possible to restrict the connection numbers for DSP to a specific range (with already presented bounds) without ruling out all optimal solutions for an instance.
The following discussion refers always to an arbitrary, but fixed commodity q with parameters v_q and v_q^*. From Lemma 2.19, we know that $\widetilde{v}_q^{\kappa_q}$ for exploiting the maximum connectivity κ_q yields a lower bound on the number of connections in any feasible routing, hence for any DSP routing as well. On the opposite side, such a general bound for *all* optimal solutions cannot be provided, but existence of optimal solutions with a generally bounded connection number is proven by the following proposition.

Proposition 2.21 *There exists always an optimal solution for survivable optical multi-hop network design based on* DSP *such that for each commodity $q \in \mathcal{Q}$ in total at most*

$$\max\{v_q, 2v_q^*\}$$

optical connections are established.

Proof. In the following, we interpret routings always as integer-valued flows and do not matter about a corresponding decomposition into individual optical connections (though referring to the flow value also as number of connections). Hence, reducing a routing does not mean to remove end-to-end connections, but to replace its flow by one with lower flow value. Since such reductions do not allow to control the total length of the finally producible connections, the proof does not apply to single-hop architectures.
Consider an arbitrary optimal problem solution and an arbitrary commodity $q \in \mathcal{Q}$ with $\widehat{v}_q$ connections in total. If $\widehat{v}_q \leq \max\{v_q, 2v_q^*\}$, nothing is to do.
Otherwise, we have a feasible DSP routing with in total $\widehat{v}_q > \max\{v_q, 2v_q^*\}$ connections that occupies the capacities $\widehat{k}_\ell \in \mathbb{Z}_+$ at any link $\ell \in L$ and $\widehat{k}_n \in \mathbb{Z}_+$ at any node $n \in N \setminus \{o_q, d_q\}$. Moreover, let $\Gamma \subset L \cup N \setminus \{o_q, d_q\}$ be an arbitrary generalized o_q, d_q-cut in the network. The routing requirements (2.28) can be mapped onto the cut Γ as follows: Ensure that

$$\left.\begin{array}{l} \text{1. at least } v_q \text{ connections traverse the cut, and} \\ \text{2. at least } v_q^* \text{ connections still traverse the cut when} \\ \quad \text{removing an arbitrary element } \gamma \in \Gamma. \end{array}\right\} \tag{2.33}$$

As long as these conditions hold, we call a cut throughput *feasible*. By application of (an appropriate adaption of) the max-flow min-cut theorem of Ford and Fulkerson (see Appendix A), it is easy to verify that a DSP routing for a commodity q is feasible if and only if the throughput for each generalized o_q, d_q-cut is feasible.
Consider an arbitrary cut element $\gamma_1 \in \Gamma$. The value $\widehat{k}_{\gamma_1}$ corresponds to the number of connections traversing γ_1. If this number can be reduced to $\widehat{k}_{\gamma_1} - 1$ without violating the throughput feasibility of the cut, we are satisfied. Otherwise, such a reduction must violate the second condition in (2.33), since the first condition is satisfied due to the prerequisite $\widehat{v}_q > \max\{v_q, 2v_q^*\}$ and thus in the reduced case still $\widehat{v}_q - 1 \geq v_q$. Hence, there must be another cut element $\gamma_2 \in \Gamma \setminus \{\gamma_1\}$ whose elimination leaves exactly the minimum number of connections traversing the remaining cut elements, i.e., it holds that

$$\sum_{\gamma \in \Gamma \setminus \{\gamma_2\}} \widehat{k}_\gamma = v_q^* \tag{2.34}$$

such that a throughput reduction in γ_1 would yield the described violation. As the total cut throughput is clearly at least $\widehat{v}_q$, we then obtain

$$\begin{aligned} & \sum_{\gamma \in \Gamma} \widehat{k}_\gamma = \widehat{k}_{\gamma_2} + \sum_{\gamma \in \Gamma \setminus \{\gamma_2\}} \widehat{k}_\gamma \overset{(2.34)}{=} \widehat{k}_{\gamma_2} + v_q^* \geq \widehat{v}_q > \max\{v_q, 2v_q^*\} \\ \Longrightarrow \quad & \widehat{k}_{\gamma_2} > \max\{v_q, 2v_q^*\} - v_q^* = \max\{v_q - v_q^*, v_q^*\} \geq v_q^* \end{aligned} \tag{2.35}$$

for the number of connections traversing the cut element γ_2. This number now can be reduced by one without violating the throughput feasibility of Γ. The first condition holds as argued above, and the second condition holds because elimination of any cut element from $\Gamma \setminus \{\gamma_2\}$ leaves γ_2 in the remaining cut with the reduced throughput of $\widehat{k}_{\gamma_2} - 1 \geq v_q^*$ according to (2.35). Moreover, removing γ_2 leaves v_q^* throughput due to (2.34) as no other element's throughput has been reduced.
As a result, each generalized o_q, d_q-cut Γ contains at least one element with throughput reducible by at least one without violating (2.33). Using the reduction potentials as capacities, the max-flow min-cut theorem of Ford and Fulkerson implies existence of a flow from o_q to d_q with value (at least) one using only links and nodes with reducible throughput. Removing this flow from the initial routing leaves a flow that, by construction, still satisfies (2.33) for each generalized o_q, d_q-cut and thus, by the equivalence observed above, also conditions (2.28) for a corresponding routing. Hence, the reduced flow represents a feasible DSP routing for q, too.
The described reduction works whenever the initial routing connection number is $\widehat{v}_q > \max\{v_q, 2v_q^*\}$. Hence, iterative application finally generates a routing for q with the desired property $\widehat{v}_q \leq \max\{v_q, 2v_q^*\}$. The same procedure can be carried out for each commodity $q \in \mathcal{Q}$ and generates in the end, as no additional cost have been incurred, another optimal solution whose existence was claimed. ■

As a consequence, the number $\widehat{v}_q$ of connections in DSP routings for each commodity $q \in \mathcal{Q}$ can be bounded by

$$\widetilde{v}_q^{\kappa_q} \leq \widehat{v}_q \leq \max\{v_q, 2v_q^*\}$$

for any survivable optical multi-hop network design instance. These bounds can be explicitly added to the models, for instance when introducing $\widehat{v}_q$ as auxiliary variables for all $q \in \mathcal{Q}$ defined by

$$\widehat{v}_q = \begin{cases} \sum\limits_{\ell \in L(o_q)} f^q_{\ell o_q} \ , & \text{for } \mathrm{OND}^{\mathrm{l}}_{\mathrm{opq}}, \\ \sum\limits_{p \in \mathcal{P}_{o_q d_q}} f^q_p \ , & \text{for } \mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}, \\ \sum\limits_{\substack{p \in \widetilde{\mathcal{P}}: \\ o_p = n}} f^q_p \ , & \text{for } \mathrm{OND}_{\mathrm{tmh}}. \end{cases}$$

2.3.2.4 Further variations

The protection scheme DSP and its variants have been introduced for the particular setting of survivable optical network design underlying this thesis. However, DSP provides a general survivability concept (in different specifications) with the characteristic feature of exploiting meshed network connectivity for routing diversification in order to limit the impact of failures and thus to reduce the required spare capacities. This concept is not bounded to a particular technology or architecture and open to further extensions, adaptions, and variations. A thorough discussion of such possibilities indeed goes beyond the scope of this thesis and is left for further work. Here, we just exemplify the capabilities by describing variations of DSP for selecting alternative sets of operating states that are to be involved for planning.

As presented, DSP has been designed for protection against all single link or node failures, but can be easily adapted to variations of this assumption. Some of these alternatives are briefly discussed. For this, we restrict on an explanation for the basic concept bDSP, as the adaptions can be carried over straightforward to the generalized schemes as well.

Restriction to link failures. In principle, bDSP does not presume any particular property of the failure states to consider. Hence, a restriction to link failures is straightforward by substituting connectivity with the weaker notion of link-connectivity throughout the description. The parameter κ'_q then refers to the link-connectivity to exploit. This way, the very same principle of bDSP applies to the restricted case of link protection. Moreover, the link-connectivity for a node pair is not less than the connectivity, but can indeed be higher. In this case, the total number of connections required for such a commodity can be reduced further according to Lemma 2.20.

Selectable failure states. Similarly, it is not necessary to take a failure of any link or node into account. Some elements of the physical topology can be declared as unfailing, making them present in each residual network. For selective failure states, we have to adapt the notion of connectivity by accounting for the maximum number of paths between a node pair which are pair-wise disjoint *on potentially failing links and nodes*. This number can be determined for each commodity by a max-flow computation where each unfailing element gets unlimited capacity, whereas all other links and nodes have a capacity of one. If the maximum flow is unlimited, there exist entirely unfailing routing paths from origin to destination of a commodity. In this case, the routing can be restricted to such paths, and no backup connections are required at all. Otherwise, bDSP is used the very same way as before, applying the modified connectivity values.

Multiple failures. The last issue concerns protection against multiple failures. Due to the diversified routing, bDSP already provides a certain extra protection for such situations. For example, the routing shown in Figure 2.6 guarantees at least one surviving connection whenever at most two links or nodes (except for origin and destination) fail. For the same set of operating states, the routing in Figures 2.5(a) on page 77 even leaves at least two connections unaffected, i.e., $v_q^* = 2$ is yet provided for up to two failing links or nodes. However, these implicit guarantees for multiple failures are irregular and depend mainly on the arbitrarily selected routing, which has originally been carried out for single element protection only. An explicit integration of multiple failure states for bDSP with a prescribed number of protected connections can be done as follows.
Consider a commodity q with connectivity $\kappa_q \geq 2$ asking for v_q connections of which a number of v_q^*, $0 < v_q^* \leq v_q$, have to be protected against all failures of at most k links or nodes, where $1 \leq k < \kappa_q$. With bDSP exploiting connectivity κ'_q, $k < \kappa'_q \leq \kappa_q$, a total number of

$$\widetilde{v}_q^k := \max\left\{v_q, \overline{v}_q^k + (k-1)\left\lceil \frac{\overline{v}_q^k}{\kappa'_q - k + 1}\right\rceil\right\} \text{ with } \overline{v}_q^k := \left\lceil \frac{\kappa'_q - k + 1}{\kappa'_q - k} v_q^*\right\rceil \quad (2.36)$$

connections (including both working and backup connections again) is routed with a diversification parameter of

$$\widetilde{\delta}_q^k := \frac{1}{\widetilde{v}_q^k} \left\lfloor \frac{\widetilde{v}_q^k - v_q^*}{k} \right\rfloor . \tag{2.37}$$

Obviously, both definitions (2.36) and (2.37) applied to single link or node failures ($k = 1$) reduce to those presented originally in (2.20) and (2.21), respectively, with parameterized connectivity κ_q'.
Again, we formally prove existence and suitability of routings for this generalization of bDSP.

Proposition 2.22 *For a commodity q with connectivity $\kappa_q \geq 2$ demanding for $v_q > 0$ connections of which v_q^*, $0 \leq v_q^* \leq v_q$, are to be protected against all multiple failures of up to k links or nodes from $L \cup N \setminus \{o_q, d_q\}$ with $1 \leq k < \kappa_q$, there exist routings with in total $\widetilde{v}_q^k$ connections according to* (2.36) *and diversification parameter $\widetilde{\delta}_q^k$ defined by* (2.37) *that satisfy all traffic and survivability requirements, i.e., provide at least v_q connections in normal operation and still at least v_q^* connections in each failure state.*

Proof. Clearly, $\widetilde{v}_q^k \geq v_q$ holds for normal operation by (2.36). In any considered failure state, at most k links or nodes break down simultaneously, and (in the worst case) the corresponding sets of affected connections can be mutually disjoint. By diversification, however, each failure cannot disrupt more than $\widetilde{\delta}_q^k \widetilde{v}_q^k$ connections, and thus the number of surviving connections is at least

$$\widetilde{v}_q^k - k\widetilde{\delta}_q^k \widetilde{v}_q^k \overset{(2.37)}{=} \widetilde{v}_q^k - k \left\lfloor \frac{\widetilde{v}_q^k - v_q^*}{k} \right\rfloor \geq \widetilde{v}_q^k - k\frac{\widetilde{v}_q^k - v_q^*}{k} = v_q^*$$

providing the demanded protection.
It remains to show that the diversification parameter $\widetilde{\delta}_q^k$ is in fact feasible and enables generation of appropriate routings. As simple observations, $\widetilde{\delta}_q^k \widetilde{v}_q^k$ is integer by (2.37), and furthermore

$$\widetilde{\delta}_q^k \leq \frac{1}{\widetilde{v}_q^k} \frac{\widetilde{v}_q^k - v_q^*}{k} \leq \frac{1}{\widetilde{v}_q^k} \frac{\widetilde{v}_q^k}{k} = \frac{1}{k} \leq 1 .$$

Showing that $\widetilde{\delta}_q^k$ is also sufficiently large is more difficult. At first, observe for $\overline{v}_q^k$ according to (2.36) that

$$\begin{aligned}
& \overline{v}_q^k = \left\lceil \frac{\kappa_q' - k + 1}{\kappa_q' - k} v_q^* \right\rceil \geq \frac{\kappa_q' - k + 1}{\kappa_q' - k} v_q^* \\
\Longrightarrow \quad & (\kappa_q' - k)\overline{v}_q^k = (\kappa_q' - k + 1)\overline{v}_q^k - \overline{v}_q^k \geq (\kappa_q' - k + 1) v_q^* \\
\Longrightarrow \quad & \overline{v}_q^k - v_q^* \geq \frac{\overline{v}_q^k}{\kappa_q' - k + 1} \\
\Longrightarrow \quad & \overline{v}_q^k - v_q^* \geq \left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1} \right\rceil
\end{aligned} \tag{2.38}$$

since the left inequality term is integer. Then we obtain

$$\begin{aligned}
\frac{\widetilde{v}_q^k - v_q^*}{k} &\overset{(2.36)}{=} \frac{1}{k}\left(\overline{v}_q^k + (k-1)\left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil - v_q^*\right) \\
&= \frac{\overline{v}_q^k - v_q^*}{k} + \frac{k-1}{k}\left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil \\
&\overset{(2.38)}{\geq} \frac{1}{k}\left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil + \frac{k-1}{k}\left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil = \left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil
\end{aligned}$$

and we can conclude now by the integer right inequality term that also

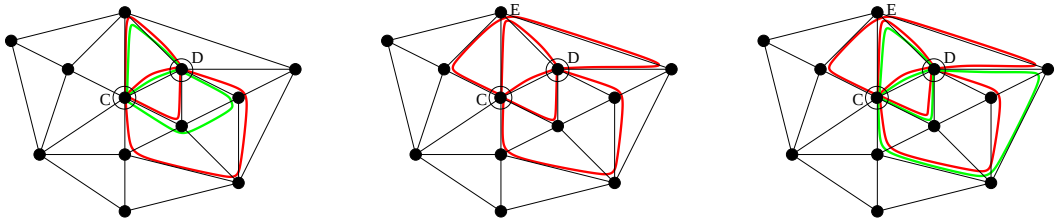
$$\left\lfloor \frac{\widetilde{v}_q^k - v_q^*}{k}\right\rfloor \geq \left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil . \tag{2.39}$$

After this preparation, we end up with

$$\begin{aligned}
\kappa_q' \ \widetilde{\delta}_q^k \ \widetilde{v}_q^k &\overset{(2.37)}{=} \kappa_q'\left\lfloor \frac{\widetilde{v}_q^k - v_q^*}{k}\right\rfloor \overset{(2.39)}{\geq} \kappa_q'\left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil \\
&= (\kappa_q' - k + 1)\left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil + (k-1)\left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil \\
&\geq \left\lceil (\kappa_q' - k + 1)\frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil + (k-1)\left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil \\
&= \overline{v}_q^k + (k-1)\left\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\right\rceil \overset{(2.36)}{=} \widetilde{v}_q^k
\end{aligned}$$

which allows to deduce the desired inequality $\widetilde{\delta}_q^k \geq \frac{1}{\kappa_q'}$. This inequality together with the integrality of $\widetilde{\delta}_q^k \widetilde{v}_q^k$ yields existence of feasible routings in exactly the same way as in Lemma 2.18, completing the proof. ■

Unfortunately, this time the proof is more technical and less self-explanatory, but an explicative derivation of the complex second term in (2.36) can be provided as well. Assume again that the routing shall use disjoint paths only. Whenever $k-1$ of these paths are disrupted, the connections routed on the remaining $\kappa_q' - k + 1$ disjoint paths are carried out as in the single failure case, which yields (as first summand) the term for $\overline{v}_q^k$ to protect the traffic against a single further failure. This way, v_q^* connections would survive k failures in total. When distributing $\overline{v}_q^k$ connections onto $\kappa_q' - k + 1$ disjoint routes in a best possible balanced way, each route carries either $\lfloor \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\rfloor$ or $\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\rceil$ connections. For the already disrupted $k-1$ paths, we have to take the higher number into account, which is exactly expressed in the second summand in (2.36). As a result, we get the full routing best balanced as well, such that it does not depend on which k paths are disrupted. In any case, still sufficiently many connections will survive. The diversification parameter is finally set such that according to (2.39) the bound $\widetilde{\delta}_q^k \widetilde{v}_q^k$ still allows to route the potentially higher connection number $\lceil \frac{\overline{v}_q^k}{\kappa_q' - k + 1}\rceil$ through any link or node.



(a) $k = 2$, $\kappa'_q = \kappa_q = 4$. (b) $k = 2$, $\kappa'_q = 5$. (c) $k = 3$, $\kappa'_q = 5$.

Figure 2.7: *b*DSP *routings for commodity* q with $v_q = 3 = v_q^*$ between nodes C and D, *each with a minimum total number of connections providing full protection for multi failure situations and selective failure states.*

Most observations for single failure bDSP carry over to the multiple failure generalization. For instance, routings are not restricted to exclusive use of disjoint paths, and it can also be shown (again by reduction to the single failure case) that the connection number (2.36) for exploiting the full available connectivity $\kappa'_q = \kappa_q$ states a minimum for the number of connections in any feasible commodity routing satisfying all requirements. However, since multiple failures are not further considered in this thesis, we omit a detailed discussion of these issues and finish with an example illustrating the bDSP mechanism for multiple failures and selective failure states.

Example. Continuing the example series, we reuse the former network and consider a commodity q between the four-connected nodes C and D. Let $v_q = 3$ demanding full protection $v_q^* = 3$ against up to $k = 2$ failures and exploit the full connectivity $\kappa'_q = \kappa_q = 4$. Then, we find $\overline{v}_q^k = \lceil \frac{3}{2} 3 \rceil = 5$ which yields $\widetilde{v}_q^k = 5 + \lceil \frac{5}{3} \rceil = 7$ and $\widetilde{\delta}_q^k = \frac{2}{7}$, hence at most $\widetilde{\delta}_q^k \widetilde{v}_q^k = 2$ connections can traverse each link or node except for C and D. When using only disjoint routing paths such as those displayed in Figure 2.5(a), a distribution of 1, 2, 2, and 2 connections on these paths must be established, whereas Figure 2.7(a) shows another possible routing for this commodity.
For a second case, assume further that failures of node E are not to be considered, i.e., this node will never fail. Taking this into account, nodes C and D become even five-connected. For the same commodity with $\kappa'_q = 5$, one obtains $\overline{v}_q^k = \lceil \frac{4}{3} 3 \rceil = 4$, and in total $\widetilde{v}_q^k = 4 + \lceil \frac{4}{4} \rceil = 5$ connections suffice. Since $\widetilde{\delta}_q^k = \frac{1}{5}$, all connections in any feasible routing must be mutually disjoint except for nodes C, D, and E, as the example routing in Figure 2.7(b) illustrates.
Let now $k = 3$. Here, we get $\overline{v}_q^k = \lceil \frac{3}{2} 3 \rceil = 5$ and further $\widetilde{v}_q^k = 5 + 2\lceil \frac{5}{3} \rceil = 9$ with $\widetilde{\delta}_q^k = \frac{2}{9}$. Similar to the very first example, a routing of 1, 2, 2, 2, and 2 connections on a appropriate disjoint path set (e.g., as that in Figure 2.7(b)) would do, and a further alternative is depicted in Figure 2.7(c).

2.3.3 Discussion and comparison

In this section, we close the presentation of DSP by a discussion of some properties in view of practical application, also in comparison to existing path protection concepts which have been described in Section 1.3. Besides total connection numbers and incurred cost, important comparison criteria for survivability schemes in prac-

	v_q	1	2	3	4	5	6	7	8	9	10	15	20	25	35	50
$\frac{1}{2}$-protected																
1+1 protection		1	1	2	2	3	3	4	4	5	5	8	10	13	18	25
max-DSP	$\kappa_q = 2$	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0
	$\kappa_q = 3$	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	$\kappa_q = 4$	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\frac{2}{3}$-protected																
1+1 protection		1	2	2	3	4	4	5	6	6	7	10	14	17	24	34
max-DSP	$\kappa_q = 2$	1	2	1	2	3	2	3	4	3	4	5	8	9	13	18
	$\kappa_q = 3$	1	1	0	1	1	0	1	1	0	1	0	1	1	1	1
	$\kappa_q = 4$	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
$\frac{3}{4}$-protected																
1+1 protection		1	2	3	3	4	5	6	6	7	8	12	15	19	27	38
max-DSP	$\kappa_q = 2$	1	2	3	2	3	4	5	4	5	6	9	10	13	19	26
	$\kappa_q = 3$	1	1	2	1	1	2	2	1	2	2	3	3	4	6	7
	$\kappa_q = 4$	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
full-protected																
1+1 protection		1	2	3	4	5	6	7	8	9	10	15	20	25	35	50
max-DSP	$\kappa_q = 2$	1	2	3	4	5	6	7	8	9	10	15	20	25	35	50
	$\kappa_q = 3$	1	1	2	2	3	3	4	4	5	5	8	10	13	18	25
	$\kappa_q = 4$	1	1	1	2	2	2	3	3	3	4	5	7	9	12	17

Table 2.1: *Number of backup connections established by 1+1 path protection and by max-*DSP *exploiting different connectivities for a series of demand values and some selected protection levels specified as demand value fraction (where resulting fractionals are always rounded up). The 1+1 path protection entries also equal the number of protected connections.*

tice concern also operational issues such as recovery time or (expected) connection availabilities.

Connections and cost. Basically, the DSP schemes are designed to exploit opportunities of the structure of meshed networks in order to save in the number of backup connections to establish for provision of the requested protection. According to Lemma 2.19, this number is in fact minimized when fully exploring the available connectivity, as proposed by max-DSP. For illustration of the potential backup connection number reductions, Table 2.1 lists such minima for a series of demand values and protection levels in comparison to the corresponding numbers generated by 1+1 path protection. The latter values correspond also to the number of protected connections at each level, since 1+1 path protection establishes an individual backup connection for each protected connection. So, the overview demonstrates how many backup connections suffice to protect the specified working connections when both are appropriately routed in a meshed network. Note further that 2-DSP is equivalent to max-DSP with connectivity two. Even for this lowest connectivity, savings in backup numbers are possible as long as partial protection is requested. Only with full protection, 2-DSP and 1+1 path protection coincide in the numbers, but 2-DSP additionally provides more possible routings for selection.

With lower total connection numbers, bDSP attempts also to reduce spare capacity consumption. Depending on the selected parameters, the diversification requirements indeed can impose to use long routing paths whose additional capacity occupation can overcompensate the savings by establishing less connections in total. Therefore, the generalized schemes pDSP and DSP have been introduced to gain more flexibility in the routing selection. In view of the solutions, one easily observes that 2-DSP, max-DSP, or any other individual bDSP scheme (which are not comparable against each other) forms a special case of pDSP, and furthermore, each solution of pDSP is also a feasible solution of the general DSP scheme. As a consequence, the schemes in this order will yield non-increasing optimal network design costs, or equivalently, the optimal solution cost for a more general scheme provides a lower bound on the network design costs for any more special scheme.
Similar relations regarding the (optimal) cost of corresponding network designs can also be derived in comparison to some known concepts. For full protection, 1+1 path protection establishes the same total number of connections as 2-DSP, but has restricted routing possibilities due to individual disjointness restrictions for each pair of a protected connection and its dedicated backup connection. Hence, the optimal cost with 2-DSP is a lower bound for any 1+1 path protection solution, and so is the optimum for DSP as well. This carries over to any partial protection case, too. In such a setting, 1+1 path protection can use more connections in total for a commodity than 2-DSP, but each such 1+1 routing can be turned into a feasible one for 2-DSP by appropriately removing some of the connections established by 1+1 path protection. Since the solution cost does not increase when connections are removed, the lower bound relations remain valid.
1:1 path protection is more similar to 2-DSP, since backup connections established for the protected connections of a commodity are already provided to the unprotected best-effort connections. Nevertheless, such routings also form special cases of 2-DSP routings, again possibly after removal of some connections, and thus the solution costs are lower bounded by 2-DSP optimal solution costs as well. When M:N path protection is used to provide the requested survivability for specified connections, M and N have to be set individually for each commodity q according to the values of v_q and v_q^*. Then, M:N path protection routings can be interpreted as a special cases of 1:1 path protection routings, inheriting the same bounding relations.
On the other side, DSP can be interpreted as a special variant of shared path protection. In fact, DSP allows to share backup connections in an end-to-end manner among the protected connections of a commodity, where the assignment of the particular backup for a protected connection depends on the failure state and all backup connections can serve unprotected best-effort traffic, too. Hence, DSP routings with this restricted kind of sharing form special case solutions for any shared path protection scheme that allows to specify individual backup connections for each failure state and offers shared capacities to accommodate best-effort traffic.

DSP **recovery mechanism.** From an operational point of view, the recovery by DSP in case of a failure depends on the affected connections. If only connections carrying unprotected traffic fail, no recovery is needed at all. Otherwise, if protected working connections are disrupted, recovery requires to reorganize (a part of) the commodity's routing by reallocating protected traffic to surviving connections. These backup connections may be occupied by best-effort traffic, which has then first

to be dropped. After that, a sufficient number of connections is offered to carry the protected traffic. Since all these connections are preestablished, the recovery process needs just to redirect each disrupted and protected channel at the origin node to a surviving connection (and a corresponding operation at the receiving destination node). Neither a routing computation has to be performed nor new connections have to be set up on request. So, the DSP recovery mechanism restricts to failure detection, propagation, and switching operations in the endnodes of affected commodities, providing a fast mechanism similar to that for M:N path protection.
Basically, DSP does not differentiate working and backup connections, but guarantees only for connection numbers to survive each (preplanned) failure situation. The operator can arbitrarily distribute the normal operation traffic over the set of routed connections, e.g., meeting second order goals such as providing the shorter routes to working connections. Alternatively, this assignment can also be carried out such that the maximum number of protected working connections traversing a link or node is minimized. This limits the damage by individual failures and can fasten DSP recovery (in the mean). A further alternative is to minimize the expected recovery time by an appropriate distribution of protected connections when having individual link and node failure probabilities, which are also used for evaluation of connection availabilities.

Connection availabilities. Given a failure probability for each link and node, the *availability* of a connection is defined as the probability that none of the traversed links or nodes fails. The endnodes of a commodity are often excluded for availability evaluations. In case of protected connections, the availability of corresponding backup connections has to be involved, too. With dedicated path protection, availabilities can be easily analyzed by evaluating the probability that either the working connection or its assigned backup connection does not fail. Such an analysis indeed becomes more difficult when sharing of backup resources occurs as in case of shared path protection or DSP.
In Hülsermann et al. [70], we report on a study on the mean availabilities of (protected and unprotected) connections by application of different survivability schemes, including unprotected shortest path routings, 1+1 path protection carried out by Suurballe/Dijkstra routings as well as in an cost optimizing way, 2-DSP, max-DSP, DSP, and shared path protection by use of a heuristic approach. For the DSP schemes, we developed a combinatorial method for exact computation of mean availabilities. Note that such a computation implicitly involves the probability of any potential failure situation, including any combination of multiple failures. For the other schemes, standard (exact or approximate) evaluation methods have been used.
In a nutshell, the study results show on the one hand that for protected connections, DSP provides a mean availability comparable to that of 1+1 path protection. The availability of unprotected connections indeed suffers from the applied routing diversification which tends to establish (substantially) longer connections than shortest path or shortest cycle routings (and disruption probability grows with the number of hops as well as with the total kilometer distance due to length-dependent link failure probabilities). On the other hand, the DSP solutions are considerably cheaper than those for 1+1 path protection. For relating availabilities and costs, we use the so-called *gain* defined as increase in mean availability over increase in cost both in comparison to the unprotected solution. In this measure, DSP clearly outperforms

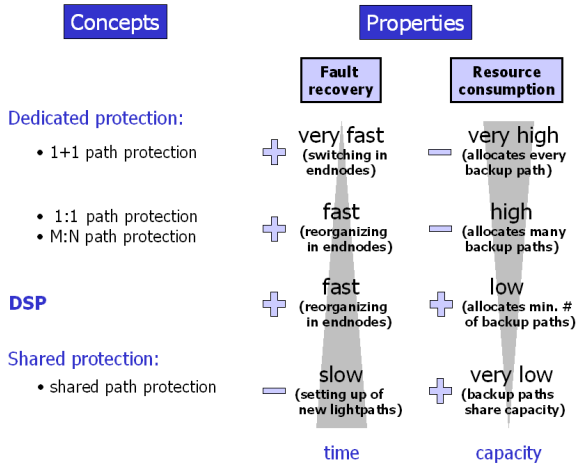


Figure 2.8: DSP *in relation to other path protection schemes with respect to (theoretical) recovery time and spare resource consumption.*

1+1 path protection. This holds also for the comparison to shared path protection as long as the traffic is not fully protected. Otherwise, shared path protection yields a slightly higher gain by providing further network cost savings. These savings are achieved with lower backup capacity consumption by sharing resources whenever possible, which in turn slows down the recovery process and complicates network management.

Categorization embedding. As a summary, DSP can be included into the scheme from Figure 1.14 on page 29 as shown in Figure 2.8. Regarding general survivability principles, DSP constitutes a mixing of dedicated and shared protection. On the one hand, backup connections are set up for each commodity individually. Thus the occupied spare capacities are dedicated to the commodity. On the other hand, the provided protection is shared among the working connections within the commodity. This way, DSP combines advantages of both dedicated and shared protection and provides another compromise between fast recovery and spare capacity requirements.

Chapter 3

Solving optical network design

In the previous chapter, we show that cost-oriented optical network design can be formulated in an adequate 'language' for mathematical optimization, as integer linear programs. By the derived models for different architecture scenarios, we also gain insight into the problem's structure. Such a modeling indeed is only the first step towards finding good network designs.

In this chapter, we present our solution approach for survivable optical network design. We strive for an exact approach, i.e., finding optimal solutions. Little experience in the field suffices to recognize that the comprehensive integer linear programs are too complex for a direct exact approach. Therefore, we make use of a well-known helpful idea: divide-and-conquer. We decompose the problem in a suitable way which makes solutions tractable, but does not sacrifice too much solution quality.

On passage from modeling to algorithms, we begin with a brief overview of related work in the literature. A refined task structuring helps in categorizing the approaches.
Next, we explain our solution approach, as preliminarily introduced in Zymolka et al. [178]. We motivate the applied decomposition and discuss its properties. The approach subdivides the task into a dimensioning and routing subproblem and a wavelength assignment subproblem, which are solved subsequently. This provides a common solution framework for all scenarios.
In Section 3.3, we present our method for the first step, the dimensioning and routing task, whereas the next chapter is dedicated to wavelength assignment with converters as second step. For dimensioning and routing, we first unify the subproblem for all scenarios. The resulting (core) problem is then solved in a three-step procedure, consisting of preprocessing, mathematical optimization, and postprocessing.

3.1 Related work

In the literature, optical network design has received growing attention with the migration to WDM networks in the early 1990s. At this time, design investigations were naturally dedicated to opaque networks. As the idea of optical switching came up in technician's minds due to its operational benefits, the development of appropriate devices was initiated. Consequently, research to investigate transparent networks was started as well. In lockstep with technological progress (or its announcement), planning tasks have been considered for innovative transparent architectures expected to arise. Meanwhile, plenty of papers and books on operation, configuration, and design of optical networks in any architecture have been published. Before we subsume work related to this thesis, we prepare the discussion by a refinement of the task structure.

Task structure refinement. In Section 1.5, we introduced the design task for optical networks and identified two main configurations involved: the hardware configuration and the lightpath configuration. A closer look on the mathematical models derived in the previous chapter reveals that in particular the configuration of lightpaths gains complexity with advanced technology. A routing of the lightpaths is indispensable in any case, whereas the need for an additional wavelength assignment is a characteristic feature of networks with transparent nodes. This gives rise to refine the structuring of the comprehensive task, as shown in Figure 3.1. We propose a classification of the many decisions to take within planning into three main subtasks:

- **hardware dimensioning**
  including
  - topological decisions (setup of links and nodes),
  - capacity decisions (sufficient provision of all types), and
  - hardware decisions (device selection);
- **lightpath routing**
  including
  - routing path selection (single-/multi-hop),
  - survivability guarantees according to a (predefined) underlying concept;
- **wavelength assignment**
  for conflict-free operation of the lightpaths to establish.

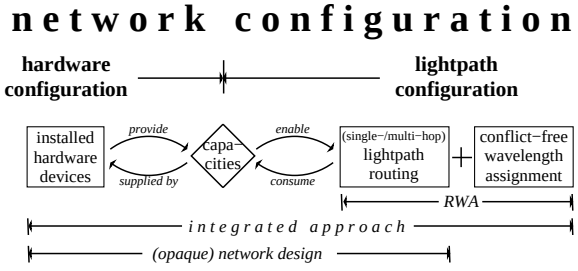


Figure 3.1: *Refined structure of the network design task.*

In this light, the design of opaque optical networks combines dimensioning and routing, whereas wavelength assignment is of minor relevance and can be easily accomplished afterwards. For the design of transparent optical networks, all three aspects state non-trivial and interdependent problem parts, with an even more sophisticated routing task in the multi-hop case. The architecture at hand determines also which hardware module sorts are involved for dimensioning. The refined task structuring provides a base for a categorization of related work in the literature.

Literature. We focus on literature that deals with modeling and solving of network planning tasks rather than those on technical issues or network operation, including survivability. For these topics, we refer to Chapters 1 and 2 and literature cited there. Moreover, the following discussion does not claim completeness as survey for an already large and permanently growing field of literature, but restricts to the most relevant work for this thesis.
For planning optical networks, most research concentrates around the new property that optics brings into play: the wavelengths. In fact, a large portion of the work has been carried out for the particular subproblem of Routing and Wavelength Assignment (RWA), i.e., restricting on the lightpath configuration as shown in Figure 3.1, while dimensioning or, more generally, (a detailed figure of) the hardware configuration received substantial less attention. We refer to any approach including all three subtasks as *transparent optical network design*, while those without assignment of wavelengths are denoted as *network design*, which includes opaque optical and non-optical networks.

Network design. An overview on classical, though still actual network design problems and associated literature is provided by Minoux [120], Yuan [175], and van Hoesel [161]. For the core task of capacity installation (or upgrade) against cost to establish multicommodity flows, fundamental work [by use of mathematical programming] has been carried out, for instance, by Bienstock et al. [19] and Günlük [63]. The extension to survivable networks is studied in Grötschel et al. [59], Dahl and Stoer [36], and Bienstock and Muratore [20]. Particular application in telecommunications is discussed in Alevras et al. [3, 4]. In context of planning SDH networks as predecessor of optical networks, further developments are provided in Wessäly [166], including alternative survivability schemes and capacity models which enables an application for various architectures. With slight adaptions, this approach suits already well for opaque optical network design, too, where distinction of wavelengths does not matter. This indeed changes with the arise of optical transparency.

Routing and Wavelength Assignment (RWA). For the design of networks, the integration of transparent optical nodes brings up lightpaths as new characteristic and adds wavelength assignment as indispensable problem part. To approach this extension, a major part of work in the literature deals with the subproblem of Routing and Wavelength Assignment (RWA), also referred to as logical topology design, which corresponds to the traffic (re-) engineering subtask. While switching capacities are often neglected completely, transmission capacities are typically fixed or unlimited both in the number of fibers per link and in the numbers of wavelengths per fiber, using a non-fixed parameter as objective (to minimize). Moreover, the demands are considered either static or dynamic. We omit work on dynamic traffic, yielding questions of different type like call admission or blocking prevention,

and rather focus on RWA with static traffic which forms a subtask of our problem. Comprehensive surveys on RWA approaches are given by Dutta and Rouskas [40] and Zang et al. [177], and we just subjoin selected recent works on this topic, postponing those more close to wavelength assignment to the next chapter.
In contrast to many approaches following (decomposing) multi-step procedures, Jaumard et al. [74, 76, 75] study various integer linear programming formulations for joint RWA without regeneration and conversion, with special focus on column generation formulations in Jaumard et al. [77]. They differentiate in asymmetric and symmetric traffic and several objectives, including minimizing the number of required wavelengths (per fiber in multi-fiber networks), minimizing network load, and maximizing the number of accommodated connections. As result, they report that the formulations are of equivalent quality regarding linear relaxation bounds, but show (significant) differences in terms of size and tractability.
While RWA is most often studied without wavelength conversion capabilities, an exception is provided by Coudert and Rivano [34] who consider RWA with prespecified link capacities and converter placements. The problem to find a feasible (unprotected) routing for all connection requests without exceeding wavelength and conversion capacities is transformed into an integer multicommodity flow problem and solved by LP-based heuristics, based on randomized rounding techniques. A similar approach is also provided by Ozdaglar and Bertsekas [134] for RWA with non-bifurcated unprotected routing in three optical network architectures: opaque, transparent, and hybrid networks with both types of nodes. In addition, they discuss methods to abet getting integer solutions of relaxed problems by application of special objectives and penalty functions.
Finally, we remark two papers on uniform fiber spectra RWA without conversion, but extended in direction of hardware integration. Kennington et al. [86] consider such a problem including a selection of nodes where photonic switches are placed to interconnect layered subnetworks on disposition, too. For minimizing cost incurred by the switches and the selected layered subnetworks, they propose an integer linear programming model and a four-step procedure of subsequent heuristics to find a solution. Nomikos et al. [130] consider two variants of RWA with variable link capacities, either fixing the capacities and minimizing the number of required wavelengths, or fixing the number of wavelengths per fiber and minimizing link-individual costs for the required fibers to install. For both problems (in further variants), they propose exact or constant-factor approximation algorithms for topologies restricted to chain, ring, star, and spider networks.

Transparent optical network design. In the studies on RWA, objectives like minimizing the number of wavelengths, fibers, or similar entities can already be interpreted as an elementary form of capacity dimensioning. However, we refer by transparent optical network design to cases where a more detailed figure of hardware devices and their properties is taken into account. Such approaches indeed often consider different settings regarding architecture, technology, devices, or traffic types. A survey on such problems and corresponding literature can be also found in van Hoesel [161], next extended by further recent work closest to our perspective. Brunetta et al. [25] study transparent optical network design under cost including different WDM systems and selective opaque nodes, but restricted to non-bifurcated and unprotected routings. For the joint dimensioning, routing, and wavelength as-

signment problem, they propose a binary linear programming model (and a variant with incremental link capacities) using path variables for small sets of preselected routing paths on disposition, i.e., without column generation. By adding strengthening inequalities, a cutting plane algorithm is derived for solving the problem. The paper is also part of the PhD thesis by Yuan [176] which contains some further studies on general network design by integer linear programming approaches.
Melián et al. [116, 117] study upgrade planning of opaque optical networks with uniform fibers, WDM systems, and (opaque) OXCs to meet varying demands by non-bifurcated unprotected routings. For this, the proposed integer linear program contains also path variables for preselected path sets (using k-shortest paths algorithms). They develop a metaheuristic solution method, basically composed of tabu search and scatter search, and compare it against another heuristic (in [117]) and solutions of the integer linear program (in [116]) which models in fact a subproblem (due to the restricted path sets).
Birkan et al. [22, 21, 23] consider the design of optical networks under cost with special emphasis on a detailed figure of signal propagation equipment on links, whereas fiber and switch costs are neglected. In [21], several protection schemes are compared, whereas [22] deals with demand uncertainty. Their evaluation of opaque and transparent architectures in [23] reveals the latter to be more cost-efficient under their assumptions.
Belotti [13] studies integer linear programming approaches for three (sub-)problems of optical network design. Besides shared protection schemes and integration of limited range converters, an interesting variation consists in the cost-based design with non-linear node cost (depending on nodal degrees).
More similar to our setting, optical networks with selective o-e-o regenerations have been recently studied as *translucent networks*. In this context, Morea and Poirrier [122] analyze potential cost savings with special focus on optical reach and thereby involved devices (and costs). They use simulations based on shortest path routing and evaluate the benefit of a translucent architecture.

Summarizing, network planning shows up in the literature with manifold settings and alternative layouts which have been studied to various extents. As far as transparent optical networks are concerned, it can be observed that an integrated handling of wavelengths plays typically a major role in these studies. Around this characteristic feature, further aspects like hardware configurations are investigated much less accurate, if at all. For cost-oriented network design, however, hardware expenses are the sole objective driver and thus should be accounted for as detailed as possible. To this end, we follow a different solution approach.

3.2 Solution approach

A drawback of the accurate integer linear programs presented in the preceding chapter is their solvability. Experiments have shown that even for moderate problem sizes, the programs are computationally intractable for state-of-the-art mixed-integer programming solvers, in particular for the transparent scenarios. The structural analysis in Section 2.2 indicates a main reason for this: the inclusion of an integer multicommodity flow problem which models the lightpath routing as well as a

coloring-like problem for the wavelength assignment. The first subtask has already been identified to be $\mathcal{NP}$-hard in Lemma 2.13. A complexity grading for wavelength assignment with fixed routing is given in the following chapter, finding this subtask to be $\mathcal{NP}$-hard as well. Since both problems can be isolated as special case of transparent optical network design, their combination makes anything but a simpler task. In practice, the number of converters at the nodes is usually not bounded as already remarked in Section 2.2, and we focus on the setting without these bounds in the sequel. So, we relax the corresponding conditions (2.12c) and (2.12d) in all models and continue with the remaining wavelength converter model (2.12a), (2.12b), and (2.12e) referred to as (2.12) in what follows. This inconspicuous variation in fact has a fundamental impact. It allows to decompose the task into two subproblems such that at least the two $\mathcal{NP}$-hard special cases are separated. This decomposition is first presented for $\mathrm{OND}_{\mathrm{tsh}}$ without survivability adaptions, which are discussed afterwards, and then also carried over to $\mathrm{OND}_{\mathrm{tmh}}$.

Decomposition of $\mathrm{OND}_{\mathrm{tsh}}$. Consider the model $\mathrm{OND}_{\mathrm{tsh}}$ from page 61. The basic relation between link capacities and wavelength capacities on links is expressed by constraints

$$y_\ell = \sum_{\lambda\in\Lambda} y_\ell^\lambda \qquad \forall\, \ell \in L \tag{3.1}$$

which can be added to the model, introducing the variables y_ℓ. Using this relation directly allows to retransform constraints (2.10) to (2.5) and (2.11) to (2.1b). For each link $\ell \in L$, we build the sum over all wavelengths of constraints (2.9) and get

$$\sum_{\lambda\in\Lambda}\sum_{w\in\mathcal{W}}\sum_{f\in\mathcal{F}_w} k^{w\lambda} z_\ell^{fw} = \sum_{w\in\mathcal{W}}\sum_{f\in\mathcal{F}_w} k^w z_\ell^{fw} = \sum_{\lambda\in\Lambda} y_\ell^\lambda = y_\ell \qquad \forall\, \ell \in L,\ \lambda \in \Lambda$$

by Definition 2.5 and relations (3.1). These redundant constraints equal conditions (2.3a) which can therefore be added to the model, too. Hence, model $\mathrm{OND}_{\mathrm{tsh}}$ now includes the link configuration formulation (2.3).
Next, the sum over all wavelengths of constraints (2.13c) results in

$$\sum_{\lambda\in\Lambda}\sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ \ell\in L(p)}} a_{p\,\ell}^\lambda = \sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ \ell\in L(p)}}\sum_{\lambda\in\Lambda} a_{p\,\ell}^\lambda \le \sum_{\lambda\in\Lambda} y_\ell^\lambda \overset{(3.1)}{=} y_\ell \qquad \forall\, \ell \in L$$

where insertion of the appropriate equations (2.13b) yields

$$\sum_{\substack{p\in\mathcal{P}:\\ \ell\in L(p)}}\sum_{\substack{q\in\mathcal{Q}:\\ p\in\widetilde{\mathcal{P}}_{o_q d_q}}} f_p^q \le y_\ell \qquad \forall\, \ell \in L \tag{3.2}$$

as further redundant inequalities to be added. Together, constraints (2.13a), (3.2), and integrality conditions $f_p^q \in \mathbb{Z}_+$ from (2.13f) form the (opaque) path flow model (2.7) just using the single-hop path set $\widetilde{\mathcal{P}}$ instead of $\mathcal{P}$. Finally, all variables f_p^q in constraints (2.13b) are replaced by duplicates $\widetilde{f}_p^q$ which are identified by adding the equations

$$f_p^q = \widetilde{f}_p^q \qquad \forall\, q \in \mathcal{Q},\ p \in \widetilde{\mathcal{P}} \tag{3.3}$$

to the model, and for which separate integrality conditions $\widetilde{f}_p^q \in \mathbb{Z}_+$ are added to (2.13f).
Since only equivalence transformations have been applied so far, the resulting model is equivalent to $\mathrm{OND}_{\mathrm{tsh}}$. Now, if we relax the relations (3.1) and (3.3) as well as constraints (2.9), the model decomposes into two separate parts as follows:

$\mathbf{OND}_{\mathbf{tsh}}^{DR}$
containing
- objective (2.8),
- setup constraints (2.1),
- node configuration model (2.2),
- link configuration model (2.3),
- capacity linking constraints (2.5),
- routing model (2.7) with $\widetilde{\mathcal{P}}$ instead of $\mathcal{P}$.

$\mathbf{OND}_{\mathbf{tsh}}^{WA}$
containing
- objective $\min \sum_{n\in N} \sum_{c\in\mathcal{C}} C_n^c x_n^c$,
- wavelength converter model (2.12), and
- wavelength assignment model (2.13b)–(2.13f) with $\widetilde{f}_p^q$ replacing f_p^q.

The model $\mathrm{OND}_{\mathrm{tsh}}^{DR}$ represents the dimensioning and routing subproblem without distinction of wavelengths, whereas $\mathrm{OND}_{\mathrm{tsh}}^{WA}$ encodes the corresponding wavelength assignment subproblem including the placement of converters. The relaxed constraints specify some variable relations between the partial problems. Hence, any solution of $\mathrm{OND}_{\mathrm{tsh}}^{DR}$ turns variables y_ℓ^λ and $\widetilde{f}_p^q$ in $\mathrm{OND}_{\mathrm{tsh}}^{WA}$ into parameters defined by (2.9) and (3.3), respectively. As main foundation of the decomposition approach, the following theorem proves feasibility of $\mathrm{OND}_{\mathrm{tsh}}^{WA}$ for any such parameters obtained from a feasible solution of $\mathrm{OND}_{\mathrm{tsh}}^{DR}$, such that the combined solution is feasible for the original model $\mathrm{OND}_{\mathrm{tsh}}$.

Theorem 3.1 *For any feasible solution of OND_{tsh}^{DR} defining the values of y_ℓ^λ and $\widetilde{f}_p^q$ by equations* (2.9) *and* (3.3)*, there exists a feasible solution of OND_{tsh}^{WA}.*

Proof. Consider an arbitrary solution of $\mathrm{OND}_{\mathrm{tsh}}^{DR}$ with the implied values for all parameters y_ℓ^λ and $\widetilde{f}_p^q$.
Let $\ell \in L$ be an arbitrary, but fixed link. We define the sets of all (individual) selected routing paths traversing ℓ and all (individual) wavelengths available on ℓ by installed systems in the $\mathrm{OND}_{\mathrm{tsh}}^{DR}$ solution as

$$P_\ell := \bigcup_{\substack{p\in\widetilde{\mathcal{P}}:\\ \ell\in L(p)}} P_p \qquad \text{with} \qquad P_p := \left\{ p^{(i)} \;\middle|\; 1 \le i \le \sum_{\substack{q\in\mathcal{Q}:\\ p\in\widetilde{\mathcal{P}}_{o_q d_q}}} \widetilde{f}_p^q \right\} \quad \forall\, p \in \widetilde{\mathcal{P}}$$

and

$$\Lambda_\ell := \bigcup_{\lambda\in\Lambda} \Lambda_\ell^\lambda \qquad \text{with} \qquad \Lambda_\ell^\lambda := \left\{ \lambda^{(i)} \;\middle|\; 1 \le i \le y_\ell^\lambda \right\} \quad \forall\, \lambda \in \Lambda\,.$$

Due to mutual disjointness of the sets P_p and of the sets Λ_ℓ^λ,

$$|P_p| = \sum_{\substack{q\in\mathcal{Q}:\\ p\in\widetilde{\mathcal{P}}_{o_q d_q}}} \widetilde{f}_p^q \;,\; |P_\ell| = \sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ \ell\in L(p)}} \sum_{\substack{q\in\mathcal{Q}:\\ p\in\widetilde{\mathcal{P}}_{o_q d_q}}} \widetilde{f}_p^q \;,\; |\Lambda_\ell^\lambda| = y_\ell^\lambda \;,\; |\Lambda_\ell| = \sum_{\lambda\in\Lambda} y_\ell^\lambda = y_\ell$$

hold. From constraints (3.2), we know that $|P_\ell| \leq |\Lambda_\ell|$ and both sets are finite. Hence, there exist injective mappings $A : P_\ell \to \Lambda_\ell$.
Let A be such an arbitrarily selected mapping, and let $A^{-1}(X)$ refer to the set of all elements in P_ℓ that are mapped by A into the set X. Then we define the values of $a^\lambda_{p\,\ell}$ according to A by

$$a^\lambda_{p\,\ell} := |P_p \cap A^{-1}(\Lambda^\lambda_\ell)| \qquad \forall\, \lambda \in \Lambda,\ p \in \widetilde{\mathcal{P}}$$

as the number of times copies of path p are assigned to the wavelength λ. Since A is injective, we have $A^{-1}(\Lambda^\lambda_\ell) \cap A^{-1}(\Lambda^{\lambda'}_\ell) = \emptyset$ whenever $\lambda \neq \lambda'$. For each path $p \in \widetilde{\mathcal{P}}$, this implies

$$\begin{aligned}\sum_{\lambda\in\Lambda} a^\lambda_{p\,\ell} &= \sum_{\lambda\in\Lambda} |P_p \cap A^{-1}(\Lambda^\lambda_\ell)| = \left|P_p \cap \bigcup_{\lambda\in\Lambda} A^{-1}(\Lambda^\lambda_\ell)\right| \\ &= \left|P_p \cap A^{-1}\left(\bigcup_{\lambda\in\Lambda} \Lambda^\lambda_\ell\right)\right| = |P_p \cap P_\ell| = |P_p| = \sum_{\substack{q\in\mathcal{Q}:\\ p\in\widetilde{\mathcal{P}}_{o_q d_q}}} \widetilde{f}^q_p\end{aligned}$$

and thus verifies that all constraints (2.13b) for ℓ are fulfilled. Moreover, the disjointness of the sets P_p together with injectivity of A imply for each wavelength $\lambda \in \Lambda$

$$\begin{aligned}\sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ \ell\in L(p)}} a^\lambda_{p\,\ell} &= \sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ \ell\in L(p)}} |P_p \cap A^{-1}(\Lambda^\lambda_\ell)| = \left|\bigcup_{\substack{p\in\widetilde{\mathcal{P}}:\\ \ell\in L(p)}} P_p \cap A^{-1}(\Lambda^\lambda_\ell)\right| \\ &= |P_\ell \cap A^{-1}(\Lambda^\lambda_\ell)| = |A^{-1}(\Lambda^\lambda_\ell)| \leq |\Lambda^\lambda_\ell| = y^\lambda_\ell\end{aligned}$$

such that constraints (2.13c) are fulfilled for ℓ as well. As an arbitrary link ℓ was considered, the same conclusions hold for all links $\ell \in L$.
Regarding the converter model (2.12), remind that bounds (2.12c) and (2.12d) have been relaxed. So, a cheapest converter $c \in \mathcal{C}$ can be established arbitrarily often in each node $n \in N$, implying existence of feasible $\mathrm{OND}^{WA}_{\mathrm{tsh}}$ solutions with arbitrarily large values of $y^{\mathcal{C}}_n$. Hence, constraints (2.13d) and (2.13e) can be fulfilled trivially, e.g., by setting for each $p \in \widetilde{\mathcal{P}}$ with $p = (n_0, \ell_1, n_1, \ell_2, n_2, \ldots, \ell_h, n_h)$ the variables $a^\lambda_{p\,n} := a^\lambda_{p\,\ell}$ for all $\lambda \in \Lambda$ and all $1 \leq i \leq h-1$, as well as for each node $n \in N$ the variables $y^{\mathcal{C}}_n := \sum_{\substack{p\in\widetilde{\mathcal{P}}:\\ n\in N(p)}} \sum_{\substack{q\in\mathcal{Q}:\\ p\in\widetilde{\mathcal{P}}_{o_q d_q}}} \widetilde{f}^q_p$ and $z^c_n := y^{\mathcal{C}}_n$ together with $x^c_n := z^c_n - e^c_n$ for the selected converter device c. Altogether, a feasible solution of $\mathrm{OND}^{WA}_{\mathrm{tsh}}$ has been constructed, proving the claim. ■

As a consequence, Theorem 3.1 inspires to solve $\mathrm{OND}_{\mathrm{tsh}}$ by first solving $\mathrm{OND}^{DR}_{\mathrm{tsh}}$, fixing the values for all y^λ_ℓ and $\widetilde{f}^q_p$ parameters by (2.9) and (3.3), and then determining a corresponding solution of $\mathrm{OND}^{WA}_{\mathrm{tsh}}$ with respect to these parameters. Note that the parameter fixing is applied by some previously relaxed constraints which are therefore automatically fulfilled as well. The satisfaction of the remaining relaxed constraints (3.1) follow implicitly from constraints (2.3a), (2.9), and Definition 2.5 since the above applied opposite transformation, involving equations only, is reversible. Hence, each combination of a solution of $\mathrm{OND}^{DR}_{\mathrm{tsh}}$ and a corresponding

solution of $\mathrm{OND}^{WA}_{\mathrm{tsh}}$ provides in fact a feasible solution of the unrelaxed modified model equivalent to $\mathrm{OND}_{\mathrm{tsh}}$.
As a relaxation of $\mathrm{OND}_{\mathrm{tsh}}$, the submodel $\mathrm{OND}^{DR}_{\mathrm{tsh}}$ has a further useful property.

Proposition 3.2 *Any lower bound on the optimal value for OND^{DR}_{tsh} is also a lower bound on the optimal value for OND_{tsh}.*

Note that the objective of $\mathrm{OND}^{DR}_{\mathrm{tsh}}$ leaves only converter costs out of consideration, while all other cost drivers are involved. Thus good lower bounds for $\mathrm{OND}^{DR}_{\mathrm{tsh}}$ can be expected to provide also good lower bounds for $\mathrm{OND}_{\mathrm{tsh}}$. Clearly, lower bounds for the subsequently solved model $\mathrm{OND}^{WA}_{\mathrm{tsh}}$ do not contribute in this regard. Nevertheless, the objective of $\mathrm{OND}^{WA}_{\mathrm{tsh}}$ takes care that completing a solution for $\mathrm{OND}_{\mathrm{tsh}}$ occurs at minimum additional cost.
Furthermore, the dimensioning and routing subproblem $\mathrm{OND}^{DR}_{\mathrm{tsh}}$ is very similar to opaque network design as modeled in $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$ (see page 56). In fact, $\mathrm{OND}^{DR}_{\mathrm{tsh}}$ differs from $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$ only by the restricted routing path set $\widetilde{\mathcal{P}}$ instead of $\mathcal{P}$. This observation has a further consequence: incorporation of survivability works well with the decomposition. One easily verifies that the model adaptions described in Section 2.3 for inclusion of any DSP survivability scheme into $\mathrm{OND}_{\mathrm{tsh}}$ affect only model parts that are transformed into $\mathrm{OND}^{DR}_{\mathrm{tsh}}$, and that the decomposition transformations turn these adaptions to those proposed for $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$ when applied to $\mathrm{OND}^{DR}_{\mathrm{tsh}}$. Therefore, a slightly adapted decomposition, including statements corresponding to Theorem 3.1 and Proposition 3.2, applies also to survivable design of single-hop transparent optical networks using an arbitrary DSP survivability scheme.

Decomposition of $\mathrm{OND}_{\mathrm{tmh}}$. A very similar decomposition as for $\mathrm{OND}_{\mathrm{tsh}}$ can also be applied for $\mathrm{OND}_{\mathrm{tmh}}$, with few changes following the single-hop model extension for multi-hop optical connections discussed in Section 2.2. In fact, all above described model transformation steps preparing the decomposition carry over literally. In addition to this, the dimensioning and routing submodel $\mathrm{OND}^{DR}_{\mathrm{tmh}}$ extends by the regenerator model (2.15), and its objective by the regenerator costs $\sum_{n\in N}\sum_{r\in\mathcal{R}} C^r_n x^r_n$. All further changes concern only the routing model. The transition from $\mathrm{OND}_{\mathrm{tsh}}$ to $\mathrm{OND}_{\mathrm{tmh}}$ required to replace constraints (2.13a) and (2.13b) by (2.16a) and (2.16b) and to add constraints (2.16c), whereas conditions (2.13c)–(2.13f) have been kept. Consequently, the decomposed submodels reflect these adaptions. For the multi-hop dimensioning and routing submodel $\mathrm{OND}^{DR}_{\mathrm{tmh}}$, constraints (2.16b) together with (2.13c) transform in the same way to (2.7b) as well as conditions (2.13f) to (2.7c), while constraints (2.7a) are substituted by (2.16a) and constraints (2.16c) are inserted additionally. Finally, the wavelength assignment submodel $\mathrm{OND}^{WA}_{\mathrm{tmh}}$ is just marginally influenced by replacing constraints (2.13b) with (2.16b) where as only difference the flow variables are summed up over all commodities rather than just those for which the considered lightpath is an end-to-end connection.
Summarizing, the equivalent transformations turn model $\mathrm{OND}_{\mathrm{tmh}}$ analogously into a modified formulation which decomposes after relaxation of constraints (3.1), (3.3), and (2.9) as follows:

$\mathrm{OND}^{DR}_{\mathrm{tmh}}$

containing

- objective (2.8) + $\sum_{n\in N}\sum_{r\in\mathcal{R}} C^r_n x^r_n$,
- setup constraints (2.1),
- node configuration model (2.2),
- link configuration model (2.3),
- capacity linking constraints (2.5),
- regenerator model (2.15),
- routing model (2.16a) and (2.16c), together with (2.7b) and (2.7c) where $\widetilde{\mathcal{P}}$ replaces $\mathcal{P}$.

$\mathrm{OND}^{WA}_{\mathrm{tmh}}$

containing

- objective min $\sum_{n\in N}\sum_{c\in\mathcal{C}} C^c_n x^c_n$,
- wavelength converter model (2.12), and
- wavelength assignment model (2.16b) and (2.13c)–(2.13f) with $\widetilde{f}^q_p$ replacing f^q_p.

A comparison to the decomposition of $\mathrm{OND}_{\mathrm{tsh}}$ shows that wide parts of the submodels are equal. In particular, the proof of Theorem 3.1 holds also for the decomposition of $\mathrm{OND}_{\mathrm{tmh}}$ when just substituting the definition of multi-sets P_p by

$$P_p := \left\{ p^{(i)} \;\middle|\; 1 \le i \le \sum_{q\in\mathcal{Q}} \widetilde{f}^q_p \right\} \qquad \forall\, p \in \widetilde{\mathcal{P}}$$

and following the same outline to construct a feasible solution of $\mathrm{OND}^{WA}_{\mathrm{tmh}}$ with (2.16b) instead of (2.13b). So, the statement of Theorem 3.1 carries over to the decomposition of $\mathrm{OND}_{\mathrm{tmh}}$, too:

Theorem 3.3 *For any feasible solution of OND^{DR}_{tmh} defining the values of y^λ_ℓ and $\widetilde{f}^q_p$ by equations* (2.9) *and* (3.3)*, there exists a feasible solution of OND^{WA}_{tmh}.*

Any such pair of corresponding solutions for $\mathrm{OND}^{DR}_{\mathrm{tmh}}$ and $\mathrm{OND}^{WA}_{\mathrm{tmh}}$ again satisfies all relaxed constraints and thus combines to a feasible solution of $\mathrm{OND}_{\mathrm{tmh}}$.
Moreover, $\mathrm{OND}^{DR}_{\mathrm{tmh}}$ is a relaxation of $\mathrm{OND}_{\mathrm{tmh}}$, and so we can conclude analogously to Proposition 3.2:

Proposition 3.4 *Any lower bound on the optimal value for OND^{DR}_{tmh} is also a lower bound on the optimal value for OND_{tmh}.*

Finally, integrating survivability again does not impede the application of the decomposition approach. The necessary adaptions, as described in Section 2.3 depending on the selected DSP scheme, concern constraints (2.16a) and the insertion of further conditions which become part of $\mathrm{OND}^{DR}_{\mathrm{tmh}}$. The wavelength assignment submodel $\mathrm{OND}^{WA}_{\mathrm{tmh}}$ is not affected at all, and hence the decomposition with all properties carries over to survivable multi-hop transparent optical network design in an straightforward way.

Decomposition approach. Summarizing, Theorems 3.1 and 3.3 guarantee that feasible network designs can be constructed from separate solutions of the decomposed models, even when survivability is involved. Furthermore, the resulting dimensioning and routing models $\mathrm{OND}^{DR}_{\mathrm{tsh}}$ and $\mathrm{OND}^{DR}_{\mathrm{tmh}}$ are quite similar to $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$ and, by Propositions 3.2 and 3.4, provide for the original problems a lower bound which involves already all costs except those for conversion. So, to relieve the com-

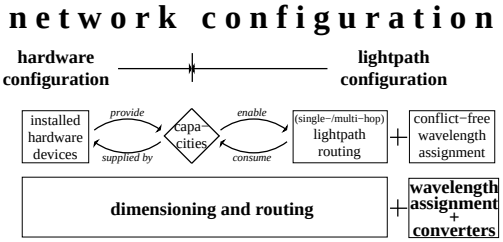


Figure 3.2: *Decomposition approach for transparent optical network design.*

putational effort without losing too much solution quality, we apply the following *decomposition approach*, illustrated in Figure 3.2, and solve the entire optical network design task in two subsequent steps:

- **dimensioning and (survivable) routing**
  for which we neglect the distinction of wavelengths and use (sub-) models $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$, $\mathrm{OND}^{DR}_{\mathrm{tsh}}$, or $\mathrm{OND}^{DR}_{\mathrm{tmh}}$, respectively, followed by
- **wavelength assignment and converter placement**
  for the established lightpaths in case of transparent optical networks as formulated by submodels $\mathrm{OND}^{WA}_{\mathrm{tsh}}$ or $\mathrm{OND}^{WA}_{\mathrm{tmh}}$.

Note that compared to the previous classification, wavelength converters are now excluded from the hardware dimensioning since their functionality is directly connected to the assignment of wavelengths.

Discussion. Solving a problem by decomposition into subproblems has some pros and contras. For the proposed approach, a major concern is the loss of the optimality guarantee. Even optimal solutions of both submodels need not combine to an optimal solution of the original problem as long as converters are required. The dimensioning and routing subproblem has no control on the number of conversions finally needed, and so converters might incur a substantial portion of the design costs, e.g., when these devices have high prices or upgrade planning already offers a lot of infrastructure for free. As a further consequence, the approach cannot guarantee to meet (non-trivial) bounds on the converter numbers.
On the other hand, application of the proposed decomposition offers some major advantages:

> **Decoupling hard subproblems.** The decomposition decouples (at least) the two identified $\mathcal{NP}$-hard subproblems, the integer multicommodity flow problem (with capacity dimensioning) and the coloring-like wavelength assignment problem. Their isolated investigation gives rise to expect better solvability properties than with a direct approach to the comprehensive models derived in Section 2.2.
>
> **Common methodology.** As a further feature, the approach allows to tackle all technology scenarios presented in the previous chapter with the same methodology. By excluding the wavelength assignment, the remaining dimensioning and routing tasks are very similar. In fact, $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$, $\mathrm{OND}^{DR}_{\mathrm{tsh}}$, and $\mathrm{OND}^{DR}_{\mathrm{tmh}}$ have only few differences and can be easily unified (see Section 3.3). An even stronger similarity holds for the $\mathrm{OND}^{WA}_{\mathrm{tsh}}$ and $\mathrm{OND}^{WA}_{\mathrm{tmh}}$ models.

Applicable know-how. A special advantage is that the wavelength-free dimensioning and routing (sub-) problems have fundamental similarities to non-optical network design which has been widely studied. Hence, we can profit from the existing results, in particular sophisticated optimization methods that are already available.

Good solutions. Our computational experiments have shown that typically few converters are required to accommodate a conflict-free wavelength assignment for a lightpath routing. Often, even converter-free solutions could be found (see Chapters 4 and 5). In such a case, an optimal dimensioning and routing subproblem solution is in fact completed to a provably optimal solution of the overall problem.

Solution method overview. Prior to the detailed presentation and discussion of the applied algorithm(s), we give a basic overview on the major steps of our solution method. Figure 3.3 illustrates the procedure schematically. The input comprises the physical and supply networks, the hardware, and the demands as introduced in Section 2.2.1. In the first phase, the dimensioning and routing subproblem is solved, with pre- and postprocessing procedures around a sophisticated optimization method. As result, we get a lower bound on the total cost, a network with properly dimensioned capacities (except for conversion) and a (survivable) optical connection routing. For transparent networks, the second phase then accomplishes the lightpath configuration by a conflict-free wavelength assignment and the hardware configuration by placement of wavelength converters. After all, the output comprises the solution as complete network configuration and a quality guarantee by the computed lower bound of the first phase.

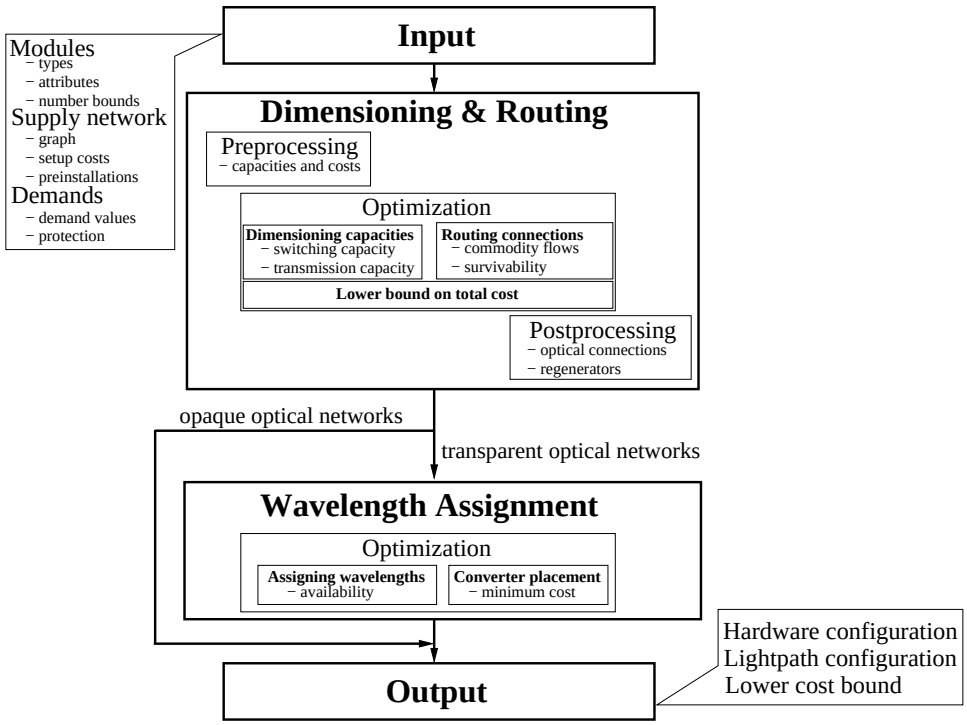


Figure 3.3: *Solution method scheme.*

In the following, we describe this comprehensive method step-wise in detail. According to the natural order, we begin with the dimensioning and routing procedure, while wavelength assignment with converters is postponed to Chapter 4.

3.3 Dimensioning and routing

As first phase of our method, we solve the dimensioning and routing subproblem with the goal to minimize the network cost, except for converters. Despite excluding wavelength assignment, this subproblem remains a rather difficult remit, representing an integer multicommodity flow problem with capacity constraints and additional restrictions according to the particular scenario. Before going into detail on the solution process, we unify the models for all scenarios.

Model unification. As result of the model decompositions in the previous section, we consider the models $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$, $\mathrm{OND}^{DR}_{\mathrm{tsh}}$, and $\mathrm{OND}^{DR}_{\mathrm{tmh}}$. Since wavelengths are not distinguished anymore in the transparent scenarios, we only have to deal with channel capacities for transmission and switching as well as a routing of optical connections in terms of entire (uncolored) channels. The connection types indeed differ. When adapting the path set $\mathcal{P}$ to $\widetilde{\mathcal{P}}$ throughout (2.7), $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$ and $\mathrm{OND}^{DR}_{\mathrm{tsh}}$ already become identical. Transparent multi-hop networks may additionally employ regenerators, concatenating lightpaths to the final connections as expressed in (2.16a). By neglecting regenerations, however, these connections can be handled exactly the same way as those in opaque networks. For this, we exclude the placement of regenerators in transparent multi-hop networks from the main optimization phase and accomplish this issue in a postprocessing step. So, the regenerator model (2.15), the corresponding objective function term, and constraints (2.16c) are removed from $\mathrm{OND}^{DR}_{\mathrm{tmh}}$, and constraints (2.16a) are replaced by (2.7a). These modifications turn the model into $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$ in the end. In view of the lower bound, this additional reduction has the same consequences as the exclusion of wavelength converters. Regenerator costs are not anymore involved in the optimization step, but any lower bound on the cost obtained for the reduced problem remains valid for the original problem.
As a result, the dimensioning and routing subproblem becomes the same for all three scenarios and is represented by the opaque network design model $\mathrm{OND}^{\mathrm{p}}_{\mathrm{opq}}$, optionally with the corresponding adaptions to integrate survivability as described in Section 2.3. The sole difference concerns the transparent single-hop case where the selection of routing paths is restricted to those of limited length, as modeled in (2.13), and needs just $\mathcal{P}$ to be replaced by $\widetilde{\mathcal{P}}$ in the (survivable) routing model (2.7).

Method selection. As already indicated, opaque network design is also very similar to the design of non-optical networks, which has been intensively studied since long before optics entered the networks. Hence, it is reasonable to avail the existing knowledge rather than to reinvent the wheel. There are several sophisticated optimization methods and solution tools which can be used to handle the (reduced) dimensioning and routing subproblem.
For our model, one particularly suitable tool is DISCNET introduced by Wessäly [166]. DISCNET exploits integer linear programming techniques to achieve two goals: to find

good feasible solutions and to derive a lower bound on the total cost of any network design that satisfies the stated requirements. Thereby, the underlying model and the solution algorithm are flexible enough to incorporate many specific aspects of our setting, such as the survivability concept. Moreover, optional parameter settings allow to balance computational effort and solution accuracy.

To take advantage of these features for our purpose, we address the necessary integration that has to be carried out to use Discnet. At first, we discuss possible transformations of our problem description into a suitable input for Discnet in a preprocessing phase. Next, we present in more detail the methodology of Discnet and its application as a solver. The intermediate result obtained from Discnet is finally handed to a postprocessing step to accomplish the solution of our dimensioning and routing subproblem.

3.3.1 Transforming the problem

As introduced by Wessäly [166], Discnet is based on a simplified representation of available hardware configurations by means of a discrete set of the corresponding capacities, each with an associated cost value. Initially restricted to links, the model has been later carried over to node capacities, too. Main advantage of this approach is that dimensioning reduces to the selection of a single configuration out of the given set for each link and node, but presumes to have such capacity-cost sets at hand.

Components and resources. Meanwhile, Kröller [99] developed and implemented a more sophisticated network framework which extends the integer linear programming formulation of Wessäly. Instead of sets with alternative capacities (and costs), the hardware composition at each link and node is taken into account in more detail. The framework uses the abstract concept of *components* and *resources* to represent real or fictive objects and their provision or consumption of definable resources. This abstraction adds high flexibility to the tool. As demonstrated in Bley et al. [24], the extension offers a direct way to model (and solve) design problems for various technologies, such as IP, SDH, ATM, or—of our particular interest—opaque WDM networks. The transfer of our modular network description into this framework is quite simple, using (wavelength) capacities and installation space as resources, whereas modules and lightpaths form the components. We omit further details, which are straightforward to carry out.

Discnet **interfaces.** As a result, both input interfaces are available for Discnet today. Clearly, the component/resource concept is more comfortable to use, but brings also further complexity into the task handed over to the optimization algorithm. In fact, whereas a simple decision of which capacities to install in the network had to be taken before, the full set of different configurations has then to be encoded in the integer linear program. Hence, the 'easy' component/resource based interface suits well as long as the multitude of alternative configurations is reasonably small, in particular without providing too many different configurations for any installable capacity. Otherwise, the complexity reduction by exclusion of such configuration details and a compressed representation of the selectable capacities is advisable for

performance improvement of the solving procedure. In view of our module variety, simplifications are often preferable, and thus we describe how to convey our module-based configuration formulations (2.2) for nodes and (2.3) for links to the alternative model on discrete sets for capacities and costs in a preprocessing.

Model for discrete sets. At first, we restate the model for discrete sets from Wessäly [166] in our notation, as introduced in Chapter 2. For each link or node (subsumed as *place*) $\pi \in N \cup L$, assume the set $K_\pi = \{(k_1^\pi, C_1^\pi), \ldots, (k_{m_\pi}^\pi, C_{m_\pi}^\pi)\}$ comprises any installable capacity $k_i^\pi \in \mathbb{Z}_+$ available at cost $C_i^\pi \in \mathbb{R}_+$, $i = 1, \ldots, m_\pi$. For the dimensioning, binary variables $x_i^\pi \in \{0, 1\}$ to express whether capacity k_i^π is selected at place $\pi \in N \cup L$ or not. The original model for dimensioning with discrete capacities then reads:

$$\min \sum_{\pi \in L \cup N} \sum_{i=1}^{m_\pi} C_i^\pi x_i^\pi \tag{3.4}$$

$$\text{s.t.} \quad \sum_{i=1}^{m_\pi} k_i^\pi x_i^\pi = y_\pi \qquad \forall\, \pi \in N \cup L \tag{3.5a}$$

$$\sum_{i=1}^{m_\pi} x_i^\pi = 1 \qquad \forall\, \pi \in N \cup L \tag{3.5b}$$

$$x_i^\pi \in \{0, 1\} \qquad \forall\, \pi \in N \cup L,\ i = 1, \ldots, m_\pi \tag{3.5c}$$

Together with the integrality conditions (3.5c), constraints (3.5b) ensure that a single option is selected at each place $\pi \in N \cup L$. The corresponding capacity is determined in constraints (3.5a), and the objective (3.4) summarizes the costs for all selections. We remark that the original model in Wessäly [166] is formulated with incremental capacities and costs, motivated by a preferable computational behavior. However, it is easy to verify that the original model is equivalent to (3.4) and (3.5).

Configurations and capacities. In order to substitute the configuration formulations by the simpler model (3.5), we have to determine the discrete sets K_π for all places $\pi \in N \cup L$. Revisiting our formulations, one easily observes that the hardware configuration of each individual link or node is independent of all others, except for parallel supply links associated to the same physical link where the installation of WDM systems and the total number of fibers can be restricted in common. In such a case, it is not possible to use sets of discrete capacities and costs for individual supply links which can be selected independently of each other, and an appropriate component and resource model with auxiliary entities representing resources consumed concertedly by parallel supply links must be applied. So, we continue for the case that no such (non-trivial) common restrictions are given. If preinstalled WDM systems are present, we fix their utilization to the currently carrying fiber (and thus supply link). Then the configuration model for parallel supply links can be decoupled, and the discrete capacity and cost set calculation can be independently carried out place-wise as next considered individually for an arbitrary supply link or node $\pi \in N \cup L$. In what follows, major notation and formulation changes can be avoided by simply refining the set of physical links such that each physical link corresponds to a single supply link, i.e., we assume mutually disjoint $L_t = \{\ell\}$ for all $t \in T$ and

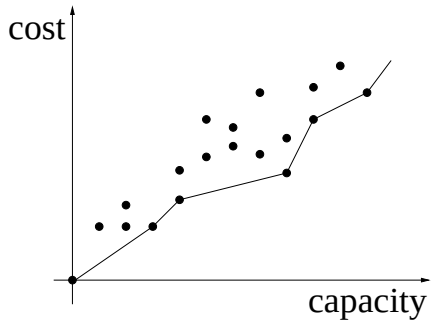


Figure 3.4: *Example of installable capacities and their costs for all potential hardware configurations at a link or a node.*

use associated physical and supply links ℓ and t synonymously.
Each feasible hardware configuration at π according to (2.2) respectively (2.3) yields an installable capacity with precomputable cost. Taking any possible configuration into account might yield a result as plotted in Figure 3.4 by a dot for each set element. In general, some configurations may be dominated by others which provide the same or higher capacities at equal or even lower cost by taking advantage of economies of scale. Such dominated configurations need not be considered, and it suffices to take only undominated configurations into account, whose corresponding dots constitute the lower envelope in Figure 3.4.
Moreover, capacities beyond a maximally required amount need also not be evaluated. Such a maximum value $\hat{k}_\pi$ can be determined as follows. For each commodity, the maximum throughput at π is either the demand value or even stricter limited by survivability restrictions. All such limits can be predetermined except for the most general DSP scheme where the total number of connections to establish for a commodity is not specified in advance. In this case, we instead use the upper bound on the total connection number provided by Proposition 2.21 on page 84 and so can derive a maximum throughput limit for each commodity at each π. Summing up these limits over all commodities, we find $\widetilde{k}_\pi$ as maximum number of connections that can traverse π in any design. Since a capacity of exactly $\widetilde{k}_\pi$ need not be installable at π, the value $\hat{k}_\pi$ is finally obtained by adding the maximum capacity of an installable switch if $\pi \in N$, or of an installable WDM system otherwise, i.e., we set $\hat{k}_n = \widetilde{k}_n + \max_{s\in\mathcal{S}} k^s$ for nodes $n \in N$ and $\hat{k}_\ell = \widetilde{k}_\ell + \max_{w\in\mathcal{W}} k^w$ for links $\ell \in L$. Note that in case no bounds on the number of installable devices are given, determination of $\hat{k}_\pi$ is in fact required to terminate the generation of capacities.

Set generation by ILPs. Given a required capacity $k \in \mathbb{Z}_+$ for a node or link, the minimum cost configuration providing at least this capacity can be determined by models (2.2) and (2.3), respectively. So, a self-evident idea to generate the discrete capacity sets is by repeatedly solving these configuration formulations individually for each supply link and node. For this, we define the following functions:

$$z_n(k) := \left\{ \min \sum_{s\in\mathcal{S}} C_n^s x_n^s \;\middle|\; (2.2),\ y_n \geq k \right\}$$

$$z_\ell(k) := \left\{ \min \sum_{f\in\mathcal{F}} C_\ell^f x_\ell^f + \sum_{w\in\mathcal{W}} C_t^w x_t^w \;\middle|\; (2.3),\ y_\ell \geq k \right\} \quad (\text{where } L_t = \{\ell\})$$

For each place $\pi \in L \cup N$, the function $z_\pi(k)$ returns the minimum cost of all configurations providing at least capacity k, and each optimal model solution specifies such a corresponding configuration and its real capacity y^*_π ($\geq k$). Starting with $k = 0$, we iteratively compute $z_n(k)$ and set $k := y^*_\pi + 1$ for the next iteration until either $z_n(k) = \infty$ or $k > \hat{k}_\pi$. After that, we discard dominated capacities, if found some, and obtain the desired minimum discrete sets. However, such a method needs to solve many integer linear programs and might become a time-consuming process. A typically more effective way is to use dynamic programming.

Set generation by dynamic programming. Any hardware configuration of a node can be seen as a collection of individual switches, each contributing a certain amount to the total capacity. Vice versa, we can 'build' any feasible node configuration by deciding for each installable switch device individually whether to participate in the configuration or not. Moreover, given an optimal configuration for capacity k containing a switch device u of type s, the configuration without that switch must be optimal for capacity $k - k^s$ among all configurations not using u. Otherwise, there would obviously exist a cheaper configuration for k, contradicting its assumed optimality. On links, the same holds for pairs of fibers and WDM systems. This idea gives the key for a (configuration) generation method according to the principle of dynamic programming (see, e.g., Nemhauser and Wolsey [127], Section II.5.5, for an introduction).
At first, we build the set of all potentially employable *units* (i.e., all switches for nodes, all pairs of fibers and WDM systems for links). Thereby, we distinguish preinstalled units available for free and new units to be installed against the corresponding cost. Note that we have also to decide on the use of preinstalled units, since it might be necessary to replace those units by (new) larger ones due to limited space. Then, we resolve the generation of all configurations in a sequence of decisions on the participation of each of these units, thereby keeping track of a cheapest configuration for all capacities upon the units involved so far. As principle of optimality, any part of a cheapest configuration forms a cheapest configuration itself, if restricting the involved units to those considered upon completion of the part. Having processed all units in the end, all feasible configurations have been evaluated implicitly, and we obtain a cheapest configuration for each installable capacity.
In the following, we describe the general methods separately for nodes and links due to the different hardware composition. We begin with the hardware configuration at nodes.

Dynamic programming method for node capacities. The switching capacity at a node $n \in N$ is provided by the installed switches whose attributes are defined in Definition 2.6 on page 49.

Generation of the discrete capacity set at a node

Input. A node $n \in N$ and its configuration model (2.2) with all associated parameters.

Method. For each switch type $s \in \mathcal{S}$, we define the set $U^s_n := \{u^s_1, \ldots, u^s_{b^s_n}\}$ of individual switch units that can be used at node n. If b^s_n is unlimited, we set $b^s_n := \lceil \hat{k}_n / k^s \rceil$. Each such unit contributes a capacity of k^s and has a cost of C^s_n except for the first e^s_n units which are preinstalled and

thus for free. For notational convenience, we introduce a capacity function $k(n,u)$ and a cost function $C(n,u)$ for these values defined by

$$k(n,u_i^s) := k^s \quad \text{and} \quad C(n,u_i^s) := \begin{cases} 0 & , \ i \le e_n^s \\ C_n^s & , \ i > e_n^s \end{cases} \ \forall\, i = 1,\dots,b_n^s \ .$$

So, $k(n,u_i^s)$ is the capacity of the i-th unit and $C(n,u_i^s)$ the associated cost. All individual units of any type are subsumed in the set

$$U_n := \bigcup_{s\in\mathcal{S}} U_n^s =: \{\ u_1,\dots,u_{B_n}\ \} \qquad \text{with} \qquad B_n := \sum_{s\in\mathcal{S}} b_n^s$$

where the unit indices constitute an arbitrary, but fixed ordering. Note that any combination of units U_n implicitly satisfies the bounds b_n^s for the number of each type, but not necessarily the bound $b_n^{\mathcal{S}}$ on the total number of switches at node n. For unlimited $b_n^{\mathcal{S}}$, we can simply set $b_n^{\mathcal{S}} := B_n$ (see below for an alternative and more efficient way to treat this special case).
Now, we define the functions $r_i(k,b)$ to hold the minimum cost for providing exactly a switching capacity of k by any configuration composed of exactly b units selected among the first i units $\{u_1,\dots,u_i\} \subset U_n$. These functions are evaluated sequentially for $i = 1,\dots,B_n$. For initialization, we set $r_0(0,0) = 0$ and $r_0(k,b) = \infty$ for all $k \neq 0$, $b \neq 0$, including negative values for k and b for notational convenience. Then we work along the unit sequence. In each step i, we consider whether additional inclusion of unit u_i yields cheaper configurations or not, which is expressed by the relation

$$r_i(k,b) = \min\{r_{i-1}(k,b), r_{i-1}(k-k(n,u_i), b-1) + C(n,u_i)\} \quad (3.6)$$

for each capacity $0 \le k \le \hat{k}_n$ and unit number $1 \le b \le b_n^{\mathcal{S}}$. The last function $r_{B_n}(k,b)$ then holds for each capacity k the minimum cost over all feasible configurations with b units. So, we finally set

$$r(k) := \min_{b=0,\dots,b_n^{\mathcal{S}}} r_{B_n}(k,b) \qquad \forall\, 0 \le k \le \hat{k}_n$$

to get the minimum cost of any configuration providing capacity k. Referring to Figure 3.4, we obtain this way the lowest dot for each capacity. The method is completed by discarding all dominated capacities and those with infinite cost. An efficient way for this is to start with the highest capacity and to proceed the entries in order of non-increasing capacities, discarding any entry with cost higher than or equal to the last non-removed capacity-cost pair.

Output. After terminating, the method returns the desired minimal discrete set $K_n = \{(k_1^n, r(k_1^n)),\dots,(k_{m_n}^n, r(k_{m_n}^n)\}$ of installable capacities and their costs at node n.

In this general form, the method has a running time of order $\mathcal{O}(B_n \cdot \hat{k}_n \cdot b_n^{\mathcal{S}})$ and thus provides an algorithm that is pseudo-polynomial in time and space.

Algorithm 3.1 Minimum discrete set of capacities and costs for node $n \in N$

Require: The set $U_n = \{u_1, \ldots, u_{B_n}\}$ (in arbitrary but fixed order) of installable units with associated capacities $k(n,\cdot) : U_n \to \mathbb{Z}_+$ and costs $C(n,\cdot) : U_n \to \mathbb{R}_+$.
Ensure: The (minimal) set K_n.

```
set K_n = {(0,0)}                                              {initialization}
for i = 1 to B_n do
    set K'_n = K_n          {configurations from previous step, without unit u_i}
    for all (k,C) ∈ K_n do
        if k + k(n,u_i) ≤ k̂_n then
            if ∃ (k̄,C̄) ∈ K'_n : k̄ = k + k(n,u_i) then
                if C̄ > C + C(n,u_i) then
                    K'_n ← (K'_n \ {(k̄,C̄)}) ∪ {(k̄, C + C(n,u_i))}   {found cheaper config.}
                end if
            else
                K'_n ← K'_n ∪ {(k̄, C + C(n,u_i))}                    {found new capacity}
            end if
        end if
    end for
    K_n ← K'_n
end for
for all (k,C) ∈ K_n do
    if ∃ (k̄,C̄) ∈ K_n \ {(k,C)} : (C̄ ≥ C) ∧ (k̄ ≤ k) then
        K_n ← K_n \ {(k̄,C̄)}                                        {dominated element}
    end if
end for
return K_n
```

Special case. As described above, the dynamic programming method applies for any set of configuration parameters at a node, whereas simplifications are possible for particular settings. For demonstration, we just pick the special case of an unlimited bound $b_n^{\mathcal{S}}$. Instead of setting it to B_n, we can exploit this property to reduce the computational effort substantially. In fact, we need not account for the unit number in any configuration throughout the computation. This allows to remove the second parameter in all functions $r_i(k,b)$, and $r(k)$ coincides with $r_{B_n}(k)$ in the end. Moreover, the running time reduces to order $\mathcal{O}(B_n \cdot \hat{k}_n)$.
For this special case, Algorithm 3.1 provides an exemplary pseudo code of the method, with an efficient way to store and update the functions $r_i(.)$ by means of a repeatedly extended set, omitting all entries with infinite cost. At the end, the lower envelope capacities can be efficiently traced as already described, if storing K_n as sorted set according to non-increasing capacities. However, we skipped these details in Algorithm 3.1 to reduce notational effort in the pseudo-code. The same holds for storage of the corresponding hardware configurations whenever a better cost value has been found.

Dynamic programming method for links. The transmission capacity on links is provided by the installed combinations of fibers carrying WDM systems according to Definitions 2.4 and 2.5 on page 48. We remind that a refined physical topology is used such that physical and supply links coincide, i.e., ℓ and t are used

synonymously according to the relations $L_t = \{\ell\}$.

Generation of the discrete capacity set at a link

Input. A link $\ell \in L$ and its configuration model (2.3) with all associated parameters.

Method. An accurate cost calculation for new and preinstalled modules with all possible recombinations requires to refine the type sets for both module sorts, representing free and new devices separately. We replace each fiber type $f \in \mathcal{F}$ with $b_\ell^f > e_\ell^f > 0$ by two auxiliary types f_{free} with zero cost and $b_\ell^{f_{free}} := e_\ell^f$, and f_{new} with cost C_ℓ^f and $b_\ell^{f_{new}} := b_\ell^f - e_\ell^f$. All other attributes carry over from the original to both substituting types. WDM system types are processed analogously. So, we obtain new type sets $\mathcal{F}'$ and $\mathcal{W}'$ where each module type has an individual cost value.
Any feasible link configuration consists of the installed fibers and WDM systems as well as their assignment into compatible pairs. Hence, we consider any such pair as an installation unit. For arbitrary $w \in \mathcal{W}'$ and $f \in \mathcal{F}'_w$, the maximum number of installable pairs of these types is given by $b_\ell^{fw} := \min\{b_\ell^f, b_t^w\}$, and we introduce the set

$$U_\ell^{fw} := \{u_1^{fw}, \ldots, u_{b_\ell^{fw}}^{fw}\}$$

containing each of these pairs as an individual unit. Each unit $u_i^{fw} \in U_\ell^{fw}$ provides capacity $k_\ell(u_i^{fw}) = k^w$ and has cost $C_\ell(u_i^{fw}) = C_\ell^f + C_t^w$. We subsume all of these installation units in the set

$$U_\ell := \bigcup_{f\in\mathcal{F}, w\in\mathcal{W}} U_\ell^{fw} = \{\ u_1, \ldots, u_{B_\ell}\ \} \text{ with } B_\ell := \sum_{w\in\mathcal{W}'} \sum_{f\in\mathcal{F}'_w} b_\ell^{fw}$$

and reset the parameter $b_\ell^{\mathcal{F}} := \min\{b_\ell^{\mathcal{F}}, B_\ell\}$.
Unfortunately, the bounds on the number of modules for an individual type are not anymore implicitly satisfied for any subset of U_ℓ, whenever a fiber type is compatible with multiple WDM system types or vice versa. Hence, we have in general to keep track of the module type utilization explicitly, too. For this, we use the counting vectors $J_{\mathcal{F}'} = (j_f)_{f\in\mathcal{F}'} \in \mathbb{Z}_+^{|\mathcal{F}'|}$ for fibers and $J_{\mathcal{W}'} = (j_w)_{w\in\mathcal{W}'} \in \mathbb{Z}_+^{|\mathcal{W}'|}$ for WDM systems. The sets of feasible counting vectors, satisfying all number bounds, are defined by $\mathcal{J}_{\mathcal{W}'} := \{\ J_{\mathcal{W}'} \mid j_w \le b_t^w \ \ \forall\ w \in \mathcal{W}'\ \}$ and

$$\mathcal{J}_{\mathcal{F}'} := \left\{ J_{\mathcal{F}'} \;\middle|\; j_f \le b_\ell^f \ \ \forall\ f \in \mathcal{F}',\ \sum_{f\in\mathcal{F}'} j_f \le b_\ell^{\mathcal{F}} \right\},$$

and we denote by e_f and e_w the corresponding unit vectors, with all entries being 0 except for a 1 at position f and w, respectively.
Now, the dynamic programming approach has to evaluate the functions $r_i(k, J_{\mathcal{F}'}, J_{\mathcal{W}'})$ which denote the minimum cost for any configuration with capacity k composed of a subset of the installation units $\{\ u_1, \ldots, u_i\ \} \subset U_\ell$ which concertedly use the numbers of fibers and WDM systems of any type as expressed by the corresponding counting vectors $J_{\mathcal{F}'}$ and $J_{\mathcal{W}'}$. Due to

this definition, only some pairs of vectors from $\mathcal{J}_{\mathcal{F}'}$ and $\mathcal{J}_{\mathcal{W}'}$ form feasible arguments for these functions. For notational convenience, however, we do not go into further detail and allow simply for any argument combination, implicitly assigning infinite cost as function value for all impossible ones. So, we start with initializing all function values with infinity except for $r_0(0,0,0) := 0$, and then consider the units $u_i \in U_\ell$ consecutively for $i = 1, \ldots, B_\ell$. Whether or not using unit u_i with $u_i = u_{i'}^{fw}$ is favorable in the composed configurations is expressed by

$$r_i(k, J_{\mathcal{F}'}, J_{\mathcal{W}'}) = r_{i-1}(k, J_{\mathcal{F}'}, J_{\mathcal{W}'})$$

whenever $(j_f = 0) \vee (j_w = 0) \vee (k < k_\ell(u_i))$, and otherwise by

$$\begin{aligned} r_i(k, J_{\mathcal{F}'}, J_{\mathcal{W}'}) = \min\{\, & r_{i-1}(k, J_{\mathcal{F}'}, J_{\mathcal{W}'}), \\ & r_{i-1}(k - k_\ell(u_i), J_{\mathcal{F}'} - e_f, J_{\mathcal{W}'} - e_w) + C_\ell(u_i)\} \end{aligned}$$

for all capacities $0 \le k \le \hat{k}_\ell$ and all (associated pairs of feasible) counting vectors $J_{\mathcal{F}'} \in \mathcal{J}_{\mathcal{F}'}$ and $J_{\mathcal{W}'} \in \mathcal{J}_{\mathcal{W}'}$. At the end, function $r_{B_\ell}(k, ., .)$ is reduced to the desired capacity-cost correspondence by

$$r(k) = \min_{J_{\mathcal{F}'} \in \mathcal{J}_{\mathcal{F}'}, J_{\mathcal{W}'} \in \mathcal{J}_{\mathcal{W}'}} r_{B_\ell}(k, J_{\mathcal{F}'}, J_{\mathcal{W}'}) \ .$$

A final list cleanup in the same way as for nodes discards all dominated entries and those with infinite cost.

Output. In the end, the method returns the desired minimal discrete set $K_\ell = \{(k_1^\ell, r(k_1^\ell)), \ldots, (k_{m_\ell}^\ell, r(k_{m_\ell}^\ell)\}$ of installable capacities and their costs at link ℓ.

The evaluation of the functions $r_i(\cdot, \cdot, \cdot)$ looks exhaustive at first sight, but again this method reflects the most general case, and as indicated above, only few pairs of counting vectors form argument combinations for a finite function cost value. Therefore, propagating only these values step-wise as exemplified in Algorithm 3.1 reduces the computational effort significantly. Though the dynamic programming algorithm can become of exponential order for particular parameter settings, there are various special cases with pseudo-polynomial running time, e.g., with a fixed bounded number of fiber types, or WDM capacities which are mutually multiples of each other, like systems with 16, 32, and 64 wavelengths. In practice, the type sets are often quite small, also due to the supply network construction, and have typically such divisibility properties, such that the dynamic programming method remains an efficient algorithm to determine the discrete capacity sets for links, too.

Setup costs. After having determined the discrete sets K_π of installable capacities and their costs for all nodes and links $\pi \in N \cup L$, an additional integration of setup costs is easy to achieve. To this end, we simply add the setup cost value C_π to the cost of each non-zero capacity in the set K_π. If not contained (due to existence of a non-zero free capacity), we also add the entry $(0, 0)$ to K_π representing not to use this link or node, i.e., providing zero capacity at zero cost without incurring the setup cost. As a consequence, the reformulation (3.5) absorbs also the constraints (2.1) from the opaque network design model.

Capacity subsets. From a computational point of view, the alternative dimensioning model (3.5) is most often favorable since the solution variety reduces considerably this way. Moreover, algorithmic performance is typically correlated with the number of possible selections. The smaller the discrete capacity sets become, the faster we can expect to find good solutions for the problem. The cardinality of any K_π indeed depends on the particular setting and can become quite large. In such a case, it is possible to restrict the dimensioning options to a subset of the complete K_π for each link and node, which shrinks the solution space further. Nevertheless, any solution for such a reduced problem is feasible for the original problem as well. As drawbacks of the manipulation, a lower bound on the network cost for the reduced problem need not anymore be globally valid, thus we loose the quality guarantee, and optimal solution of the reduced problem need not anymore be a global optimum. However, the computation acceleration by reducing the search space can be used for quickly computing a couple of comprehensive network designs in a heuristic manner. In particular for practice, such a heuristic is often preferable for fast generation of designs at first instance, and thus this option has been integrated into our software. We provide several heuristic levels to rule the extent of the set reduction. Having found suitable designs this way, a solution quality can finally be regained by running the exact method until a (satisfying) lower bound is achieved.

Summarizing, two basic approaches are applicable to transform the unified dimensioning and routing subproblem into a suitable model for Discnet, either by direct use of the component-resource concept or with a precalculation of discrete capacity sets. As a result, we gain access to the optimization engine of Discnet.

3.3.2 Applying Discnet

In principle, Discnet can be applied comfortably as a black-box solver for the transformed dimensioning and routing problem. Nevertheless, it is worth to take a closer look inside and to see how the sophisticated mathematical optimization algorithm works. Discnet integrates a combination of decomposition techniques, cutting plane techniques, and linear programming based heuristic algorithms. Together, this constitutes a flexible method with several options to balance effort against solution accuracy and further parameters to guide the solution process. Thus, we briefly describe the algorithmic outline and explain different application possibilities for our purpose.

Model specification. Although we consider the dimensioning and routing problem in an unified form, optional model formulations have been proposed for hardware representation and survivability integration. The algorithmic outline of Discnet, however, is the same in any case. Therefore, we pick a specific setting in order to ease the explanation.
We use the model on the discrete capacity sets for hardware dimensioning and the basic DSP variant for survivability, with the precomputed DSP demand values $\widetilde{v}_q$ (cf. Section 2.3). Moreover, Discnet is based on a routing model with path variables. So, a complete formulation of the problem is given as:

DISCNET **integer linear program**

minimizing (3.4) subject to

- the hardware dimensioning model (3.5),
- the capacity linking constraints (2.5),
- the routing model (2.7), and
- the diversification constraints (2.24).

Even for moderate problem sizes, this integer linear program is typically too large for standard methods. As default, DISCNET solves a relaxation of the program by neglecting integrality of the path variables, i.e., with fractional routings. Note that a lower bound for this relaxation remains valid for the integer linear program. We first describe the solution method for the relaxed problem, before we stress possibilities to generate integer routings.

Algorithmic outline. DISCNET applies an approach similar to a Benders decomposition (cf. Benders [15, 16]), splitting the program into two parts. As first part, a master integer linear program comprises the objective (3.4) and the hardware dimensioning constraints (3.5) and (2.5). The second part models survivable (fractional) routings by constraints (2.7) and (2.24) and is used as a decision subproblem. Due to link-oriented node dimensioning, these partial programs are only coupled by the auxiliary link capacity variables y_ℓ. In contrast to Benders original approach, the subproblem is solved as linear program in our case. Figure 3.5 illustrates the decomposition and sketches also the basic outline of the solution algorithm which works as follows.
The central procedure is a branch-and-cut method based on the linear relaxation of the master program. Throughout the process, the current relaxation is strengthened by separation of cutting planes, such as Gomory mixed-integer rounding cuts (cf. Marchand and Wolsey [108]), band inequalities (cf. Dahl and Stoer [35]), GUB cover inequalities (cf. Wolsey [170]), and generalizations (cf. Wessäly et al. [168]) of metric inequalities (cf. Iri [72]). Whenever an integer hardware solution has been found in the branch-and-bound tree, the corresponding link capacities are handed

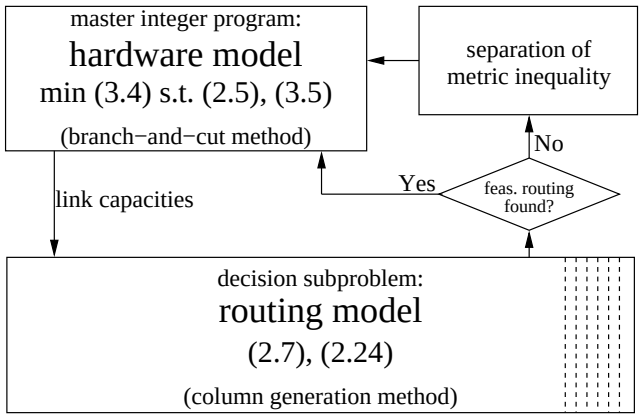


Figure 3.5: *Algorithmic approach for dimensioning and routing by* DISCNET.

over as parameters to the decision subproblem in order to verify whether these capacities admit a feasible routing or not. If the subproblem turns out as infeasible, the dual objective function of the routing linear program enables to derive a violated metric inequality which is added to the master program and cuts off the infeasible hardware configuration. Otherwise, a routing has been found and completes, added to the current hardware dimensioning, a feasible solution of the relaxed problem.
Running this procedure until the global lower and upper bounds meet ensures to end up with an optimal solution of the relaxed problem with fractional routings. This however might become a very time-consuming task since the branch-and-bound tree to be searched typically grows extremely fast for increasing problem size. To allow for a scaling of the computational effort, a couple of stopping criterions can be set, e.g., a total time limit, a gap limit, or a tree node number limit, among others. If such a criterion applies, the method terminates and returns (an adjustable number of) the best solution(s) found as well as the final lower bound on the total cost.

Routing subproblem. Given link capacities as parameters, the subroutine has to decide whether a feasible fractional routing exists within these capacities. Since no objective for the routing program is specified, a common trick is to turn the feasibility dichotomy into a threshold of an additional variable which at the same time guarantees for a solution of the modified program. To this end, the routing model is extended by a single unbounded capacity extension variable $y \in \mathbb{R}_+$ added to each link capacity, i.e., replacing y_ℓ by $y_\ell + y$ in constraints (2.7b) for all links $\ell \in L$, and its value is minimized by the objective $\min y$. An optimal solution of this extended linear program answers the question for feasibility by the objective value. An optimum value of zero indicates that indeed a feasible routing exists within the given capacities. Otherwise, the current capacities are insufficient for the demanded traffic, and the dual solution implies a feasible metric inequality as a Benders cut which is added to the master program.
The modified routing subproblem is solved by a column generation approach. The associated pricing problem consists of a shortest path computation (for each commodity) with weights computed by use of the dual variable values. Without further restrictions, the pricing is polynomially solvable. For single-hop transparent networks, however, limited path lengths have to be taken into account, adding further constraints to the pricing problem. Hop-count limits affect the complexity only marginal and leave the shortest path computation polynomial (cf. Garey and Johnson [52]), whereas a limited total distance with respect to individual link lengths turns the pricing into a pseudo-polynomial shortest path algorithm (cf. Handler and Zang [65]).

Integer routings. The proposed algorithm provides an exact method for the relaxed problem with fractional routings. To generate integer routings, two heuristic approaches can be followed. One idea is to run Discnet as described and to turn the obtained solutions into integral ones in a postprocessing (cf. Section 3.3.3). Here, we discuss the alternative to integrate such a heuristic search into Discnet's method which is offered optionally and returns the best integer solution(s) found.
During the branch-and-cut procedure, a routing computation is initiated for each identified integer hardware solution. Since existence of a feasible integer routing presumes that a fractional one is found, a search can be restricted to those subroutine calls that prove feasibility for the relaxed problem. Often, the feasibility

test already generates a routing that is nearly integral, i.e., only few path variables have fractional values. Hence, it is natural to focus on methods to adapt a fractional routing. DISCNET offers several heuristics for this, some with the emphasis on fast completion due to frequent execution, some more elaborate to achieve good integer solutions. For instance, it is possible to remove all fractional path flows, which leaves integer residual capacities, and to sequentially resettle their routings by min-cost flow computations with the residual capacities for free and additional capacity expansions (in integer amounts) against cost. With all capacities being integer, these flows become integer as well, and diversification conditions can be taken into account by bounding the usable capacities.
An advanced method is to solve the actual routing program for the removed flows and the resulting residual capacities as an integer linear program without further pricing, i.e., only by use of already generated routing paths. By adding appropriate capacity expansions variables individually for each link, an integer routing is completed against minimum cost increase (for this set of routing paths). This method applies in particular for the transparent single-hop scenario due to implicit selection of feasible paths only.
At last, a very fast heuristic with specific properties is highlighted. Instead of rerouting, the method simply rounds up all fractional path variables and, if necessary, adjusts the dimensioning in case of exceeded capacities. This way, we obtain an integer solution with more than the required number of connections routed for some commodities. While just this could be acceptable, diversification restrictions might be violated additionally. Nevertheless, we will show in the postprocessing that such integer routings can be fixed to meet both the actual connection number and the diversification requirements, except for the transparent single-hop case. However, if survivability is attained by the most general DSP concept, all corresponding routing constraints remain satisfied by the rounding for all scenarios, in particular for the transparent single-hop case as well. Hence, the rounding heuristic is the favorite method for these settings.

In either way of application with fractional or integer routings, DISCNET constitutes a suitable algorithmic framework for the dimensioning and routing subproblem and features a twofold outcome: a valid lower bound on the total cost for any network design satisfying all requirements, and a pool of the best found solutions. Making these solutions feasible for the original settings is accomplished by further procedures, as described next.

3.3.3 Accomplishing solutions

Although DISCNET can be applied to generate solutions with an integer routing, it might be reasonable to investigate dimensioning and routing with fractional flows at first instance. Without spending effort on heuristics to achieve integrality on the run, a larger portion of the solution space can be browsed within the same time limit, and better solutions could be achieved if a postprocessing ensures integer routings in the end. Both approaches can also be run competitively.
Moreover, regenerators have been excluded from transparent multi-hop networks initially to unify the dimensioning and routing subproblem. As a consequence, con-

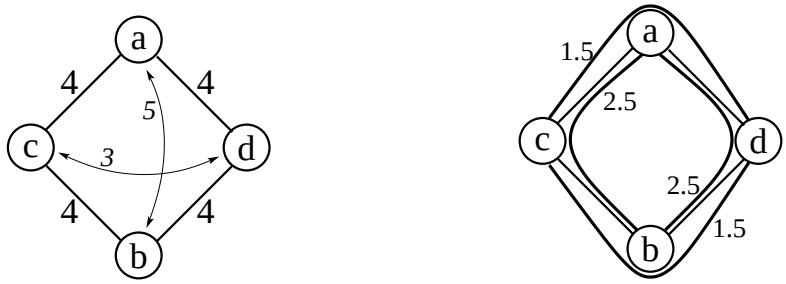
nections can become too long for direct optical transmissions, and thus regenerators have to be placed for a proper decomposition into consecutive lightpaths which meet the length restrictions.

For both issues, we present a selection of heuristics for transforming a solution obtained from Discnet into one with the desired properties. In order not to delay the computation too much, the heuristics have been designed for a fast completion of good solutions for practical instances. The development of more elaborate methods is left for further research.

3.3.3.1 Generating integer routings

In general, the problem of generating feasible integer routings looks different for any network architecture and any survivability scheme, due to the specific requirements imposed by both issues. In the previous section, some possible mechanisms have already been sketched for integrated use in Discnet. Any of these heuristics can in principle be applied as a postprocessing algorithm as well (and vice versa), but some exploit information that is most easy accessible on the run. In transparent single-hop networks, for instance, only routing paths of limited lengths are selectable. During execution of Discnet, most promising candidates for such paths are continuously priced out and hold in a collection that is regularly updated. Hence, it is preferable to benefit from these efforts for immediate generation of suitable integer routings rather than trying to regain comparable knowledge afterwards from scratch. We therefore advise use of integrated heuristics for the transparent single-hop scenario and neglect this complicating setting in the following.
Moreover, the method of rounding up fractional flows has been proposed as a simple and very fast heuristic to generate integer routings. In particular, this method maintains feasibility and thus is preferable for the most general DSP variant, which is not discussed further. Instead, we focus on the other DSP schemes with prespecified demand value $\widetilde{v}_q$ for each commodity $q \in \mathcal{Q}$ and present a method that turns a fractional routing into a feasible integer one by rerouting fractional path flows commodity-wise.

Rerouting fractional flows. The algorithm follows a simple outline. At first, we remove all fractional path flow portions and fix the remaining integer routing parts, thereby obtaining integer residual capacities. These residual capacities need not be sufficient to accommodate an integer routing. Figure 3.6 shows an example where a fractional, but no integer routing can be achieved within the capacities, even if no survivability restrictions are involved. Hence, we allow for capacity expansion against cost by appropriate transformations of the supply network to a directed network where all capacities and costs are attached to arcs. In the transformed network, we then reroute the removed flow sequentially for each commodity by a min-cost flow computation with individual capacities reflecting the diversification requirements. Since all these capacities and the flow to reroute are integer, theory guarantees for the existence of an optimal integer flow (see Ahuja et al. [2], Section 11.5), and several min-cost flow algorithms provide such solutions on request. In addition, the cost objective directs the search towards small expansion costs.
Regarding the costs, some simplifications are made in order to keep the flow problem



(a) *Network topology with installed link capacities and demands (as arrows).*

(b) *Fractional routing solution with flow values at each path.*

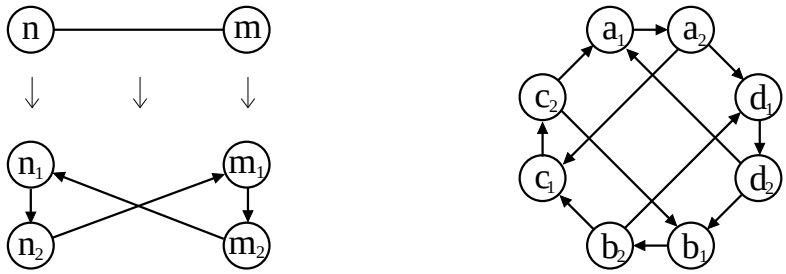
Figure 3.6: *Example of a network design problem with two commodities, between a and b of value 5, and between c and d of value 3. Within the installed link capacities 3.6(a) (node capacities are neglected), a fractional routing 3.6(b), but no integer routing solution exists.*

polynomial. First, we neglect potential savings by capacity cutbacks, but take only additional cost for required capacity extensions into account. Second, if such an expansion to a next higher capacity level provides more than one additional channel, a correct formulation would result in a min-cost fixed-charge network flow problem, which is $\mathcal{NP}$-hard (see Garey and Johnson [52], where it is called minimum edge-cost flow). To avoid this, the cost for any such capacity expansion is fully accounted for each additional unit of flow, thus oversizing the effectively implied expansion costs. And third, node expansions are initiated in a traffic-oriented way similar to link capacity upgrades. Altogether, such a min-cost flow need not be a (re-)routing that in fact causes the minimum capacity extension cost. For a heuristic, however, we are content to approximate these values, where good solutions with cost overestimation are hopefully good as well in effective costs, and in case we end up with an integer flow for all commodities at zero total cost, such a solution is provably optimal.
In detail, the heuristic works as follows.

Integer routing generation from fractional flows

Input. Given is an integer hardware configuration of the supply network $\mathcal{N} = (N, L)$, with the currently selected capacities given by the variable values y_ℓ, $y_n \in \mathbb{Z}_+$ for all nodes $n \in N$ and supply links $\ell \in L$, and a fractional routing expressed by path flows $f_p^q \in \mathbb{Q}_+$ for all commodities $q \in \mathcal{Q}$ and all paths $p \in \mathcal{P}$.
In addition, we have for each link or node $\pi \in N \cup L$ the discrete set $K_\pi = \{(k_1^\pi, C_1^\pi), \ldots, (k_{m_\pi}^\pi, C_{m_\pi}^\pi)\}$ of installable capacities and their costs (or compute these sets by the methods described in Section 3.3.1). In each of these sets, the index $j_\pi \in \{1, \ldots, m_\pi\}$ marks the element corresponding to the selected capacity, i.e., $y_\pi = k_{j_\pi}^\pi$ for all $\pi \in N \cup L$.

Method. At first, we remove the fractional portion $f_p^q - \lfloor f_p^q \rfloor$ of any non-integer path flow $f_p^q \notin \mathbb{Z}$ and freeze all remaining integer path flows. For each commodity $q \in \mathcal{Q}$, the total removed flow value $\rho_q := \sum_{p \in \mathcal{P}_q} (f_p^q - \lfloor f_p^q \rfloor)$ is integer as difference of two integer values, the number of connections to establish and the total value of the fixed flow part. Moreover, the flow



(a) *Link transformation.* (b) *Network transformation.*

Figure 3.7: *Demonstration of the network transformation for a single link and its endnodes (at left) and for the complete network from Figure 3.6 (at right).*

reduction leaves on each link or node $\pi \in N \cup L$ the residual capacity

$$\widetilde{k}_\pi := y_\pi - \sum_{q \in \mathcal{Q}} \sum_{\substack{p \in \mathcal{P}_q: \\ \pi \in N[p] \cup L(p)}} \lfloor f_p^q \rfloor$$

which is obviously integer as well.
Next, the supply network $\mathcal{N}$ is transformed to a directed topology $\vec{\mathcal{N}}$ as follows. Each node $n \in N$ is split into a pair n_1, n_2 of nodes with a connecting *node arc* (n_1, n_2), and each supply link $\ell = mn \in L$ is substituted by the two *link arcs* (m_2, n_1) and (n_2, m_1). Formally, the directed topology $\vec{\mathcal{N}} = (N', A)$ is defined by

$$\begin{aligned} N' &:= \{\, n_1, n_2 \mid n \in N \,\} \\ A &:= A_N \cup A_L \qquad \text{with} \quad A_N := \{\, (n_1, n_2) \mid n \in N \,\} \\ &\qquad\qquad\qquad\quad A_L := \{\, (m_2, n_1), (n_2, m_1) \mid \ell = nm \in L \,\} \end{aligned}$$

Figure 3.7 illustrates the transformation for an individual link with the incident nodes and for an entire exemplary ring network with four nodes. Then, we process all reduced commodities $q \in \mathcal{Q}$ with $\rho_q > 0$ sequentially in an arbitrary order. For each such commodity q, the procedure consists of three steps: individual adaption of $\vec{\mathcal{N}}$ to an extended network $\vec{\mathcal{N}}'$ including capacity expansion capabilities, min-cost flow computation in $\vec{\mathcal{N}}'$ to complete the integer routing for q, and reduction of the residual capacities for the remaining commodities to be processed. These steps are next described in detail for an arbitrary, but fixed commodity q from $o = o_q$ to $d = d_q$ being on turn.
Taking the prefixed routing part for q into account, the diversification requirements imply that at most $\bar{k}_\pi := \widetilde{\delta}_q \widetilde{v}_q - \sum_{p \in \mathcal{P}_q : \pi \in N(p) \cup L(p)} \lfloor f_p^q \rfloor$ further routing paths can traverse link or node $\pi \in L \cup N \setminus \{o_q, d_q\}$. When such a value exceeds the available residual capacity at a place, i.e., $\bar{k}_\pi > \widetilde{k}_\pi$, the corresponding arc is copied as many times as further upgrade stages must be provided until capacity $\bar{k}_\pi$ is finally reached. In general, the following transformation is applied to the single arc (n_1, n_2) for a node $\pi = n \in N$ or in the same way for each of the arcs (n_2, m_1), (m_2, n_1) for a link $\pi = \ell = nm \in L$. The original arc, indexed by 0, gets the capacity bound $\bar{k}^0_\pi := \min\{\widetilde{k}_\pi, \bar{k}_\pi\}$, thus holding only residual capacity,

with zero flow unit cost $C^0_\pi := 0$, and we define $\underline{k}^0_\pi := \bar{k}_\pi - \bar{k}^0_\pi \geq 0$ as remaining exploitable capacity to be provided by upgrades with incremental amounts. If $\underline{k}^0_\pi = 0$, the residual capacity already suffices. Otherwise, for $i = 1, 2, \ldots$, we insert an i-th arc copy with capacity bound $\bar{k}^i_\pi := \min\{\underline{k}^{i-1}_\pi, k^\pi_{j_\pi+i} - k^\pi_{j_\pi+i-1}\}$ and flow unit cost $C^i_\pi := C^\pi_{j_\pi+i} - C^\pi_{j_\pi}$. Note that thereby the cost increases with i so that the represented capacity upgrades are occupied in the correct order. We set $\underline{k}^i_\pi := \underline{k}^{i-1}_\pi - \bar{k}^i_\pi$ and iterate as long as $\underline{k}^i_\pi > 0$ and $j_\pi + i < m_\pi$. After all, the resulting network with all multiplied arcs is referred to as $\vec{\mathcal{N}}' = (N', A')$.
In this extended network $\vec{\mathcal{N}}'$, an integer min-cost flow $\tilde{f}$ of value ρ_q from o_1 to d_2 is computed. Appropriate algorithms can be found, for instance, in Ahuja et al. [2]. After removing directed cycles if present, the flow $\tilde{f}$ can be decomposed (arbitrarily) into simple routing paths p' in $\vec{\mathcal{N}}'$ with integral flow, denoted by $\tilde{f}^q_{p'}$. In a straightforward way, any routing path p' in $\vec{\mathcal{N}}'$ corresponds to a path p in the original network $\mathcal{N}$ by

$$\begin{aligned} n \in N[p] &\quad\Leftrightarrow\quad (n_1, n_2) \in A'(p') \\ \ell = mn \in L(p) &\quad\Leftrightarrow\quad ((m_2, n_1) \in A'(p')) \vee ((n_2, m_1) \in A'(p')) \end{aligned}$$

and we set $\tilde{f}^q_p = \tilde{f}^q_{p'}$ to get the indicated min-cost flow routing in $\mathcal{N}$ for q. In combination with the previously fixed routing part, the total path flow $\sum_{p\in\mathcal{P}_q} \tilde{f}^q_p + \lfloor f^q_p \rfloor = \widetilde{v}_q$ has the requested value and satisfies all diversification constraints due to the definition of $\bar{k}_\pi$ for each link and node $\pi \in N \cup L$, thus forms a feasible integer routing for q. So, we fix the new flow $\tilde{f}^q_p$ as routing completion for q, i.e., the full routing for q is expressed by $\lfloor f^q_p \rfloor + \tilde{f}^q_p$ for all paths $p \in \mathcal{P}_q$.
Before turning to the next commodity to process, we have to adapt the network capacities according to the added fixed flow and possibly required capacity upgrades. For each link $\ell \in L$, let i_ℓ be the highest index of a parallel arc used by $\tilde{f}$. This indicates that the link capacity y_ℓ is increased by an amount of $\Delta_\ell := k^\ell_{j_\ell+i_\ell} - k^\ell_{j_\ell}$ at an effective cost of $C^\ell_{j_\ell+i_\ell} - C^\ell_{j_\ell}$. After finishing all link upgrades, the node capacities are considered, but without using parallel arc occupation as indicator. Instead, the new link capacities imply the capacity requirements at the nodes. For this, we determine the smallest indices $i_n \geq 0$ for all nodes $n \in N$ such that the node dimensioning constraints (2.5) on page 54 are satisfied with $y_\ell := k^\ell_{j_\ell+i_\ell}$ and $y_n := k^n_{j_n+i_n}$. Hence, the capacity y_n at a node n is increased by an amount of $\Delta_n := k^n_{j_n+i_n} - k^n_{j_n}$ at an effective cost of $C^n_{j_n+i_n} - C^n_{j_n}$. From the resulting capacities, the routing completion occupies at each node or link $\pi \in N \cup L$ a value of $\tilde{f}_\pi := \sum_{p\in\mathcal{P}:\pi\in N[p]\cup L(p)} \tilde{f}^q_p$. As a consequence, we reset the residual capacities to the new value $\widetilde{k}_\pi + \Delta_\pi - \tilde{f}_\pi$ and the indices j_π to $j_\pi + i_\pi$ for all nodes and links $\pi \in N \cup L$, and then proceed with the next commodity on turn.

Output. After termination, the method returns a feasible integer path routing $\tilde{f}^q_p + \lfloor f^q_p \rfloor$ for each commodity $q \in \mathcal{Q}$ and new capacities $y_\pi := k^\pi_{j_\pi}$ for any node or link $\pi \in N \cup L$. The total cost increase caused for the routing integrality is the sum over all effective capacity upgrade costs generated throughout the procedure.

In general, this sequential rerouting method by min-cost flows can fail at some point when capacities and their expansion capabilities are strictly limited. Though such instances are not hard to construct, we have rarely observed the algorithm failing for practical instances. If this occurs, it often helps to process the commodities in a different order. Another idea is to vary the assigned costs such that very scarce capacities at bottlenecks become more expensive. A similar variation allows also to seek for improving solutions with an already integer routing.

Improving integer routings. When integer routings are generated heuristically, the obtained solutions need obviously not be optimal. Hence, it might be possible to achieve capacity and thus cost savings by varying the commodity routings. For this, we briefly sketch an appropriate adaption of the previous algorithm that attempts to reroute commodities such that potential capacity cutbacks can be realized.
Basically, the improvement heuristic works in the same way as before, considering the commodities sequentially. This time, however, we remove the entire path flows of the commodity q on turn, while fixing those of all other commodities. Thereby, the value $\widetilde{v}_q$ of the removed flow and all residual capacities are integer, and we can again use a min-cost flow computation for the rerouting. Moreover, we additionally know that capacity supplements will not be required, since a feasible integer routing within the current capacities y_π for all $\pi \in N \cup L$ exists. So, we use the cost function to indicate that potential cost *savings* by declining to lower capacity levels are *not* realized.
To this end, we determine the smallest discrete capacities $k^\pi_{j_\pi}$ ($\leq y_\pi$) at each node or link π required for the fixed routing of all other commodities. With $\bar{k}_\pi := \widetilde{\delta}_q \widetilde{v}_q$, the extended network is constructed by multiplying arcs as before, but allowing for upgrades only up to the input capacities y_π since we focus solely on potential cutbacks and do not allow for further expansions. Regarding the costs, $k^\pi_{j_\pi}$ now mark the free capacities, and we assign costs to parallel arcs representing upgrade stages in relation to $k^\pi_{j_\pi}$. Then a rerouting for q is computed as min-cost flow and yields finally required capacities $k^\pi_{j_\pi+i_\pi}$. Whenever $k^\pi_{j_\pi+i_\pi} < y_\pi$ holds for some node or link π, an improved solution has in fact be found.
This basic improvement method is further extendible in various ways. For instance, the capacity upgrade capabilities need not be limited up to the given capacities. This way, further expansions can be found at some places with a total cost less than the total savings by potential cutbacks at other places. Moreover, note that the improvement algorithm iteratively generates feasible solutions, i.e., with an entirely integer routing and appropriate capacities after each commodity rerouting. Hence, the algorithm can be terminated after each iteration. Alternatively, it can be run as an iterative search heuristic, revisiting commodities multiple times in any suitable order or by dynamically selecting a most promising commodity next. Furthermore, the improving algorithm suits also for fixing integer flows generated with a rounding procedure (which might hold more connections than requested).

Fixing rounded flows. Among the heuristics that Discnet applies internally, we have stressed the naive, but very fast procedure to make a fractional routing integer by simply rounding up each fractional flow to the next integer value, followed by proper adjustment of the required capacities. This way, some commodities get more connections than initially demanded, which is only feasible for the most general DSP

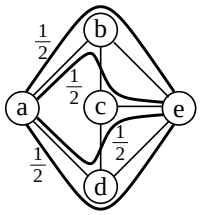


Figure 3.8: *Example for infeasible routing by flow rounding. For a commodity q requesting one protected connection between the 2-connected nodes a and e, we route in total 2 connections, at most one traversing each link or node except for a and e. The displayed routing with a flow of $\frac{1}{2}$ on each path is feasible, but rounding up each path flow to 1 violates several diversification constraints. With the same commodity parameters, the routing shown in Figure 2.5(c) on page 77, again with flow $\frac{1}{2}$ on each path, yields similar violations, and even more, no cardinality 2 subset of the rounded routing paths is a feasible integer routing.*

concept. For other DSP variants, diversification constraints might become violated as well, and it can even happen that no subset of the routing paths obtained by rounding forms a feasible routing. Figure 3.8 provides examples for such violations. Nevertheless, the following lemma shows that a feasible integer routing of the originally requested demand value exists within the capacities occupied by the rounded flow. Note that it suffices to consider the link capacities since the corresponding node capacities are dimensioned in a link-oriented way and thus any routing not exceeding the link capacities does also not exceed the node capacities.

Lemma 3.5 *For a commodity $q \in \mathcal{Q}$, let f_p^q for all $p \in \mathcal{P}_q$ denote an arbitrary feasible fractional routing of total value $\widetilde{v}_q$, i.e., satisfying all corresponding flow and diversification constraints. Then there exists a feasible integer routing that does not exceed the link capacities $y_\ell^q := \sum_{p \in \mathcal{P}_{od} : \ell \in L(p)} \lceil f_p^q \rceil$ reserved for q by the rounding procedure.*

Proof. Assume such a feasible integer routing does not exist. Consider a maximum flow from o_q to d_q with respect to the link capacities $\min\{\widetilde{\delta}_q \widetilde{v}_q, y_\ell^q\}$ for all $\ell \in L$. Since all these capacities are integer (by (2.21)), there exists also an integer maximum flow (cf. Nemhauser and Wolsey [127], Part I.3.4). If such a maximum flow would have a value of at least $\widetilde{v}_q$, (part of) this flow would provide a feasible integer routing for q. By the assumption, the maximum flow value must hence be strictly less than $\widetilde{v}_q$. From the max-flow min-cut theorem (cf. Ford and Fulkerson [50]), we know that there exists an o_q, d_q-cut $\Gamma \subset L$ with a total capacity equal to the maximum flow value, i.e., $\widetilde{v}_q > \sum_{\gamma \in \Gamma} \min\{\widetilde{\delta}_q \widetilde{v}_q, y_\gamma^q\}$.
By prerequisite, the fractional flow f has value $\widetilde{v}_q$, which implies that $\widetilde{v}_q \leq \sum_{\gamma \in \Gamma} \sum_{p \in \mathcal{P}_q : \gamma \in p} f_p^q$ holds for the cut Γ. Due to its feasibility, the fractional flow also satisfies the survivability restriction $\sum_{p \in \mathcal{P}_q : \gamma \in p} f_p^q \leq \widetilde{\delta}_q \widetilde{v}_q$ at each cut element $\gamma \in \Gamma$, and since the reserved capacities base on rounding up each path flow, we have $\sum_{p \in \mathcal{P}_q : \gamma \in p} f_p^q \leq y_\gamma^q$, combining to $\sum_{p \in \mathcal{P}_q : \gamma \in p} f_p^q \leq \min\{\widetilde{\delta}_q \widetilde{v}_q, y_\gamma^q\}$ for all $\gamma \in \Gamma$. With these inequalities, we finally obtain the contradiction

$$\widetilde{v}_q \leq \sum_{\gamma \in \Gamma} \sum_{p \in \mathcal{P}_q : \gamma \in p} f_p^q \leq \sum_{\gamma \in \Gamma} \min\{\delta_q v_q, y_\gamma^q\} < \widetilde{v}_q \; .$$

■

Lemma 3.5 implies that a network design solution with capacities sufficient for an integer routing that results from rounding up a feasible fractional routing can be afterwards transformed into a solution with a feasible integer routing without further capacity expansions. For this, the previously described integer routing improvement method can be used, since capacity supplements will never be required by Lemma 3.5. In fact, the feasible integer routing can free previously occupied capacities due to the reduced total flow amount, and thus the solution cost can decrease by the postprocessing.
As a result, the flow rounding heuristic can be used in Discnet without the risk to produce infeasibility. Since rounding and capacity adaptions are fast to perform, integer solutions can be provided on the run with marginal effort, using the fixing procedure in the end only for the best found rounded solution(s). Alternatively, the rerouting can also be applied after each rounding, trying to improve the integer solution immediately. This modus operandi might provide a better final result against additional effort.

To conclude, the provided methods serve to generate or improve an integer routing for any scenario. This completes a solution of the dimensioning and routing subproblem except for the transparent multi-hop case which requires the check for regenerator placements.

3.3.3.2 Regenerator placement

In the following, we focus on the transparent multi-hop scenario where regenerators have been initially excluded in order to simplify the routing model and to unify a core task for all scenarios. Given a previously generated integer routing, regenerators have now to be placed such that the connections decompose into lightpaths which meet the length restrictions. The objective is to keep the additionally implied total regeneration cost as low as possible. For this task, we present two generic combinatorial heuristics.
We apply a simplified module and cost model for regenerators, similar to that for wavelength converters which will be extensively discussed in the next chapter. In a nutshell, we assume to have a single regenerator type $\mathcal{R} = \{r\}$ with the same cost C^r and unbounded applicability $b_n^r = \infty$ throughout the network and movable preinstalled modules. For this setting, minimizing the total regeneration cost is equivalent to minimizing the total number of required regenerators, and the latter objective is used for both proposed methods. The methods differ in the extent to which the integer routing given as input can be manipulated.

Supply network based method. The first method gets a connection routing specified on the supply network and shall not change any of these end-to-end paths. We remind that the supply network has been constructed such that all transmission systems at a supply link have the same propagation properties. For this setting, the placement of regenerators for each connection is independent from all other connections, and a total minimum number of regenerators is achieved if any individual connection does not use more regenerators than necessary. The routing paths are considered individually to determine such an optimal regenerator placement.

Algorithm 3.2 Routing path decomposition into lightpaths

Require: A routing path $p = (n_0, \ell_1, n_1, \ldots, \ell_h, n_h)$ in supply network $\mathcal{N} = (N, L)$ and the maximum lightpath length Ω.

Ensure: A decomposition of p into a sequence $s_1, \ldots, s_{j_p}$ of lightpath routes.

1: set $j := 1$, $j_p := 1$ {initialization}
2: **while** $j \leq h$ **do**
3: determine a maximum $j' \in \mathbb{Z}_+$ such that
$$\left(\sum_{i=j}^{j+j'} \omega_{\ell_i} \leq \Omega \right) \wedge \big(j + j' \leq h \big)$$
4: set $s_{j_p} := (n_{j-1}, \ell_j, n_j, \ldots, \ell_{j+j'}, n_{j+j'})$
5: $j \leftarrow j + j' + 1$
6: $j_p \leftarrow j_p + 1$
7: **end while**
8: return $s_1, \ldots, s_{j_p}$

Regenerator placement for routing in supply network

Input. Given is a supply network $\mathcal{N} = (N, L)$ with sufficiently dimensioned transmission and switching capacities, a total number $e^{\mathcal{R}} \in \mathbb{Z}_+$ of preinstalled regenerators, and an integer routing of all connections to establish denoted by the path flow variables f_p^q for all commodities $q \in \mathcal{Q}$ and all paths $p \in \mathcal{P}$.

Method We begin with initializing the numbers of required regenerators by $y_n^{\mathcal{R}} := 0$ for all nodes $n \in N$. The flow variables $f_p^q > 0$ indicating used routing paths are then processed sequentially in an arbitrary order.
Let p be the routing path of the variable currently on turn. If $p \in \widetilde{\mathcal{P}}$, i.e., the path has a total length of $\sum_{\ell \in L(p)} \omega_\ell \leq \Omega$, the associated connections can be established as a single lightpath, and we leave the variables unchanged. Otherwise, $p \notin \widetilde{\mathcal{P}}$, thus these connections exceed the maximum optical transmission distance and have to be split into a sequence of lightpaths. For this, we apply the method listed in Algorithm 3.2 which determines a decomposition of p into a sequence of consecutive subpaths $s_1, \ldots, s_{j_p} \in \widetilde{\mathcal{P}}$ that match the lightpath length restriction. This decomposition is incorporated into the actual solution by the following variable adaptions. For placing the required regenerators, we add the value of f_p^q to $y_n^{\mathcal{R}}$ for each node $n \in \{d_{s_1}, \ldots, d_{s_{j_p-1}}\}$ as an intermediate node between two consecutive lightpaths of the decomposition. Moreover, we increase the values of the routing variables $f_{s_1}^q, \ldots, f_{s_{j_p}}^q$ by the value of f_p^q. Then we set $f_p^q = 0$ to mark that these routing paths have been replaced.
After having processed the given routing this way, all flow variables f_p^q for paths $p \notin \widetilde{\mathcal{P}}$ have been vanished, i.e., have value 0.

Output. When finished, the method returns the required number $y_n^{\mathcal{R}}$ of regenerators at each node $n \in N$ and a feasible lightpath routing denoted by f_p^q for all commodities $q \in \mathcal{Q}$ and all lightpath routes $p \in \widetilde{\mathcal{P}}$ such that all constraints (2.16a) and (2.16c) hold. The total cost incurred for regenerations in this solution is given as $\max\{0, C^r \cdot (-e^{\mathcal{R}} + \sum_{n \in N} y_n^{\mathcal{R}})\}$.

The following lemma shows that Algorithm 3.2 decomposes a path p in fact into a minimum number of lightpaths.

Lemma 3.6 *For any routing path p in the supply network $\mathcal{N} = (N, L)$, Algorithm 3.2 returns a lightpath decomposition of minimum cardinality.*

Proof. Due to selecting the maximum value for j', Algorithm 3.2 ensures that the first subpath cannot be extended by further links without exceeding the lightpath length restriction. Hence, in any proper decomposition of p, the first regenerator(s) cannot be placed later than in node d_{s_1}. The same argument holds iteratively for all consecutively following subpaths and their terminating regeneration nodes, too. This implies that any decomposition needs at least the same number of regeneration nodes along path p as established by Algorithm 3.2, proving the claim. ■

As a consequence, the proposed method generates a solution with a minimum total number of regenerators when the connection routing in the supply network is prespecified and not to be changed. However, allowing for routing manipulation of the given connections widens the solution space for regenerator placement and generation of a feasible lightpath routing in the transparent multi-hop scenario.

Physical topology based method. Without imposing path length restrictions, the dimensioning and routing subproblem for transparent multi-hop networks transforms to that for opaque networks without parallel supply links corresponding to the same physical link, as described in Section 3.3.1. This way, the connections are routed directly in the (possibly refined) physical topology $\mathcal{T} = (N, T)$, and the supply network with parallel links L_t for each physical link $t \in T$ is constructed *after* dimensioning of the link capacities. For accomplishing the solution, an assignment of the connections to a specific supply link on each traversed physical link then has to be carried out in addition. In relation to the previous method, this corresponds to allow accessorily for a connection regrouping on the supply links associated to the same physical link.
The second heuristic has been designed for this extended task with a physical topology based routing as input and consists of a two-step procedure. In the first step, we place the minimum number of unavoidable regenerators on each routing path, thereby assuming that the shortest parallel supply link can be used on each physical link. This way, we get two results: a lower bound on the total number of required regenerators, and a candidate multi-set of potential lightpaths to accommodate the routing. In the second step, these candidates are subsequently assigned to specific supply links with capacity left, using longest possible alternative links as long as not exceeding the lightpath length restriction. Additional regenerators might become necessary if the lightpath candidates become too long nonetheless.

Regenerator placement for routing in physical topology

Input. Given is the supply network $\mathcal{N} = (N, L)$ with link capacities $y_\ell > 0$ for all $\ell \in L$ and sufficiently dimensioned nodes, as well as a routing of all connections in the **physical** topology $\mathcal{T} = (N, T)$ denoted by variables $f_{\overline{p}}^{q}$ for paths $\overline{p} = (n_0, t_1, n_1, \ldots, t_{h_{\overline{p}}}, n_{h_{\overline{p}}})$.

Method During the procedure, we use R as lower bound on the number of unavoidable regenerators and the auxiliary values k_ℓ for all $\ell \in L$

Algorithm 3.3 Routing path decomposition into balanced lightpath candidates

Require: A routing path $p = (n_0, \ell_1, n_1, \ldots, \ell_h, n_h)$ in supply network $\mathcal{N} = (N, L)$, and the number $R_p \geq 1$ of regenerators to place.

Ensure: A decomposition of p into a sequence $s_1, \ldots, s_{R_p+1}$ of lightpath routes such that $\max_{i=1,\ldots,R_p+1} \omega_{s_i}$ is minimal, where $\omega_s := \sum_{\ell \in L(s)} \omega_\ell$ denotes the length of a subpath s.

1: set $\omega(j,1) := \sum_{m=1}^{j} \omega_{\ell_m}$ for all $j = 1, \ldots, h$ {initialization}
2: **for** $j = 2, \ldots, h$ **do**
3: **for** $i = 2, \ldots, \min\{j, R_p + 1\}$ **do**
4: set $\omega(j,i)$ by (3.7), and let $n(j,i) := j'$ be an index for which the minimum in (3.7) is taken
5: **end for**
6: **end for**
7: set $j_d := n_h$
8: **for** $i = R_p + 1, R_p, \ldots, 2$ **do**
9: set $j_o := n(j_d, i)$
10: set $s_i := (n_{j_o}, \ell_{j_o+1}, n_{j_o+1}, \ldots, \ell_{j_d}, n_{j_d})$
11: set $j_d := j_o$
12: **end for**
13: return $s_1, \ldots, s_{R_p+1}$

to keep track of still unused link capacities. We initialize the variables $y_n^{\mathcal{R}} := 0$ for all nodes $n \in N$, the lower bound by $R := 0$, and the capacities with $k_\ell := y_\ell$ for all $\ell \in L$. Moreover, we apply the supply link selection functions $\underline{\ell}(t) := \arg\min \{\, \omega_\ell \mid \forall\, \ell \in L_t \,:\, k_\ell > 0 \,\}$ and $\overline{\ell}(t) := \arg\max \{\, \omega_\ell \mid \forall\, \ell \in L_t \,:\, k_\ell > 0 \,\}$ to get the shortest respectively longest still available supply link on a physical topology link, as well as the functions $\underline{\omega}(\overline{p}) := \sum_{t \in T(\overline{p})} \omega_{\underline{\ell}(t)}$ and $\overline{\omega}(\overline{p}) := \sum_{t \in T(\overline{p})} \omega_{\overline{\ell}(t)}$ for the minimum respectively maximum length of a physical topology path $\overline{p}$ when mapped into the (residual) supply network.

Step 1: At first, we place unavoidable regenerators by splitting each path into candidate lightpaths. Due to always selecting the shortest alternative supply links, these computations are independent of each other, and the variables $f_{\overline{p}}^q > 0$ can be processed in an arbitrary order.

Let $\overline{p} = (n_0, t_1, n_1, \ldots, t_h, n_h)$ be the path of an arbitrary variable on turn. Then we define an associated shortest supply network routing path as $p := (n_0, \underline{\ell}(t_1), n_1, \underline{\ell}(t_2), n_2, \ldots, \underline{\ell}(t_h), n_h)$ and use Algorithm 3.2 to determine the minimum number $R_p := j_p - 1$ of regenerators required for p. If $R_p = 0$, the path p is a lightpath candidate itself, and we set $f_p^q = f_{\overline{p}}^q$. Otherwise, p must be accommodated by (at least) $R_p + 1$ consecutive lightpaths, and we increase R by R_p. The path decomposition provided by Algorithm 3.2 has the undesirable property that all subpaths are longest possible except for the last which might become very short. When finally fixing the link assignments, the long candidate subpaths probably become too long if the assumed shortest supply links were not available anymore. Therefore, we compute another decomposition with better balanced subpath lengths to lower the risk of needing additional regenerations.

For this, we apply Algorithm 3.3 encoding a dynamic programming ap-

proach to split p into $R_p + 1$ subpaths minimizing the maximum subpath length in the decomposition as objective. The algorithm iteratively evaluates the function $\omega(j, i)$ defined as the objective for any decomposition of the front path part $(n_0, \ell_1, n_1, \ldots, \ell_j, n_j)$, $j \geq 1$, into exactly i parts, with $1 \leq i \leq \min\{j, R_p + 1\}$. For $i = 1$, the value of $\omega(j, 1)$ is simply the total path part length, which is used for initialization. If $i > 1$, any decomposition of $(n_0, \ell_1, n_1, \ldots, \ell_j, n_j)$ into i parts must place the last regenerator in one of the nodes $n_{i-1}, \ldots, n_{j-1}$. Each regenerator place selection $n_{j'}$ fixes the last subpath (length), and the objective for any decomposition of the remaining front path into $i - 1$ parts is given by $\omega(j', i - 1)$, yielding the relation

$$\omega(j, i) = \min_{j'=i-1}^{j-1} \max \left\{ \omega(j', i-1), \sum_{m=j'+1}^{j} \omega_{\ell_m} \right\} \tag{3.7}$$

for all $j = 2, \ldots, h$ and all $i = 2, \ldots, \min\{j, R_p+1\}$. In the end, $\omega(h, R_p+1)$ holds the optimal objective value, and keeping further track of a minimum argument in (3.7) allows to reconstruct the corresponding decomposition. Note that the setting of R_p implies $\omega(h, R_p + 1) \leq \Omega$, but $\omega(h, i) > \Omega$ for all $i < R_p + 1$.

Algorithm 3.3 returns such a balanced decomposition of p, and the subpaths $s_1, \ldots, s_{R_p+1}$ are retranslated into the physical topology by replacing each link ℓ by the corresponding t with $\ell \in L_t$. This way, $\overline{p}$ is decomposed into the corresponding subpaths $\overline{s}_1, \ldots, \overline{s}_{R_p+1}$. For adaption of the routing, we add the value $f^q_{\overline{p}}$ to the variables $f^q_{\overline{s}_i}$ for all $i = 1, \ldots, R_p + 1$, and set $f^q_{\overline{p}} = 0$ afterwards. Moreover, we increment the variables $y^{\mathcal{R}}_n$ for each $n \in \{d_{s_1}, \ldots, d_{s_{R_p}}\}$ by 1, before proceeding with the next path.

Step 2: As result of the first step, R gives a lower bound for the number of unavoidable regenerators, and the variables $f^q_{\overline{p}}$ contain the candidate lightpath routings for all commodities. The second step fixes the assignment to specific supply links for each candidate lightpath, continuously updating the remaining supply link capacities. Since early occupation of short supply links increases the risk that later processed paths need additional regenerators, the lightpath candidates $\overline{p}$ are individually inserted into a processing priority queue ordered by non-decreasing length $\overline{\omega}(\overline{p})$, i.e., with respect to the total length for selecting the longest supply links for each traversed physical link.

Let $\overline{p} = (n_0, t_1, n_1, \ldots, t_h, n_h)$ be the next candidate lightpath to process (at first position in the queue). We remove it from the priority queue and decrement $f^q_{\overline{p}}$. Whenever $\overline{\omega}(\overline{p}) \leq \Omega$, we in fact assign on each hop the longest alternative supply link which is still available, i.e., we define $p := (n_0, \overline{\ell}(t_1), n_1, \ldots, \overline{\ell}(t_h), n_h)$, increment the variable f^q_p, and decrement $k_{\overline{\ell}(t_i)}$ for all $i = 1, \ldots, h$. In the other case, $\underline{\omega}(\overline{p})$ is first evaluated with respect to the remaining link capacities.

If $\underline{\omega}(\overline{p}) > \Omega$, any still possible link assignment yields a supply network path exceeding the maximum lightpath length (which can happen when shorter alternative supply links have already been occupied by previously processed lightpaths). Hence, $\overline{p}$ has to be decomposed further. Reusing

Algorithm 3.4 Supply link assignment for lightpath candidate

Require: A path $\overline{p} = (n_0, t_1, n_1, \ldots, t_h, n_h)$ in the physical topology $\mathcal{T} = (N, T)$, supply link capacities k_ℓ for all links $\ell \in L$ in the supply network $\mathcal{N} = (N, L)$, and the maximum lightpath length Ω.

Ensure: A supply network path $p = (n_0, \ell_1, n_1, \ldots, t_h, n_h)$ with $k_{\ell_i} > 0$ for all $i = 1, \ldots, h$ and maximum total length $\omega_p \leq \Omega$.

```
set p := (n_0, ℓ(t_1), n_1, ..., ℓ(t_h), n_h), ω := ω(p̄)
set P := {p}
for j = 1, ..., h do
    set P' := ∅
    ω ← ω − ω_{ℓ(t_j)}
    for all p ∈ P, p = (n_0, ℓ_1, n_1, ..., ℓ_h, n_h) do
        for all ℓ ∈ L_{t_j}, k_ℓ > 0 do
            set ω_{j−1}(p) := Σ_{i=1}^{j−1} ω_{ℓ_i}
            if ω_{j−1}(p) + ω_ℓ + ω ≤ Ω then
                set p' := (n_0, ℓ_1, n_1, ..., n_{j−2}, ℓ_{j−1}, n_{j−1}, ℓ, n_j, ℓ_{j+1}, n_{j+1}, ..., ℓ_h, n_h)
                P' ← P' ∪ {p'}
            end if
        end for
    end for
    set P := P'
end for
set p := arg max_{p'∈P} ω_{p'}
return p
```

Algorithm 3.3 from the previous step, we obtain a decomposition of $\overline{p}$ into subpaths $\overline{s}_1, \ldots, \overline{s}_{R_{\overline{p}}+1}$ which are inserted into the priority queue, and we also increment the regenerator counting variables $y^{\mathcal{R}}_{d_{\overline{s}_i}}$ for all $i = 1, \ldots, R_p$ as well as the flow variables $f^q_{\overline{s}_i}$ for all $i = 1, \ldots, R_p + 1$.
Otherwise, $\underline{\omega}(\overline{p}) \leq \Omega$ implies the existence of link assignments turning $\overline{p}$ into a feasible lightpath route. In order to avoid blocking of shorter supply links as far as possible, the fixed assignment is selected such that the resulting path has a maximum total length below Ω. For any path $\overline{p}$, Algorithm 3.4 computes such a link assignment by link-wise progressed enumeration of selectable alternatives and returns the corresponding path $p \in \widetilde{\mathcal{P}}$ in the supply network. This path is inserted into the lightpath routing by incrementing the variable f^q_p, while decrementing $k_{\overline{\ell}(t_i)}$ for all $i = 1, \ldots, h$ to account for the induced reduction of remaining capacities. Having $\overline{p}$ finished, the next path in the priority queue is processed until the queue is empty, terminating the method.

Output. After all, the outcome consists of the variables $y^{\mathcal{R}}_n$ for all nodes $n \in N$, representing the regenerator placement, and the variables f^q_p for all commodities $q \in \mathcal{Q}$ and lightpath routes $p \in \widetilde{\mathcal{P}}$, holding a feasible lightpath routing, such that all original constraints (2.16a) and (2.16c) are satisfied. As before, the total cost incurred for regeneration in this solution is given by $\max\{0, C^r \cdot (-e^{\mathcal{R}} + \sum_{n \in N} y^{\mathcal{R}}_n)\}$.

The two-step method provides an optimum solution if either the total cost is zero, or the total number $\sum_{n \in N} y_n^{\mathcal{R}}$ of placed regenerators coincides with the final lower bound R.

Method comparison. Both presented methods generate a feasible lightpath routing with regenerators for the transparent multi-hop scenario. In comparison, the first method is obviously much faster since Algorithm 3.2 is only of order $\mathcal{O}(h_p)$ for each variable $f_p^q > 0$, whereas the second method uses Algorithm 3.3 of order $\mathcal{O}(h_{\overline{p}}^3)$ for each flow variable $f_{\overline{p}}^q > 0$, and Algorithm 3.4 of order $\mathcal{O}((\max_{t \in T(\overline{p})} |L_t|)^{h_{\overline{p}}})$ for each generated candidate lightpath $\overline{p}$. However, the higher computational effort for the two-step heuristic typically pays off by solving the unified dimensioning and routing subproblem in advance on a sparser network.

The algorithms presented in this section enable to accomplish solutions for the original dimensioning and routing subproblem after various applications of Discnet as core optimizer. In principle, any of these postprocessing procedures can also be incorporated into Discnet's method for being executed in an integrated way. However, we suggest to restrict such extensions on very fast algorithms in order to avoid slowing down progress in solving the main task too much.
In summary, the composition of a complete solution procedure allows for several options and variations of individual steps. We therefore conclude the discussion with a brief overview of alternative methodologies for all scenarios.

3.3.4 Methodology overview

The general methodology outline for solving the (unified) dimensioning and routing subproblem in optical network design consists of three steps:

1. a preprocessing transforming the hardware model accordingly,
2. the application of Discnet as main optimization routine, and
3. a postprocessing to adapt solutions for the original problem.

With respect to different scenarios and performance aspects, alternative layouts for these steps have been discussed and allow for various combinations which results in a toolbox of individual solution methods. Table 3.1 gives an overview of the main options and indicates compatible compositions of the complete procedure for each network architecture.
Throughout all scenarios, two hardware representations provide a suitable input formulation: the presented models reinterpreted in the more abstract concept of components and resources (Comp. & Res.), or the transformation to a model based on precomputed discrete capacities (Discr. Cap.). The latter approach renders a solution space reduction which can improve the optimization performance, compensating the additional transformation effort.
As core solver, Discnet is applied for solving the problem relaxation with fractional routings, optionally extendible by regular execution of heuristics to generate integer routings on the run. This is in particular preferable for transparent single-hop networks with additional connection length restrictions, where valuable information

scenario	alternative selections				
	hardware model		routing		solution adaptions
opaque	Comp. & Res. Discr. Cap.	$\times$	fractional integer	$\geq$	integer (re-)routing none
transparent single-hop	Comp. & Res. Discr. Cap.	$>$	integer		
transparent multi-hop	Comp. & Res. Discr. Cap.	$\times$	fractional integer	$\geq$	int. (re-)rout. + reg. regenerations
	preprocessing	$\rightarrow$	DISCNET	$\rightarrow$	postprocessing

Table 3.1: *Solution method overview for the dimensioning and routing subproblem.*

provided by the enhanced pricing is best exploited immediately. For the other scenarios, the alternative to solve the relaxed problem with fractional routings at first and to derive and improve integer routings by postprocessing reroutings has been discussed as well. Moreover, transparent multi-hop networks require an additional completion step, for which two algorithms have been presented to place regenerators in order to turn the routing into a feasible lightpath routing.
With any method generated as a way along the lines in Table 3.1, a feasible solution of the dimensioning and routing subproblem for the considered scenario is obtained. In case of transparent networks, Theorem 3.1 guarantees further for the existence of a feasible wavelength assignment with converters, and the construction of such assignments is discussed in the following chapter.

Chapter 4

Wavelength Assignment with Converters

In mathematical optimization, coloring problems belong to the most fascinating and challenging problems. As prominent example, the coloring of maps states the historical origin of such problems, dating back to the 19th century, and gives also a hint on their deepness. Originally stated by Francis Guthrie in 1852, the famous four-color theorem waited for over a century for its proof, found by Appel and Haken [7, 8] in 1976. Having this one solved, the next intricacy is not far away. So, vertex coloring for arbitrary graphs, (seeming) as a slight generalization of map coloring, shows up to date to be a very hard problem which allows rarely for progress, neither theoretical nor algorithmic (see Jensen and Toft [79] for a monography on this and other graph coloring problems and results).
In optical networks, the introduction of WDM brings different wavelengths or, as an evident equivalent interpretation, colors into play. Being easy to handle in the opaque scenario, transparency introduces the wavelength continuity property for lightpaths and turns wavelength assignment with converters into a further challenge.

In this chapter, we investigate the wavelength assignment problem with converters in transparent optical networks. Theorem 3.1 ensures feasibility by unbounded wavelength converter employments, but our objective dictates economic utilization to complete network designs at low cost. To this end, we develop suitable methods for determining conflict-free assignments of the available wavelengths to a given set of lightpaths with as few converters as possible.

We start with a specification of the studied problem which is based on some preliminary simplifications in the decomposition. We introduce required notation and briefly discuss related work in the literature.
Next, we analyze the problem's theoretical complexity. We establish close relations to graph coloring problems and find the problem to remain hard even under strong assumptions. Only very simple network or routing structures allow for an efficient solving.
In Section 4.3, we revisit integer linear programming formulations for the problem. Since the adapted model part from Chapter 2 reveals poor properties, we make use

of the close relation to the vertex coloring problem for which Mehrotra and Trick [115] developed a successful approach. We derive a similar formulation for our problem and in fact find promising qualities.
The last section is devoted to the algorithms. We open with some heuristics from Zymolka et al. [178] which have been evaluated in Koster and Zymolka [93] and enhanced further in Koster and Zymolka [96]. For exact methods, we come back to the more promising integer linear program. Due to the large variable number, we describe a column generation procedure for solving the linear relaxation, yielding a good lower bound as shown by the results in Koster and Zymolka [94, 96]. Moreover, this method is extended to a full branch-and-price algorithm.

4.1 Problem specification

We cope with the wavelength assignment subproblem after having solved the dimensioning and routing subproblem according to our solution approach for optical network design, presented in Section 3.2. For opaque networks, assigning the available wavelengths link-wise from a proper hardware configuration to the traversing optical connections can be carried out arbitrarily without any wavelength converters. Therefore, we neglect the opaque scenario and focus on transparent networks in this chapter.
For both transparent scenarios, we derived in Section 3.2 an appropriate decomposition of the comprehensive models $\mathrm{OND}_{\mathrm{tsh}}$, $\mathrm{OND}_{\mathrm{tmh}}$ into the dimensioning and routing submodels $\mathrm{OND}^{DR}_{\mathrm{tsh}}$, $\mathrm{OND}^{DR}_{\mathrm{tmh}}$ and the wavelength assignment submodels $\mathrm{OND}^{WA}_{\mathrm{tsh}}$, $\mathrm{OND}^{WA}_{\mathrm{tmh}}$, respectively. Thereby, the formulations $\mathrm{OND}^{WA}_{\mathrm{tsh}}$ and $\mathrm{OND}^{WA}_{\mathrm{tmh}}$ vary only with respect to the sets of selectable lightpaths composing to the (end-to-end) connections in the routing models. Now, any given solution of $\mathrm{OND}^{DR}_{\mathrm{tsh}}$ respectively $\mathrm{OND}^{DR}_{\mathrm{tmh}}$ specifies all variables $\widetilde{f}^q_p$, which thus become parameters in $\mathrm{OND}^{WA}_{\mathrm{tsh}}$ and $\mathrm{OND}^{WA}_{\mathrm{tmh}}$. Theorems 3.1 and 3.3 guarantee feasibility of these wavelength assignment models for any parameter setting obtained from a feasible dimensioning and routing solution. By a parameter transformation, replacing the right hand sides in both (2.13b) and (2.16b) by the integer values $v_p = \sum_{q \in \mathcal{Q}} \widetilde{f}^q_p$ as number of times a lightpath $p \in \widetilde{\mathcal{P}}$ is used throughout the routing, we obtain the same formulation for both scenarios. Hence, the same problem is to be solved in either case. We postpone an integer linear program discussion to Section 4.3 and change over to a combinatorial perspective.

Combinatorial description. We consider a transparent optical network $\mathcal{N} = (N, L)$ with a feasible solution for the dimensioning and routing subproblem. Such a solution comprises the allocation of all hardware modules except for wavelength converters, and the routing of all connections by means of (consecutive) lightpaths. Each lightpath is routed on a link path $p \in \widetilde{\mathcal{P}}$ in the supply network $\mathcal{N}$, but no specific fiber on multi-fiber links is selected. Independent of commodities, the values v_p denote for each routing path $p \in \widetilde{\mathcal{P}}$ how many lightpaths are routed along this path. The multi-set P contains all lightpaths individually, i.e., holds v_p copies of each routing path $p \in \widetilde{\mathcal{P}}$. For differentiating the paths without multiplicity, the set $P_1 \subset P$ contains each used routing path just once, and the set $P_\ell \subset P_1$ comprises all

used routing paths that share link $\ell \in L$. All links on which no lightpath is routed can be removed from the network.

Observation 4.1 *W.l.o.g., we assume* $|P_\ell| > 0$ *for all links* $\ell \in L$.

The hardware configuration is dimensioned such that the switching and regeneration capacities at each node are large enough to handle all traversing connections. Likewise, the links provide sufficient wavelength channels to accommodate all traversing lightpaths, i.e.,

$$\sum_{p \in P_\ell} v_p \leq \sum_{\lambda \in \Lambda} y_\ell^\lambda \qquad \forall\, \ell \in L \tag{4.1}$$

holds, where y_ℓ^λ is the number of times wavelength $\lambda \in \Lambda$ is available by the (fibers and) WDM systems on link $\ell \in L$. Note that the values y_ℓ^λ come as parameters from the dimensioning solution. These quantities define the wavelength multiplicity function $K : \Lambda \times L \to \mathbb{Z}_+$ by $K(\lambda, \ell) := y_\ell^\lambda$ for all $\lambda \in \Lambda$, $\ell \in L$, and we write abbreviately $k_\ell^\lambda := K(\lambda, \ell)$ to avoid confusion in parameter and variable notation.
Conditions (4.1) ensure that all lightpaths containing only a single link can always be assigned a remaining wavelength for any feasible assignment to all other lightpaths (and never require converters). Such single-edge lightpaths can therefore be handled in a postprocessing step.

Observation 4.2 *W.l.o.g., we assume* $|L(p)| \geq 2$ *for all* $p \in P_1$.

The task now is to find an assignment of the available wavelengths on each link to all traversing lightpaths. Any exchange of the operated wavelength along a lightpath occupies a wavelength converter in the intermediate node, and the objective is to minimize the total cost for the installation of wavelength converters.

Assumptions. With regard to the flexible optical network design framework introduced in Chapter 2 and the corresponding models, the use of wavelength converters is subject to several side constraints, including the involvement of different converter types, preinstalled modules, and installation limitations. These aspects are practically less relevant and introduced mainly for completing the models in a uniform way. Basically, an accurate provision for such issues can be implemented into all developed algorithms, but would complicate the explanation. Therefore, we omit some details and transform the problem to a simpler setting as described in the following.
Regarding the objective, the most general case occurs when the cost for wavelength converters depends on the place of installation. In practice, however, this is rarely the case. The cost of a converter module is typically the same throughout the network. Moreover, the cheapest module type is commonly used (arbitrarily often) at all nodes. In such a situation, the total conversion cost depends linearly on the total number of new converters needed.
Moreover, as for all modules, the models have been formulated for preinstalled converters to be reused at their place. Typically, the converters are plugged directly onto the switch ports. Assuming that such individual allocation capabilities make converters also easy to reallocate, it is natural to consider them as movable modules. In this case, minimizing the total conversion cost in fact reduces to minimizing the

total number of converters employed. In addition, we need not take the actual place of preinstalled modules into account, but can restrict on reminding their number for the final total cost determination. As a consequence, these modules can be left out completely for the wavelength assignment.
In summary, we consider the wavelength assignment problem based on the following assumptions (unless stated otherwise):

- there is only one type of wavelength converters
- the installation cost for converters is independent of the particular node
- the total number of converters at each node is unbounded
- preinstalled converters are movable

In the sequel, we therefore investigate the resulting problem variant to which we refer as the Minimum Converter Wavelength Assignment Problem (MCWAP).

Problem 4.3 Minimum Conversion Wavelength Assignment Problem

INSTANCE:
- *the supply network* $\mathcal{N} = (N, L)$ *of the optical network*
- *the multi-set* P *of all lightpaths*
- *the set* Λ *of all available wavelengths*
- *the wavelength multiplicity function* $K : \Lambda \times L \to \mathbb{Z}_+$

SOLUTION:
- *a wavelength assignment* $\alpha_p : L(p) \to \Lambda$ *for all lightpaths* $p \in P$ *not exceeding the wavelength availabilities* k_ℓ^λ *on any link* $\ell \in L$ *for any wavelength* $\lambda \in \Lambda$
- *the number* $k_n^c \in \mathbb{Z}_+$ *of required wavelength converters at each node* $n \in N$

OBJECTIVE:
- $\min z := \sum_{n \in N} k_n^c$

For short notation, we denote an instance of MCWAP by $(\mathcal{N}, P, \Lambda, K)$.

Related work. With the (proclaimed) entering of transparency into WDM optical networks, wavelength assignment becomes an inevitable part of the design task and received increasing attention in the literature over the last decade. However, most of this work does not take conversion capabilities into account and deals with the combination of routing and wavelength assignment (RWA), as discussed in Section 3.1. Often, such approaches are also referred to as *wavelength routing* to indicate that no wavelength exchange is allowed on the routing paths. Feasibility is typically ensured by the assumption that (link) capacities are variable and can be determined afterwards. A typical setting uses uniformly equipped networks where any link holds k fibers each providing (the same) w wavelengths, and k, w, or a weighted sum of both is to be minimized. In some works, RWA is split into the routing and the wavelength assignment subtasks with major focus on the wavelength assignment and integrated routing path selection, see for instance Pal and Patel [135], Li and Simha [103], or Noronha and Ribeiro [132], all investigating partition coloring for unions of alternative routing paths for each connection. For the special case of RWA in ring networks with two routing alternatives only, Lee et al. [101] solve the joint problem with an

integer linear program applying column generation and branch-and-price. Further results for this and other topologies are subsumed in Beauquier et al. [12].
Wavelength assignment for a fixed routing has been studied less frequently in the literature, if not in connection with special network structures where the routing is fixed by the topology, like chains (see Winkler and Zhang [169]) or trees (see, for instance, Auletta et al. [11] or Erlebach and Jansen [44] and the surveys in Caragiannis et al. [29] or Kaklamanis [82]). Again, converters are rarely involved in such investigations. Instead, different objectives are focused, the most prominent one of minimizing the number of wavelengths required to accommodate all lightpaths conflict-free (see, among many others, Nomikos [129], Caragiannis et al. [27] or Caragiannis and Kaklamanis [28]). From a practical perspective, such approaches disregard the fact that the spectra provided by commercially available WDM systems are fixed. In particular, it is not clear how to cope with solutions demanding for more wavelengths than offered by the systems. For this, alternative approaches for fixed spectra deal with maximizing the number of lightpaths that can be established without conversion, see for instance Nomikos [129], Hu and Shuai [69], or Andrews and Zhang [6].
In case of fixed spectra, another problem variant without conversion focuses on optimizing the number (or total cost) of fibers required to realize a set of lightpaths (see for instance Nomikos et al. [131] or Andrews and Zhang [5, 6]). A common approach consists in deriving upper bounds or approximations for the required fiber multiplicity in relation to the *network load* l (see Margara and Simon [109, 110] or Li and Simha [102, 104]), in our notation defined as $l := \max_{\ell \in L} |P_\ell|$. These investigations stay abreast of the fact that optical network links in practice are usually furnished with multiple fibers. In some cases, the fiber number (and spectrum size) is assumed to be uniform throughout the network (as in Margara and Simon [109] or Ferreira et al. [48]). Our experiences show that the traffic distribution in practice is typically not homogeneous and contains often highly occupied centers. In Germany, for instance, only a few nodes are linked to international connections along which high amounts of data leave or enter the network. Other areas or subnetworks carry low traffic. Hence, designing homogeneously dimensioned networks is not always suitable.
Architectural comparisons reveal that the use of converters can increase the network performance significantly, allowing for better resource utilization (see for instance Jaumard et al. [78] or Ramamurthy et al. [144]). Nevertheless, only little research addresses problems involving conversion. In addition, many different assumptions regarding technical and operational properties of wavelength converter devices are taken. The most often considered problem employs both opaque and transparent nodes in a network, referred to as *network with sparse conversion*. In this case, the designer picks some nodes which obtain full conversion capability for all traversing lightpaths, while all other (transparent) nodes do not allow for wavelength exchanges at all (see, e.g., Kleinberg and Kumar [88], Belotti and Stidsen [14], Chu et al. [31], Sengezer and Karasan [151], Karasan et al. [83], or Erlebach and Stefanakos [45]). Other papers deal with *limited conversion* restricting the spectral conversion range (see, for instance, Belotti [13] or Ramaswami and Sasaki [146]), consider converters capable to handle multiple channels under additional restrictions (cf. Auletta et al. [10]), or hybrid architectures (as an example, see Cavendish et al. [30] with hybrid OXCs uniformly equipped with a number of o-e-o converters).

In the following sections, we study Mcwap in our particular setting and develop appropriate solution methods for completing the design of transparent optical networks. As a first step, we investigate the theoretical complexity of Mcwap and establish relations to known coloring problems.

4.2 Complexity

For analyzing the complexity of Mcwap, it is helpful to restate the problem in its decision variant, which asks for the existence of solutions with at most a given number $M \in \mathbb{Z}_+$ of converters.

Problem 4.4 Mcwap-Decision(M) (Mcwap-D(M))
INSTANCE:
- Mcwap *instance* $(\mathcal{N}, P, \Lambda, K)$
- *an integer* $M \in \mathbb{Z}_+$

QUESTION: *Does a* Mcwap *solution with at most* M *wavelength converters exist?*

We first investigate the relation of Mcwap-D(M) to classical graph coloring problems. By appropriate simplifications, we identify special cases of Mcwap-D(M) for which a complexity classification can be given.

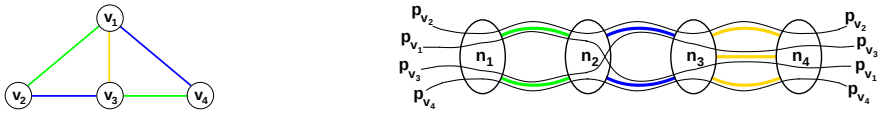
Mcwap special cases. In the general form, Mcwap differs from vertex coloring or edge coloring in several aspects. A major difference concerns the objective. Using a predefined color multi-set on each link, Mcwap asks to assign the available colors to the given paths with a total minimum number of color *changes* along the paths. Clearly, zero states a natural lower bound on the objective value, reflecting the case in which no color changes or, respectively, wavelength converters are required at all. If such a solution exists, all lightpaths can be operated on a single wavelength from origin to destination, and the construction of an associated wavelength assignment reduces to allocate wavelengths to lightpaths—a task much more similar to vertex or edge coloring where colors are allocated to vertices or edges, respectively. In our notation, Mcwap-D(0) corresponds to this special case, which occurs as subproblem in the design of optical networks without wavelength converters and has been investigated by several authors.
A further difference to classical graph coloring regards the color availability. In presence of multiple types of WDM systems offering different wavelength spectra, the wavelength multiplicities can vary on a link and among the links. The lightpaths have to choose link-wise from different sets of available colors which turns the task into a list coloring problem. For the sake of simplicity, it is therefore often assumed in the literature that all WDM systems offer the same set of wavelengths, i.e., all fibers are equipped with uniform WDM systems. We refer to this as *uniform fiber spectra* case where for each link $\ell \in L$, we have $k_\ell^{\lambda_1} = k_\ell^{\lambda_2}$ for all $\lambda_1, \lambda_2 \in \Lambda$. Then each lightpath can be potentially operated on each wavelength.
The assumption of uniform fiber spectra unifies the multiplicities of available wavelengths on each link, but not among the links since any link can be equipped with an individual number of fibers. These numbers determine how many lightpaths can share the same wavelength on a link. The general case of multi-fiber links exhibits another difference regarding the conflict conditions. Classical coloring problems

state assignment conflicts between pairs of items (vertices or edges). In multi-fiber MCWAP, wavelength allocation conflicts form conditions for the whole set of lightpaths sharing a link. Only on links with a single fiber, these set-based conditions can be dissolved into a set of pair-wise lightpath conflicts as equivalent formulation. This indicates another important special case of MCWAP in which all links offer (at most) a single fiber, referred to as the *single-fiber* case. Due to the possible reformulation with pairwise lightpath conflicts, this problem variant again becomes more similar to familiar coloring problems.
We remark that dedicating each lightpath initially to a specific fiber on each link would automatically yield a single-fiber MCWAP in which multi-fiber links are replaced by multiple parallel links, one for each fiber. Such a modification restricts the MCWAP solution space, however, significantly since each pair of lightpaths dedicated to the same fiber on a multi-fiber link could in fact share the same wavelength. Hence, all solutions with the latter property are excluded from the consideration. Moreover, such a preassignment would be arbitrary since dimensioning and routing does not make a discrimination. In order to avoid unnecessary solution space reductions, we consider MCWAP without preassignment of lightpaths to fibers and rather treat the single-fiber MCWAP as a special case of the general problem.

Equivalence to vertex coloring. Combining all described simplifications, we obtain the special case of *single-fiber* MCWAP-D*(0) with uniform fiber spectra*. In this variant, the wavelength assignment problem can be transformed into a vertex coloring problem of the so-called *path conflict graph* $G_P = (V_P, E_P)$. The path conflict graph contains a vertex v_p for each lightpath $p \in P$, and two vertices $v_{p_1}, v_{p_2} \in V_P$ are adjacent if the associated lightpaths $p_1, p_2 \in P$ share a common link, i.e., $v_{p_1}v_{p_2} \in E_P \Leftrightarrow L(p_1) \cap L(p_2) \neq \emptyset$. Any feasible vertex coloring of G_P with at most $|\Lambda|$ colors indicates a conflict-free wavelength assignment without converters, and vice versa. This way, single-fiber MCWAP-D(0) with uniform fiber spectra is reduced to the vertex coloring problem in graphs.
Due to this close relation of single-fiber MCWAP-D(0) with uniform fiber spectra and vertex coloring, equivalence of both problems is often suggested (or stated) in the literature. The following reverse reduction from vertex coloring to MCWAP shows an even stronger result.

Theorem 4.5 MCWAP-D*(M) is $\mathcal{NP}$-complete, even for the special case of single-fiber* MCWAP-D*(0) with uniform fiber spectra and load $l \leq 2$ on a supply network forming a chain with parallel links (multi-chain).*

Proof. Given a claimed solution for MCWAP-D(M), the confirmation of feasibility requires to verify a number of conditions which is polynomial in the input data. Hence, MCWAP-D(M) is in fact in $\mathcal{NP}$.
For a reduction of vertex coloring to MCWAP, we show that any graph G can be the path conflict graph of a MCWAP instance $(\mathcal{N}_G, P_G, \Lambda, K)$ such that G is C-colorable if and only if the instance with $|\Lambda| = C$ is solvable without converters. The actually constructed MCWAP instance will also have the claimed additional properties.
Consider an arbitrary vertex coloring instance asking whether a graph $G = (V, E)$ can be colored with at most $C \geq 3$ colors (for $C = 2$, the problem is equivalent to verifying bipartiteness of G and thus polynomial). W.l.o.g., we can assume G to be simple. Karp [84] first proved this problem to be $\mathcal{NP}$-complete.



(a) *Vertex coloring graph G with an edge coloring.*

(b) *Resulting* Mcwap *network $\mathcal{N}_G$ with paths p_v.*

Figure 4.1: *Example of the construction of a* Mcwap *instance from a vertex coloring instance by help of edge coloring.*

With G to be a path conflict graph, any edge represents a conflict between the paths corresponding to the incident vertices, and these paths have to share a link in the supply network to construct. For this construction, we setup an individual supply link for each conflict and route the paths accordingly. In order to keep the network small, we exploit an arbitrary edge coloring of G with $E_1, \dots, E_k \subset E$ denoting the edge color classes. Note that $k \leq |E|$. The minimum number of colors needed for such an edge coloring is the chromatic index $\chi'(G)$, but for our purpose, any edge coloring will do (e.g., for simple graphs, an edge coloring with $\Delta(G) + 1$ colors can be computed in polynomial time, see Vizing [163]).
Based on the k-edge coloring of G, we construct a supply network $\mathcal{N}_G = (N, L)$ with $k+1$ nodes $N = \{ n_1, \dots, n_{k+1} \}$ forming a multi-chain where each consecutive pair is connected by parallel links. The instance construction is exemplary depicted in Figure 4.1. For $i = 1, \dots, k$, we introduce $|V| - |E_i|$ parallel links between n_i and n_{i+1}. Each link is equipped with a single fiber carrying a WDM system that offers all wavelengths in a spectrum Λ with $|\Lambda| = C$, i.e., $K(\lambda, \ell) = 1$ for all $\lambda \in \Lambda$ and $\ell \in L$.
In the network $\mathcal{N}_G$, we define the following path set. For each vertex $v \in V$, we setup a path p_v from node n_1 to node n_{k+1} traversing all intermediate nodes. The particular link at each hop to use by a path is derived from the edge coloring. If v is incident to an edge $e = vw \in E_i$, then p_v shares one of the links between n_i and n_{i+1} with path p_w associated to vertex w. Otherwise p_v uses one of the parallel links exclusively. At the i-th hop, all $|V|$ paths require $|E_i|$ pair-wise shared plus $|V| - 2\,|E_i|$ exclusively used links, such that in total $|E_i| + |V| - 2\,|E_i| = |V| - |E_i|$ links suffice between n_i and n_{i+1}. In addition, any supply link in $\mathcal{N}_G$ is traversed by at most two paths, hence its channel capacity $|\Lambda| = C > 2$ is sufficient to accommodate the complete path set $P_G = \{ p_v \mid v \in V \}$.
By construction, adjacent vertices in G correspond to paths in $\mathcal{N}_G$ that share a link, and vice versa. Hence, G is the path conflict graph associated to the Mcwap instance $(\mathcal{N}_G, P_G, \Lambda, K)$. Consequently, G is C-colorable if and only if $(\mathcal{N}_G, P_G, \Lambda, K)$ has a converter-free solution for $|\Lambda| = C$. Moreover, the transformation is polynomial in the size of the vertex coloring instance, since $\mathcal{N}_G$ has $\mathcal{O}(|E|)$ nodes and $\mathcal{O}(|V| \cdot |E|)$ links, $|\Lambda| = C$, and $|V|$ paths each of hop-length $k \leq |E|$ are used.
As a result, vertex coloring has been reduced to single-fiber Mcwap-D(0) with uniform fiber spectra on a multi-chain network with load $l \leq 2$, proving the claim. ■

In combination with the preceding argument of reducing Mcwap to vertex coloring, Theorem 4.5 yields in fact the equivalence of vertex coloring and single-fiber Mcwap-D(0) with uniform fiber spectra on multi-chain supply networks with load

at most 2. As a further consequence, this also implies that any instance of single-fiber Mcwap-D(0) with uniform fiber spectra on an arbitrary network can be transformed into an equivalent instance on a multi-chain supply network.
Moreover, results similar to Theorem 4.5 hold also for other variations of the Mcwap special case properties. For instance, giving up the chain structure of the physical topology, a restriction to simple networks can be easily achieved by placing an additional node on each but one parallel supply link at each hop in the constructed network $\mathcal{N}_G$. This in fact generates a simple series-parallel network. With this modification, $\mathcal{NP}$-completeness carries over to the problem class of single-fiber Mcwap-D(0) with uniform fiber spectra on simple series-parallel supply networks (with load $l \leq 2$).

Polynomially solvable cases. A further restriction to single-fiber Mcwap-D(0) with uniform fiber spectra on a *simple* chain network finally makes the problem polynomially solvable. In this case, the associated path conflict graph forms an interval graph. Interval graphs are perfect (see for instance Schrijver [150]), and the chromatic and clique numbers coincide. Hence, the number of required wavelengths (chromatic number) equals the maximum load (clique number) in the network, and the paths can always be assigned wavelengths without need of conversions in polynomial time. This result indicates that the $\mathcal{NP}$-completeness statement in Theorem 4.5 cannot be strengthen further.
Expanding Mcwap-D(0) with uniform fiber spectra to the multi-fiber case, the corresponding path conflict graph coloring transforms to a *generalized interval graph coloring*. Winkler and Zhang [169] proved that this problem is polynomially solvable, too.

Theorem 4.6 (Winkler and Zhang [169]) *Multi-fiber* Mcwap *with uniform fiber spectra on a simple chain network has always a converter-free solution.*

By a constructive proof, they also provide a method to compute optimal assignments for this problem variant.
In connection with Theorem 4.5 and Theorem 4.6, the following remark has to be made. We mentioned that preassigning lightpaths to particular fibers turns a multi-fiber problem on a simple chain into a single-fiber problem on a multi-chain network, with the intention to ease the task. Indeed, the theorems show that exactly the converse happens on chains: A preassignment makes an easy problem hard, and even more, an optimal (converter-free) solution could be lost due to the solution space shrinking. Hence, the purposed 'simplification' shows up as bad idea.

Mcwap **with converters.** In the proof of Theorem 4.5, we mentioned that vertex coloring for $C = 2$ is polynomial, and thus the derived problem equivalence yields that single-fiber Mcwap-D(0) with uniform fiber spectra on multi-chains and $|\Lambda| = 2$ is polynomially solvable, too. But even such a strongly restricted variant of Mcwap turns hard if converters are involved. Note that individual lightpaths can make use of more than one converter, hence there is no one-to-one correspondence between converters and converted lightpaths. However, the minimum number of converted lightpaths over all solutions gives a lower bound on the minimum number of converters needed. Its determination is already a difficult matter.

Theorem 4.7 *For general $M > 0$, it is $\mathcal{NP}$-complete to decide whether a single-fiber* Mcwap *instance with uniform fiber spectra on a multi-chain network has a solution with at most M converted lightpaths, even if only $|\Lambda| = 2$ wavelengths are involved.*

Proof. Consider an arbitrary Mcwap instance with all claimed properties. Obviously, the question whether a solution with at most M converted lightpaths exists or not is equivalent to the question whether deletion of at most M lightpaths leaves an instance with a converter-free solution or not, i.e., whether Mcwap-D(0) returns a positive answer for the reduced instance.
We have already shown that single-fiber Mcwap-D(0) with uniform fiber spectra of size $|\Lambda| = C$ is equivalent to C-colorability of the corresponding path conflict graph. Now, removing a lightpath from such a Mcwap instance corresponds to removing the associate vertex in the equivalent vertex coloring instance in G_P and vice versa. Hence, the equivalence relation holds also for instances reduced by removing lightpaths respectively the associated vertices. As a consequence, the claimed decision problem is equivalent to the question whether the path conflict graph has a C-colorable subgraph induced by a vertex subset of cardinality at least $|V| - M$. In the considered special case $|\Lambda| = C = 2$, the latter problem is further equivalent to the question whether G_P has a bipartite subgraph obtained by removing at most M vertices. Yannakakis [173] (see also Garey and Johnson [52] where this problem belongs to the problems on 'induced subgraph with property Π') proved this last problem to be $\mathcal{NP}$-complete, and by the derived problem equivalence chain, the statement carries over to the claimed problem, too. ■

Theorem 4.7 shows that in case converters are in fact required, determining the minimum number of converted lightpaths and this way getting an estimation for the number of unavoidable converters for Mcwap cannot be performed efficiently even under very strong assumptions. Hence, larger classes of polynomially solvable problems are more likely to be detected for Mcwap-D(0) special cases.

Conflict hypergraphs. Without preassigning lightpaths to fibers, multi-fiber Mcwap-D(0) cannot be transformed to usual vertex coloring, since path conflicts cannot anymore be equivalently modeled by edges. In a natural extension of the path conflict graph, the conflicts at a link form a hyperedge, containing all paths traversing the link. Ferreira et al. [48] investigated such an extension to coloring of hypergraphs in context of wavelength assignment. Besides uniform wavelength spectra with w wavelengths per fiber, they also assume all links to be uniformly equipped with the same number k of fibers, resulting in so-called (k, w)-*networks*. Using a construction similar to $\mathcal{N}_G$ in Theorem 4.5, the wavelength assignment problem without converters in these networks is proven to be equivalent to a vertex coloring generalization on hypergraphs, where the number of equally colorable nodes on each hyperedge is limited by k. Transferred to our setting, they obtain:

Theorem 4.8 (Ferreira et al. [48]) Mcwap-D*(0) in a (k, w)-network is $\mathcal{NP}$-complete.*

Hence, the assumption of a regular dimensioning does not reduce the complexity of Mcwap in general networks.

Ring networks. By Theorem 4.6, multi-fiber Mcwap with uniform fiber spectra on chain networks is polynomially solvable. This encourages to seek for further specific network structures whose exploitation allows for efficient solution methods.
A natural generalization of chains consists in ring networks, which often occur in practice as building blocks of network topologies. In the single-fiber case with uniform fiber spectra, it is easy to see that Mcwap-D(0) on rings is equivalent to the coloring of circular-arc graphs, which has been shown to be $\mathcal{NP}$-complete by Garey et al. [53]. For the multi-fiber extension, Li and Simha [102] studied k-fiber rings where each link contains k fibers. By an appropriate extension of the problem reduction from circular-arc coloring, they proved:

Theorem 4.9 (Li and Simha [102]) Mcwap-D*(0) in (k, w)-ring networks is $\mathcal{NP}$-complete.*

As straightforward generalization, the result carries over to rings with individual fiber numbers per link. Hence, rings in any variant do not offer structural opportunities exploitable for Mcwap. Even worse, any meshed network topology with cycles contains a ring subproblem and thus, unless further strong simplifications are applied, inherits the complexity from Theorem 4.9. This observation motivates to focus on cycle-free topologies, i.e., tree-like structures, in order to identify further polynomially solvable cases.

Star networks. Among tree topologies, star networks are the most simple structure. While single-fiber Mcwap-D(0) on stars was already shown to be $\mathcal{NP}$-complete by Raghavan and Upfal [143], the multi-fiber extension for the regular case of k fibers per link was again studied by Li and Simha [102]. Surprisingly, the difficulty vanishes for specific fiber multiplicities:

Theorem 4.10 (Li and Simha [102]) Mcwap-D*(0) with uniform fiber spectra on star networks is $\mathcal{NP}$-complete in general. In (k, w)-star networks with k even,* Mcwap-D*(0) can be answered in polynomial time.*

In fact, they show that no conversion is necessary in even-dimensioned (k, w)-stars for any path set not exceeding the capacities.

Special routings. Li and Simha [102] observed further that Theorem 4.10 can also be applied to Mcwap instances on arbitrary networks with the specific connection pattern that all paths span at most two links. Harder [66] investigated such instances, originally for the single-fiber case, and proved:

Theorem 4.11 (Harder [66]) Mcwap-D*(0) with uniform fiber spectra and paths of hop-length at most two in arbitrary networks is $\mathcal{NP}$-complete.*

Li and Simha [102] transformed such an instance to an instance on a star network as follows. Each original link $\ell \in L$ is replaced by a node n_ℓ, which is connected to an additional central node n_c. Each original path is transformed to a path on those links in the star that are incident to the vertices representing its original links. An example for this transformation is depicted in the left part of Figure 4.2. It is easily verified that any wavelength assignment for the constructed star network instance can be retransformed in a straightforward manner to the original instance,

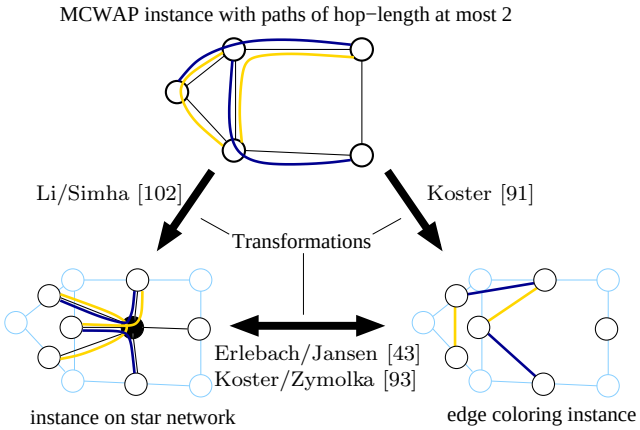


Figure 4.2: *Transforming* MCWAP *instances with paths of hop-length at most 2.*

preserving the assigned wavelengths path-wise. As a consequence, Theorem 4.10 directly implies:

Corollary 4.12 (Li and Simha [102]) *For* MCWAP *with paths of hop-length at most two in an arbitrary network with k fibers per link, k even, there exists always a solution without converters.*

For MCWAP-D(0) with uniform fiber spectra on single-fiber star networks, Erlebach and Jansen [43] proved the equivalence to edge coloring of (multi-) graphs, first shown to be $\mathcal{NP}$-complete by Holyer [68]. While paths with a single link can be neglected, each path spanning two links is replaced by an edge connecting the path's origin and destination. The star center is removed together with all incident links. This reversible transformation is illustrated in the lower part of Figure 4.2. Now, any edge coloring of the resulting (multi-) graph with at most $|\Lambda|$ colors corresponds to a converter-free wavelength assignment of the single-fiber MCWAP instance on the star network, and vice versa.
Extending this equivalence transformation to capture the multi-fiber case for star networks was adumbrated in Koster and Zymolka [93]. In contrast to vertex coloring, edge coloring can be generalized for multiple fibers without transition to hypergraphs as follows. Let $k_\ell \in \mathbb{N}$ denote the number of fibers on link $\ell \in L$. Then, k_ℓ paths on link ℓ can share the same wavelength. Equivalently, k_ℓ edges incident to vertex n_ℓ can be assigned the same color in the multi-graph. The latter problem is known as *k-edge-coloring*, see Hakimi and Kariv [64] (who originally used f instead of k). This way, the result of Erlebach and Jansen [43] is lifted to the equivalence of multi-fiber MCWAP on a star network and k-edge-coloring of a multi-graph. Note that the fiber multiplicities need not be uniform in the network, and we denote such networks as *(k_ℓ, w)-star networks* for the case of uniform fiber spectra.
Alternatively, Koster [91] proposes a direct way to transform a multi-fiber MCWAP instance with paths of length at most two into a k-edge-coloring problem on a multi-graph, as indicated in the right part of Figure 4.2. This way, Koster [91] generalizes Corollary 4.12 to:

Theorem 4.13 (Koster [91]) *Any* MCWAP *instance with paths of hop-length at most two in an arbitrary (k_ℓ, w)-network with individual, but even fiber numbers $k_\ell \in 2\mathbb{N}$ for all $\ell \in L$ has a solution without converters.*

polynomial cases	$\mathcal{NP}$-complete cases
multi-fiber MCWAP-D(0) on simple chain networks	single-fiber MCWAP-D(0) on multi-chain networks
single-fiber MCWAP-D(0) on multi-chain networks with $\|\Lambda\| = 2$	becomes single-fiber MCWAP-D(0) on a multi-chain network with $\|\Lambda\| = 2$ true after deletion of $\leq M$ lightpaths?
single-fiber MCWAP-D(0) on trees of bounded degree	single-fiber MCWAP-D(0) on arbitrary trees
MCWAP-D(0) on (k_ℓ, w)-star networks, k_ℓ even $\forall\ \ell \in L$	MCWAP-D(0) on general (k_ℓ, w)-star networks
MCWAP-D(0) on (k_ℓ, w)-networks, k_ℓ even $\forall\ \ell \in L$, with $\|L(p)\| \leq 2\ \forall\ p \in P$	MCWAP-D(0) on general (k_ℓ, w)-networks with $\|L(p)\| \leq 2\ \forall\ p \in P$

Table 4.1: *Complexity results for* MCWAP *with uniform fiber spectra.*

Theorem 4.13 can be in particular applied to star networks, where simple paths naturally have length at most two, and thus Theorem 4.13 extends the second statement in Theorem 4.10 to the irregularly dimensioned case of (k_ℓ, w)-networks with all k_ℓ even as well.

Tree networks. With stars as special case, MCWAP-D(0) on arbitrary tree networks clearly remains $\mathcal{NP}$-complete in general according to Theorem 4.10. However, it is an interesting question whether the polynomially solvable special cases for star networks can be extended to tree networks. For the single-fiber problem, Erlebach and Jansen [44] argue that a path coloring with minimum number of wavelengths can be obtained in polynomial time by merging optimal path colorings on local star-like instances. In contrast to arbitrary single-fiber stars discussed above, trees (and stars) of bounded degree allow to solve the local problem polynomially, and thus the corresponding MCWAP-D(0) instance, too. Unfortunately, such a merging of locally optimal solutions on excised star networks cannot be straightforward carried over to the multi-fiber case.

Overview. As conclusion, Table 4.1 summarizes the complexity results for the discussed MCWAP special cases such that closely related problem variants are arranged together. One observes that polynomial solvability requires very strong assumptions on the dimensioned supply network structure or the path routing. Releasing any of the properties listed in the left part turns MCWAP into a hard problem.
It is likely that further complexity strengthenings, such as non-approximability results of Hochbaum [67] for graph coloring, can be carried over to (particular variants of) MCWAP. For instance, Nomikos [129] proved non-approximability for path coloring on general single-fiber topologies for both minimizing the number of colors as well as, for a fixed spectrum, maximizing the number of paths which can be assigned wavelengths without conflicts (and converters). More recently, in relation to Theorem 4.8, Andrews and Zhang [5] proved that there is no constant-factor approximation for MCWAP-D(0) in (k, w)-networks with fixed w unless $\mathcal{NP} \subset \mathcal{ZPP}$. Such questions are important for the development of approximation algorithms with guaranteed solution quality. However, this approach goes beyond the scope of this

thesis. Instead, we focus on efficient heuristic algorithms for MCWAP as well as on exact methods, beginning with a modeling in terms of integer programming.

4.3 Integer linear program formulations

In this section, we derive and compare two integer linear programming formulations for MCWAP. The first formulation—based on the corresponding model parts from Chapter 2—can be seen as a straightforward approach, which directly encodes the assignment decisions and leads to a compact *assignment formulation*. The second formulation is inspired by a successful column generation approach for the vertex coloring problem by Mehrotra and Trick [115]. Our adaption to MCWAP integrates structural information on the possible path combinations for each wavelength into model columns and yields the large-scale *path packing formulation*.

4.3.1 Assignment formulation

A natural way to formulate coloring-like problems is by introduction of decision variables that directly represent the color assignments. In case of MCWAP, we have to assign wavelengths to every link of a lightpath. Similar to Section 2.2.3, multiple lightpaths routed along the same path are considered cumulative, and we use the following variables to express the wavelength assignments:

$a^{\lambda}_{p\ell} \in \mathbb{Z}_+$ denotes the number of times wavelength $\lambda \in \Lambda$ is assigned to lightpaths along path $p \in P_1$ at link $\ell \in L(p)$.

These variables are integer because multiple lightpaths can be routed along each path $p \in P_1$ and wavelengths can be available multiple times on a link by parallel fibers. The alternative to model the wavelength assignment for each individual lightpath by binary variables is discussed below.
To calculate the number of needed converters, a second set of variables is necessary:

$a^{\lambda}_{pn} \in \mathbb{Z}_+$ denotes the number of lightpaths routed along path $p \in P_1$ that are converted from wavelength $\lambda \in \Lambda$ to any other wavelength at node $n \in N(p)$.

Note that the variables are indexed only with the incoming wavelengths at node n, since it does not matter for the converter counting to which wavelength a lightpath is converted. Instead, we just account for lightpaths that cannot continue with the same wavelength on the next link after n and thus need a converter in n.
Using these variables, the straightforward assignment formulation of MCWAP reads:

$$z_A = \min \sum_{\lambda\in\Lambda} \sum_{p\in P_1} \sum_{n\in N(p)} a^{\lambda}_{pn} \tag{4.2a}$$

$$\text{s.t.} \quad \sum_{\lambda\in\Lambda} a^{\lambda}_{p\ell} = v_p \qquad \forall\, p \in P_1,\ \ell \in L(p) \tag{4.2b}$$

$$\sum_{p \in P_\ell} a^\lambda_{p\ell} \leq k^\lambda_\ell \qquad \forall\, \lambda \in \Lambda,\ \ell \in L \tag{4.2c}$$

$$a^\lambda_{p\ell_i} - a^\lambda_{p\ell_{i+1}} \leq a^\lambda_{pn_i} \qquad \forall\, \lambda \in \Lambda,\ p \in P_1,\ 1 \leq i \leq h-1, \text{ where } p = (n_0, \ell_1, n_1, \ell_2, n_2, \ldots, \ell_h, n_h) \tag{4.2d}$$

$$a^\lambda_{p\ell}, a^\lambda_{pn} \in \mathbb{Z}_+ \qquad \forall\, \lambda \in \Lambda,\ p \in P_1,\ \ell \in L,\ n \in N \tag{4.2e}$$

A detailed explanation of this model has been already given in Section 2.2. As only difference, the multi-set of lightpaths is not determined anymore by the routing formulation, but given as input. Hence, the capacity conditions (4.2c) restrict on the given set P_ℓ of different routing paths traversing link ℓ, and constraints (4.2b) hold at the right hand side a parameter v_p that reflects the number of lightpaths routed along path $p \in P_1$, as introduced in the problem description in Section 4.1. Note that the size of the assignment formulation (4.2) is polynomial in the number and hop-length of the paths in P_1 and the number of wavelengths.

Individual lightpath assignments. By variables $a^\lambda_{p\ell}$, wavelengths are at first assigned cumulatively to all lightpaths routed along a path $p \in P_1$. It remains to resolve such a solution for parallel lightpaths to a wavelength assignment for each individual lightpath without need of additional converters.
Regarding constraints (4.2d), one observes that converters are only placed for those wavelengths at a link that are not assigned (sufficiently often) at the following link on the path. In turn, this dictates to assign any wavelength on as much consecutive links as possible to the same lightpath in order to avoid unaccounted conversions. Consequently, the cumulative wavelength assignment for parallel lightpaths decomposes into a multi-set of maximal subpaths each with a continuous wavelength. An appropriate concatenation of these subpaths finally provides the individual lightpath assignments. This procedure is easily realized for each path $p \in P_1$ in polynomial time of order $\mathcal{O}(v_p \cdot |L(p)|)$.
Since the resolving of an aggregated assignment to parallel lightpaths can be performed efficiently, it becomes clear that the alternative modeling by binary variables for individual lightpath assignments does not provide an advantage. In contrast, the formulation would grow in size, and the solutions would degenerate by allowing for permutation of the assignments for parallel lightpaths. Their cumulative assignment subsumes such equivalent solutions and thus overcomes this type of degeneracy.

Wavelength symmetry degeneracy. Anyway, a major disadvantage of the assignment formulation is the degeneracy of solutions caused by the symmetry of the spectrum Λ. Typically, many wavelengths are permutable due to their concerted provision by the WDM systems. The worst situation occurs for the uniform fiber spectra case. Then, not only for all integer solutions, but also for fractional solutions of linear relaxations, $\mathcal{O}(|\Lambda|!)$ equivalent solutions exist. Cutting plane approaches and branch-and-bound based on similar assignment formulations have been shown to be computationally intractable for problems with such characteristics (like vertex coloring, see Pardalos et al. [137], or frequency assignment, see Aardal et al. [1]). These experiences motivate to investigate an alternative formulation.

4.3.2 Path packing formulation

For the vertex coloring problem, Mehrotra and Trick [115] introduced a column generation approach to overcome the color symmetry degeneracy. Given a graph G, it is well known that all vertices that can be colored by the same color form a *stable set*. Thus, the chromatic number $\chi(G)$ of a graph G is given as the minimum number of stable sets needed to cover all its vertices. By introducing a selection variable for every stable set in the graph, an alternative formulation for vertex coloring is derived. Due to the huge number of variables, the columns corresponding to the stable sets are generated dynamically. Although the associated pricing problem is $\mathcal{NP}$-hard, the method proved computationally tractable and was able to solve many vertex coloring instances.
This successful approach has inspired to develop a similar formulation for Mcwap. A step-by-step generalization of the formulation from vertex coloring to Mcwap can be found in Koster and Zymolka [96]. To simplify the explanation, we at first assume uniform fiber spectra and discuss the adaption to the non-uniform case at the end.

Subpath packings. The key to our approach is the following observation. A wavelength assignment for any individual lightpath can be interpreted as a partitioning into subpaths assigned the same wavelength on all links. Throughout the lightpaths, all subpaths that are assigned the same wavelength can be viewed as a *packing of subpaths* with respect to the wavelength's multiplicities. This way, a wavelength assignment for all lightpaths decomposes into $|\Lambda|$ subpath packings. The number of converters needed for a lightpath partitioned into i subpaths is exactly $i-1$. Since the total number of lightpaths is fixed, the objective of Mcwap is therefore equivalent to minimizing the total number of subpaths involved.
In order to formulate the above problem reinterpretation as an integer linear program, we have to characterize feasible packings of subpaths. We introduce the following notation. For each $p \in P_1$, let S_p be the set of all subpaths s of p. Note that $|S_p| = \frac{1}{2}|L(p)|\,(\,|L(p)|+1)$, and the same subpath can be in multiple sets S_p. We define $S := \cup_{p\in P_1} S_p$ as the set of all possible subpaths. Then a *path packing* ϕ is a multi-set of items from S such that all subpaths $s \in \phi$ can be assigned the same wavelength λ, i.e., for every link $\ell \in L$, at most k_ℓ^λ subpaths containing link ℓ are in the multi-set ϕ. The multiplicity of each subpath $s \in S$ in the path packing ϕ is denoted by $m_\phi^s \in \mathbb{Z}_+$. The set Φ subsumes all path packings ϕ.

Path packing formulation. Now, Mcwap consists in the task to find $|\Lambda|$ path packings such that all lightpaths are properly covered. Thereby, a path packing $\phi \in \Phi$ can be selected more than once, if the multi-set of subpaths to be assigned the same wavelength is identical for multiple wavelengths. For every path packing $\phi \in \Phi$, we therefore introduce a general integer variable:

$x_\phi \in \mathbb{Z}_+$ — denotes the number of times path packing $\phi \in \Phi$ is selected in the assignment (where each path packing states that all contained subpaths $s \in \phi$ are assigned the same wavelength).

In addition, we have to specify which subpaths are used for the partitioning of each lightpath. Again, we consider all lightpaths along the same routing path concertedly

and use the following variables for the correspondence:

$y_p^s \in \mathbb{Z}_+$ denotes the number of times subpath s is used in the partitionings of lightpaths routed along path $p \in P_1$.

Now, MCWAP reads alternatively:

$$z_P = \min \sum_{p \in P_1} \sum_{s \in S_p} y_p^s \qquad \left[- \sum_{p \in P_1} v_p \right] \tag{4.3a}$$

$$\text{s.t.} \sum_{s \in S_p : \ell \in L(s)} y_p^s = v_p \qquad \forall\, p \in P_1,\ \ell \in L(p) \tag{4.3b}$$

$$\sum_{\substack{p \in P_1: \\ s \in S_p}} y_p^s = \sum_{\phi \in \Phi} m_\phi^s x_\phi \qquad \forall\, s \in S \tag{4.3c}$$

$$\sum_{\phi \in \Phi} x_\phi \leq |\Lambda| \tag{4.3d}$$

$$y_p^s, x_\phi \in \mathbb{Z}_+ \qquad \forall\, p \in P_1,\ s \in S_p,\ \phi \in \Phi \tag{4.3e}$$

As expressed by (4.3a), the total number of converters is given by the total number of subpaths selected for all lightpath partitionings minus the total lightpath number, the latter being a fixed value and therefore put in brackets. For any routing path $p \in P_1$ at every link $\ell \in L(p)$, v_p subpaths covering that link must be provided for the lightpath partitionings, which is enforced by constraints (4.3b). Constraints (4.3c) model that every subpath $s \in S$ is offered by the selected path packings as often as chosen for partitioning of lightpaths. The single constraint (4.3d) restricts the number of selectable path packings to the size of the available spectrum Λ, and constraints (4.3e) guarantee integrality of the solution.

Solutions. Regarding the solutions of the path packing formulation (4.3), two retransformations are required to obtain a solution for the original problem: deriving individual wavelength assignments for the lightpaths and the placement of converters. Both issues are easy to accomplish. At first, we assign the wavelengths in Λ to the selected paths packings, i.e., to each subpath in a packing. For the individual lightpath wavelength assignments, we proceed as for the assignment formulation, except for that the variables y_p^s already specify the subpaths to use for each bunch of parallel lightpaths. Hence, we have only to concatenate these subpaths accordingly to get the lightpath assignments.
The placement of converters follows also directly from the partitioning variables y_p^s. Whenever a subpath does not reach the associated lightpath's destination node, its wavelength cannot be used further, and a wavelength converter is needed. Hence, the number of required converters at a node $n \in N$ is given by

$$y_n^{\mathcal{R}} = \sum_{\substack{p \in P_1: \\ n \in N(p)}} \sum_{\substack{s \in S_p: \\ d_s = n}} y_p^s \,.$$

Non-uniform fiber spectra. For the case of non-uniform fiber spectra, formulation (4.3) is to be adapted as follows. The wavelength spectrum Λ is partitioned into subsets $\Lambda^1, \ldots, \Lambda^w$ such that all wavelengths in a subset have the same multiplicities on all supply links, i.e., for each $i = 1, \ldots, w$, any wavelengths $\lambda_1, \lambda_2 \in \Lambda^i$

have $k_\ell^{\lambda_1} = k_\ell^{\lambda_2}$ for all links $\ell \in L$. Each subset Λ^i specifies an individual set Φ^i of feasible path packings with respect to the corresponding wavelength multiplicities. A solution now composes of selecting $|\Lambda^i|$ path packings from set Φ^i for each $i = 1, \ldots, w$. With $\Phi := \bigcup_{i=1}^{w} \Phi^i$, all but one constraints in formulation (4.3) remain unchanged. Only the path packing selection constraint (4.3d) is replaced by an individual restriction for each wavelength subset as

$$\sum_{\phi \in \Phi^i} x_\phi \leq |\Lambda^i| \qquad \forall\, i = 1, \ldots, w\,.$$

Note that $1 \leq w \leq |\Lambda|$, where $w = 1$ reflects the uniform fiber spectra case in model (4.3), and $w = |\Lambda|$ is the opposite extreme that each wavelength has unique multiplicities. For practical instances, however, w is typically small since few different wavelength spectra are offered by the available WDM systems. Then, many wavelengths are provided concertedly on all fibers, which yields large subsets Λ^i.

In the path packing formulation (4.3), the number of constraints is comparable to that of the assignment formulation, but the number of variables is tremendously large. However, the advantage of the formulation is that wavelengths are not distinguished explicitly, but represented by the corresponding path packings to select. As a consequence, solutions are unaffected by wavelength permutations (within the subsets Λ^i).

4.3.3 Formulation comparison

The two presented formulations for MCWAP show very different characteristics. The assignment formulation is much more compact in size, but suffers from the spectral symmetry degeneracy, whereas the path packing formulation breaks the symmetry at the price of a huge number of variables. For computational purposes, a further important aspect of integer linear program formulations is the strength of the linear relaxation, reflecting how good the linear program approximates integer solutions. Some results are presented to evaluate and compare this quality for both models.

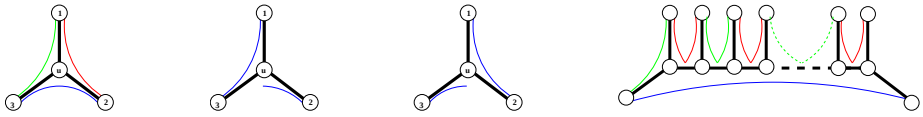
Assignment model relaxation. Let z_A^* denote the optimal value of the linear relaxation of the assignment formulation (4.2). In the uniform fiber spectra case where each link $\ell \in L$ offers all wavelengths with the same link-individual multiplicity $k_\ell \in \mathbb{Z}_+$, the following observation can be made:

Lemma 4.14 *If $k_\ell^\lambda = k_\ell$ for all $\lambda \in \Lambda$, $\ell \in L$, then $z_A^* = 0$.*

Proof. Consider the fractional solution with $a_{p\ell}^\lambda := v_p / |\Lambda|$ and $a_{pn}^\lambda := 0$ for all $\lambda \in \Lambda$, $p \in P_1$, $\ell \in L(p)$, $n \in N(p)$. This solution obviously satisfies all constraints (4.2b) and (4.2d). For constraints (4.2c), the dimensioning ensures by (4.1) that

$$\sum_{p \in P_\ell} a_{p\ell}^\lambda = \frac{1}{|\Lambda|} \sum_{p \in P_\ell} v_p \overset{(4.1)}{\leq} \frac{1}{|\Lambda|} \sum_{\lambda \in \Lambda} k_\ell^\lambda = k_\ell^\lambda$$

for all $\lambda \in \Lambda$, $\ell \in L$, since $\sum_{\lambda \in \Lambda} k_\ell^\lambda = k_\ell\, |\Lambda| = k_\ell^\lambda\, |\Lambda|$ by prerequisite.
Hence, the solution is feasible for the linear relaxation of (4.2) and has objective value 0, which is optimal due to non-negativity of all variables. ■



(a) *Instance.* (b) *Two path packings with* $x_\phi = 1$ *in the linear relaxation of* (4.3). (c) *Extended network with arbitrarily large difference between the LP values.*

Figure 4.3: *Star network with different linear relaxation values.*

In the general case with non-uniform wavelength multiplicities, z_A^* can clearly take positive values since paths for which consecutive fibers have no offered wavelengths in common always require converters in any feasible solution, including fractional ones.

LP value comparison. Let z_P^* denote the value of the linear relaxation of the path packing formulation (4.3). It is easy to verify that every (fractional) solution of (4.3) can be transformed to a (fractional) solution of (4.2) with the same objective value, which yields:

Lemma 4.15 *The objective value of the linear relaxation of* (4.3) *is not worse than the objective value of the linear relaxation of* (4.2)*, i.e.,* $z_P^* \geq z_A^*$.

Equivalence of both relaxations does not hold. This can be shown by the instance displayed in Figure 4.3(a) with $v_p = 1$ for all paths. In the single fiber case with uniform fiber spectra of size $|\Lambda| = 2$ and $k_\ell = 1$ for all links, one wavelength converter is clearly needed. The linear relaxation of (4.2) has the value zero, according to Lemma 4.14. In contrast, the linear relaxation of (4.3) has value one. To see this, note that in single-fiber cases each path packing cannot cover each link more than once. Since all links in the instance are traversed by two lightpaths and (in sum) at most $|\Lambda| = 2$ path packings can be taken, only path packings covering all links can occur in any (fractional) solution. Any such path packing indeed must contain single links as subpaths, cf. Figure 4.3(b).
In fact, by extending the star network as displayed in Figure 4.3(c), where a converter has to be placed in each node with degree three for any feasible wavelength assignment with $|\Lambda| = 2$ (and thus $z_P^* = \frac{1}{2}(|V| - 2)$, while $z_A^* = 0$ still), the following result is obtained:

Lemma 4.16 *The difference* $z_P^* - z_A^*$ *can be arbitrarily large, even if only two wavelengths are involved.*

For even $|\Lambda| > 2$, the example of Figure 4.3(a) can be alternatively generalized by setting $v_p = \frac{1}{2}|\Lambda|$ for each of the three lightpath requests. Then $z_P = z_P^* = \frac{1}{2}|\Lambda|$, whereas again $z_A^* = 0$. As a result, the difference $z_P^* - z_A^*$ is of order $\mathcal{O}(|\Lambda|)$ and hence can be arbitrarily large even for single-fiber MCWAP on a simple star network.

Path packing formulation strength. The comparison of the linear realaxtion for both MCWAP models in Lemma 4.15 indicates that the path packing formulation is preferrable due to the better LP value. In fact, the computational experiments documented in Koster and Zymolka [96] show that in those cases where we know the optimal MCWAP solution, the value of the linear relaxation of the path packing

formulation (4.3) equals the optimal integer value. But this is not always the case. For construction of a counterexample, we come back to some results from Section 4.2 about the close relation of MCWAP and graph coloring.
By the proof of Theorem 4.5, equivalence of vertex coloring and single-fiber MCWAP with uniform fiber spectra in multi-chain networks has been shown. For any graph G, an instance $(\mathcal{N}_G, P_G, \Lambda, K)$ has been constructed such that G is $|\Lambda|$-colorable if and only if a MCWAP solution without converters exists.
Recall that the chromatic number $\chi(G)$ of a graph G is the minimum number of stable sets such that all vertices are covered. If we allow to take stable sets in fractional amounts under the restriction that the sum of the fractions still covers all vertices, we obtain the *fractional chromatic number* $\chi^*(G)$ (cf. Scheinerman and Ullman [148]). In terms of integer linear programming, the fractional chromatic number is the value of the linear relaxation of the stable set formulation for vertex coloring by Mehrotra and Trick [115]. The difference between $\chi(G)$ and $\chi^*(G)$ can be arbitrarily large for some graph classes, see Molloy and Reed [121]. With such graphs, we obtain the following result:

Theorem 4.17 *Let $G = (V, E)$ be a graph with $2 \leq \chi^*(G) \leq \chi(G) - 1$, and let $(\mathcal{N}_G, P_G, \Lambda, K)$ be the corresponding* MCWAP *instance as constructed in the proof of Theorem 4.5. For this instance with $|\Lambda| = \chi(G) - 1$, we get $z_P^* = 0$, whereas $z_P > 0$.*

Proof. The fractional chromatic number $\chi^*(G)$ is attained by a collection $\mathcal{B}$ of stable sets $B \subset V$ in G, each one taken with a fractional value $0 < w_B \leq 1$, such that $\sum_{B \ni v} w_B = 1$ for all vertices $v \in V$. From such an optimal solution of the vertex coloring relaxation, we construct a solution of the relaxation of (4.3) for the MCWAP instance $(\mathcal{N}_G, P_G, \Lambda, K)$ with $|\Lambda| = \chi(G) - 1 \geq 2$, which suffices to accommodate all paths due to the load $l \leq 2$ by the instance definition.
For each $B \in \mathcal{B}$, we define the corresponding path packing ϕ_B by $m^s_{\phi_B} := 1$ if path $s = p_v$ for some $v \in B$, and $m^s_{\phi_B} := 0$ otherwise. We set $x_{\phi_B} := w_B$ for all $B \in \mathcal{B}$ and $x_\phi := 0$ for all other path packings. The solution is completed by $y^s_p := 1$ if $s = p_v$ for some $v \in V$, and $y^s_p := 0$ for all proper subpaths $s \in S_{p_v} \setminus \{p_v\}$. Note that the paths p_v are mutually different by construction, i.e., no parallel lightpaths occur, and $v_p = 1$ for all $p \in P_1$. Using this, the constructed solution is easily found feasible for the linear relaxation of (4.3) and has objective value $z_P^* = 0$.
Since G is not colorable with $\chi(G) - 1$ colors, the MCWAP instance with that number of wavelengths cannot be solved without converters. Hence, $z_P > 0$, completing the proof. ■

It is easy to find graphs for which $\chi^*(G) < \chi(G)$, e.g., for an odd cycle C_{2k+1} with $2k + 1$ vertices, we get $\chi^*(C_{2k+1}) = 2 + 1/k$ for any $k > 0$, whereas $\chi(C_{2k+1}) = 3$. Thus, $\chi^*(C_{2k+1}) < \chi(C_{2k+1})$ for $k \geq 2$. Another example is the Petersen graph with $2.5 = \chi^*(G) < \chi(G) = 3$. However, the gap $\chi(G) - \chi^*(G)$ for all these graphs is smaller than one, and thus we cannot apply Theorem 4.17.
Kneser graphs are a generalization of the Petersen graph, introduced by Kneser [89] in 1955. Given two positive integers n and k, the Kneser graph $KG_{n,k}$ is the graph whose vertices represent the cardinality k subsets of $\{1, \ldots, n\}$ and where two vertices are adjacent if and only if they correspond to disjoint subsets. $KG_{n,k}$ has $\binom{n}{k}$ vertices, each one with degree $\binom{n-k}{k}$ (we assume $\binom{n-k}{k} = 0$ if $n - k < k$). The Petersen graph equals to the case $n = 5$, $k = 2$. Kneser graphs turned out as a

challenging class in view of their chromatic number. It was first proven by Lovász [107] in 1978 that $\chi(KG_{n,k}) = n-2k+2$, whereas $\chi^*(KG_{n,k}) = \frac{n}{k}$, see Scheinerman and Ullman [148]. So, as suggested by Matoušek and Ziegler [111], for $n = 3k-1$, it holds that $\chi(KG_{3k-1,k}) = k+1$, whereas $\chi^*(KG_{3k-1,k}) = \frac{3k-1}{k} < 3$. Thus for $k \geq 3$, we find that $\chi^*(KG_{3k-1,k}) \leq \chi(KG_{3k-1,k}) - 1$ yielding a sufficient gap, and we can construct a Mcwap instance with the property of Theorem 4.17. In particular, there exists an edge coloring of $KG_{8,3}$ with $\Delta(KG_{8,3}) + 1 = 11$ colors, resulting in an optical network with 12 nodes and 336 links for which $z_P^* = 0$ whereas $z_P > 0$.

Hence, the path packing formulation does not provide the opportunity to match the integer optimum by its linear relaxation value in general.[1] Nevertheless, from the results in Lemmas 4.15 and 4.16, we can conclude that the path packing formulation is favorable compared to the assignment formulation in view of supporting the solving of Mcwap by additional information. Since the linear relaxation value of any formulation provides a dual bound on the optimum, we prefer the alternative that yields the better approximation. Hence, we focus on the path packing formulation in our algorithmic solution approaches described next.

4.4 Algorithms for Mcwap

The complexity results from Section 4.2 reveal Mcwap to be an $\mathcal{NP}$-hard problem except for very strong assumptions which are far off practical instance properties. As a consequence, we cannot expect to find suitable efficient exact methods unless $\mathcal{P} = \mathcal{NP}$.
Therefore, we follow a twofold approach. On the primal side, we describe several heuristics to generate feasible solutions for the problem. Their application is accompanied and guided by derivation of a lower bound for the optimum objective value as dual side approach. A good lower bound allows to rank the quality of any solution found and, in the best case when meeting a solution's objective value, can prove optimality. For this, we describe how to solve the linear relaxation of the path packing formulation, which has turned out as more promising model. The developed column generation method is finally extended to a full branch-and-price algorithm, followed by a report on preliminary computational experiences.

4.4.1 Heuristics

The intention in designing heuristics for a hard problem is to keep a balance between the quality of the outcome and the required computation time. We distinguish between constructive and iterative methods. Where constructive heuristics try

[1] Notice that this is not immediately clear, as this property would not turn an $\mathcal{NP}$-hard problem into a polynomially solvable linear program. In fact, the path packing formulation is not of polynomial size, and the associated pricing problem for column generation is an $\mathcal{NP}$-hard problem for itself, as will be shown in Section 4.4.2.
On the other hand, the counterexample by use of Kneser graphs has a strongly artificial flavor, exploiting specific properties of a constructed structure that will hardly occur in the routings of real-world networks. In this light, the initial suggestion still remains a motivating observation.

to generate good assignments from scratch, the iterative approaches start with an assignment and try to reduce the converter number by clever transformations. We present some basic strategies as well as extensions by which further advancements can be achieved. As common feature, all these algorithms base on the principle of sequential wavelength assignment.

4.4.1.1 Sequential wavelength assignment

The main idea for both the constructive and iterative heuristics is to assign wavelengths to the lightpaths in a sequential way, i.e., the lightpaths are processed one by one in a certain order. In each step, the locally best possible decision is to assign the wavelengths to the lightpath on turn such that the number of required converters for that path is minimized. Each time a lightpath has been processed, the availability of the assigned wavelengths is adapted, and we call the remaining problem a *residual* Mcwap *instance*. Even if we start with uniform wavelength capacities at the links, the remaining set of wavelengths differs from link to link during the process, and thus we have to find out how to compute a locally best assignment with respect to such non-uniform spectra of available wavelengths. Since the goal is to avoid converters whenever possible, a natural idea consists in reusing a wavelength on as much consecutive links as possible. This yields the following Greedy-like procedure:

> *Beginning with the first link, repeatedly select a wavelength that can be assigned as far as possible, until no links of the lightpath are left anymore.*

For breaking ties in case multiple wavelengths reach equally far, we select the first alternative wavelength according to an arbitrary, but prefixed ordering of the wavelengths. Algorithm 4.1 lists a pseudo-code for computing such an assignment for a single lightpath in a residual Mcwap instance. A simple argument shows that this procedure indeed generates the locally best assignment for a single lightpath:

Lemma 4.18 *For a (residual)* Mcwap *instance* $(\mathcal{N}, P, \Lambda, K)$ *with arbitrary wavelength multiplicities* $k_\ell^\lambda \in \mathbb{Z}_+$ *for all* $\lambda \in \Lambda$, $\ell \in L$, *and an arbitrarily selected path* $p \in P$, *Algorithm* (4.1) *generates a wavelength assignment for* p *with a minimum number of converters.*

Proof. For path $p = (n_0, \ell_1, n_1, \ldots, \ell_h, n_h)$, let $\hat{z}(p)$ be the least possible number of converters in any feasible solution for the residual Mcwap instance and $z(p)$ the number of converters placed by Algorithm 4.1. By the optimality assumption of $\hat{z}(p)$, we have $\hat{z}(p) \leq z(p)$.
Algorithm 4.1 starts by selecting a farthest reaching wavelength. If this wavelength can be assigned to all links of the path, a converter-free and thus optimal assignment has been found. Otherwise, let $n_{i_1} \in N(p)$ be the last node reached. Then in any feasible wavelength assignment for path p in the residual Mcwap instance, a wavelength converter has to be placed in one of the nodes n_{j_1} with $j_1 \leq i_1$ due to the policy of selecting a farthest reaching wavelength. Since Algorithm 4.1 continues again with a farthest reaching wavelength, the same argumentation applies (iteratively) for each converter node n_{i_m} and the m-th converter node n_{j_m} with $j_m \leq i_m$ in any feasible assignment, finally needing at least the same number of converters as the above procedure places. As a conclusion, $z(p) \leq \hat{z}(p)$, completing the proof. ■

Algorithm 4.1 Locally optimal single lightpath wavelength assignment

Require: A path p with consecutively ordered link set $L(p) = \{\ell_1, \ldots, \ell_h\}$ and wavelength multiplicities $k_\ell^\lambda \in \mathbb{N}$ for all $\ell \in L(p)$, $\lambda \in \Lambda = \{\lambda_1, \ldots, \lambda_w\}$ ordered arbitrarily.

Ensure: A feasible wavelength assignment $\alpha_p : L(p) \to \Lambda$ with minimum number of converters.

1: set $i := 1$
2: **while** $i \leq h$ **do**
3: set $j := 0$
4: $\Lambda_r = \left\{ \lambda \in \Lambda \mid k_{\ell_i}^\lambda > 0 \right\}$
5: **while** $(i + j + 1 \leq h) \cap (\exists\, \lambda \in \Lambda_r \;:\; k_{\ell_{i+j+1}}^\lambda > 0)$ **do**
6: $\Lambda_r \leftarrow \Lambda_r \setminus \left\{ \lambda \in \Lambda_r \mid k_{\ell_{i+j+1}}^\lambda = 0 \right\}$
7: $j \leftarrow j + 1$
8: **end while**
9: set $s := \min \{\, i \mid \lambda_i \in \Lambda_r \,\}$
10: **for** $m = 0$ to j **do**
11: set $\alpha_p(\ell_{i+m}) := \lambda_s$
12: **end for**
13: $i \leftarrow i + j$
14: **end while**
15: return α_p

Algorithm 4.1 is of order $\mathcal{O}(\,|L(p)| \cdot |\Lambda|\,)$ and thus offers a polynomial procedure to compute the locally best assignment for a single lightpath.

Lightpath order. Using Algorithm 4.1, sequential methods for generating Mcwap solutions are completed by specifying the order in which the lightpaths are to be processed. For each selected sequence, the resulting Mcwap solution follows deterministically. As a consequence, the obtained objective value as required number of converters depends only on the processing order of the lightpaths. Hence, a natural question concerns the correspondence between the set of all lightpath sequences with their resulting solutions and the set of all Mcwap solutions, in particular optimal ones.

As Mcwap is $\mathcal{NP}$-hard, a guaranteed optimal sequence cannot be expected to be determinable in polynomial time. In fact, we show that there need not exist such an ordering at all. Consider the (pathological) example in Figure 4.4, where two chains as parts of an optical network are displayed. In each part, we have four consecutive links on which three parallel lightpaths spanning all links have to be established. Each row represents a certain wavelength, which is available on a link if its endnodes are connected in this row. Note that both parts differ only in exchanged availability of wavelengths λ_2 and λ_3. Moreover, thin lines illustrate the lightpaths constructed by using the farthest reaching wavelength method according to the wavelength ordering $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. Changing a row means to place a converter at the appropriate node. While in the left part, only three converters are needed, the right solution requires five converters. Reordering the wavelengths to $(\lambda_1, \lambda_3, \lambda_2, \lambda_4)$ simply exchanges both parts. Each other order of the wavelengths results in a solution with at least the same number of converters or even more, while

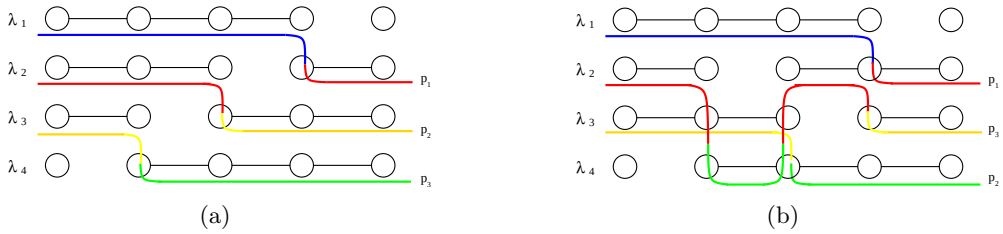


Figure 4.4: *Example for an instance, where a sequential heuristic finds an optimal wavelength assignment in (a), but not in (b).*

an optimal solution for the whole network clearly uses only six converters (applying different wavelength orders in the parts). This shows that there exist instances of the minimum converter wavelength assignment problem for which not all solutions can be generated by a sequential method. As a consequence, the space of all solutions that can be generated by a sequential procedure may be restricted to a subset of all feasible assignments. Especially, it is not sure that it contains an optimal solution.
On the other hand, however, sequential wavelength assignment has promising features in case an optimal assignment needs only few converters. If a wavelength assignment without converters exists, then there exists an ordering of the lightpaths such that the sequential wavelength assignment algorithm finds an optimal solution. To see this, notice that an optimal assignment implies such an ordering. Arrange all lightpaths using the same wavelength in the solution consecutively. Then the sequential procedure finds the optimal assignment (without converters) if the wavelengths are assigned in the same order as they appear in the lightpath sequence.
In case an optimal solution needs a single converter, a similar argument applies. For such a fixed optimal solution, consider the only lightpath that needs a converter. Order all wavelengths and lightpaths as before except for the converted lightpath which is processed last, and place the wavelength used in the solution after the conversion as last, whereas that used before conversion becomes the last but one. This way, again no converter is required by the sequential method until the last lightpath is on turn. By the wavelength ordering, an assignment as in the solution is then still possible for the last lightpath, and thus not more than one converter will be placed by Algorithm 4.1 with this ordering. Hence, the constructed sequence generates an optimal solution.
This result can be generalized to cases where only few converters are needed each by a different lightpath in different parts of the network. So, for many instances where the number of required converters is small, lightpath sequences resulting in an optimal solution are likely to exist.

The previous sequence constructions have been derived from knowing an optimal solution to (re-) generate. Without such an input, promising sequences have to be specified by other means, as discussed next.

4.4.1.2 Constructive methods

The construction of a Mcwap solution based on sequential wavelength assignment requires just a single run processing each lightpath once by Algorithm 4.1. For this, the lightpath processing order is to be specified, either statically by a priori fixing the

entire sequence or dynamically by selecting the next lightpath to process according to some rule. Both variants occur by application of rules with greedy character.

Greedy-based sequences. Among the many possibilities, three Greedy ordering rules for the lightpaths have been implemented and tested:

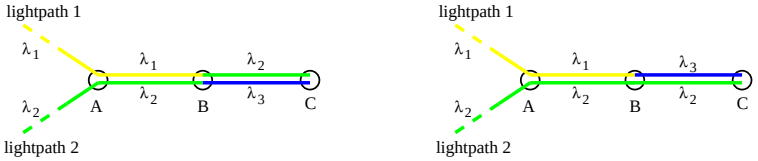
- longest path first
- most inflexible path first
- most inflexible longest path first

The *longest path first rule* selects among the not yet processed lightpaths one that contains the largest number of links. Ties are broken arbitrarily. The underlying idea is that the step-wise reduction of wavelength availabilities makes it in particular for long lightpaths more and more difficult to avoid need for converters. Note that this rule yields a static ordering, i.e., the sequence can be determined before the algorithm is executed.
The *most inflexible path first rule* selects among the not yet processed lightpaths one for which the number of continuing wavelengths is minimal. In case each remaining lightpath can still get a wavelength assignment without conversion, lightpaths with scarce availability of common wavelengths at the complete path are selected first. The idea is that those lightpaths are most likely to need converters if other assignments would further reduce the set of available (ongoing) wavelengths. This ordering is dynamic, since the selection of the next lightpath to be processed is based on information generated by the previous assignments.
The *most inflexible longest path rule* combines the two previous rules. Among the most inflexible paths, a longest one is selected. In this way, the increased risk for the need of wavelength conversion on long paths with a scarcity of wavelengths is taken into account. As for the previous rule, this ordering is also dynamic.

Evaluation. A computational evaluation of the constructive heuristics has shown that none of the tested rules proved to be superior to the others, i.e., there is always an instance where a rule is outperformed by other rules (see Koster and Zymolka [93] for the detailed study). This observation indicates that instances may offer structural preferences favoring any of the specific rules. The outcome in number of converters reflects these differences, but since the method terminates after the assignment is completed, no further advantage of such additional information is taken. To do so, the possibility to reorder a sequence on the basis of the actual assignment is therefore an attractive idea.

4.4.1.3 Iterative methods

Given a Mcwap solution generated by sequential wavelength assignment, the number of converters might be reducible by a (partial) exchange of the wavelengths assigned to two lightpaths. Such an exchange can often be acquired by interchanging the position of the lightpaths in the processing order, as shown in the following example.
Consider the situation in Figure 4.5. There are two lightpaths with different starting points, but both use the links AB, BC and end at node C. The wavelengths are



(a) *Lightpath 1 before lightpath 2.*

(b) *Lightpath 2 before lightpath 1.*

Figure 4.5: *Example for improvement by reordering the lightpath sequence.*

ordered as $\lambda_1, \lambda_2, \lambda_3$. First suppose lightpath 1 is ordered before lightpath 2 and wavelength λ_1 is not available on link BC anymore, whereas wavelength λ_3 is not available on the links of lightpath 1 before node B. So, wavelength λ_1 is assigned to lightpath 1 on all links up to B, and wavelength λ_2 is used for link BC. Next, wavelengths are assigned to lightpath 2. Since wavelength λ_2 is not available on BC anymore, lightpath 2 gets wavelength λ_2 on the links up to B, but then needs to use wavelength λ_3 on link BC. This assignment requiring two converters in node B is shown in Figure 4.5(a).
However, a reordering of the lightpaths allows for the reduction of the number of converters by one. If processed first, lightpath 2 gets wavelength λ_2 on all links, whereas lightpath 1 is assigned wavelength λ_1 up to node B and wavelength λ_3 on its last link BC. This solution is shown in Figure 4.5(b).

Iterative approach. The example shows that the solution of a sequential method can indicate possible improvements of the processing sequence. We exploit this observation by an iterative approach, repeatedly processing lightpath sequences which are reordered in view of the previously required converters. In addition, we presume to have a lower bound on the converter number. Such a bound can be determined by the methods described in Section 4.4.2 (or is simply set to zero). In the best case, a solution matches the bound and is thus identified as optimal solution, terminating the search. The general outline of an iterative procedure is as follows.

Iterative sequential wavelength assignment.

Input. A MCWAP instance $(\mathcal{N}, P, \Lambda, K)$ as defined in Problem 4.3, a lower bound $B \geq 0$ on the objective value, and a computation time limit.

Method. The flow chart of the method is depicted in Figure 4.6. The iteration starts with an arbitrary initial ordering of the lightpaths, for instance generated randomly or as for the constructive methods. In each step, the current sequence is processed by sequential wavelength assignment using Algorithm 4.1 for each lightpath. If an assignment with in total B converters is obtained, the method terminates immediately (STOP 1) and returns this provably optimal solution. Otherwise, maybe too much converters are still required. In case the time limit is not yet exceeded, we reorder the sequence by moving lightpaths with converters to the front of the sequence according to some predefined scheme. The reordered sequence is then processed in the same way as next step. During the iteration, the procedure keeps track of the best found solution which is finally returned in case the time limit is exceeded (STOP 2).

Output. A MCWAP solution as defined in Problem 4.3.

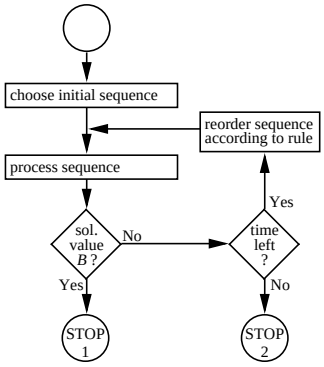


Figure 4.6: *Iterative sequential wavelength assignment flow chart.*

Note that this procedure is no local search method. In the best case, optimality follows directly from matching the lower bound. However, the lower bound need not equal the optimum value, and thus it can happen that an optimal solution is returned without having it identified as such one.
In any way, the method always returns a feasible solution for the Mcwap instance. Moreover, different initialization sequences can yield different results. Besides the initialization, another flexible issue is the scheme according to which the lightpath sequence is reordered.

Reordering schemes. As core step in the procedure, the reordering scheme encodes how information from a generated solution is exploited in order to achieve improvements in reordering the lightpath processing sequence. For our computational experiments, we implemented four variants:

- push the first lightpath with converter(s) to the beginning of the ordering (First Path Reordering, FPR)
- push the last lightpath with converter(s) to the beginning of the ordering (Last Path Reordering, LPR)
- push all lightpaths with converter(s) to the beginning of the ordering, remaining the order of these lightpaths (All Paths Reordering, APR)
- push all lightpaths with converter(s) to the beginning of the ordering, reversing the order of these lightpaths (All Path Reverse Reordering, APRR)

Pushing (even single) lightpaths to the beginning of the ordering may yield a completely different result, a worse as well as a much better one. However, the first lightpaths in the ordering usually do not need conversions, so that at least the prepended lightpath(s) with converters in the former solution have best chances to get rid of them.
In this way, the iterative method explores many sequences, each started with a slight push in a promising direction. This effect may be accelerated by the last two variants, which push many lightpaths at once to the beginning.

Evaluation. A comparison study of these methods over 24 Mcwap instances is documented in Koster and Zymolka [93]. In essence, Figure 4.7 shows some typical

converter number progressions during the iterations. Although observations are clearly restricted to our test set, the results give rise to several remarks. First of all, those variants that move multiple lightpaths at once outperform the other methods by far. The most impressive example is provided by Figure 4.7(a), while the other diagrams 4.7(b)–4.7(d) show gradually more similar behaviors. Note that the number of iterations required by the best performing method (APRR) increases in the same diagram order. The comparison results indicate that moving just a single lightpath to the sequence front does not change much in the converter number generated by the next iteration. A larger jump between two iterations rarely happens if the order is just slightly rearranged. The lightpath set moves (APR and APRR) allow for faster progress towards considerably better solutions. In fact, the APRR method seems to be most effective. Compared to APR, reversing the order of the moved lightpaths brings additional motion into the lightpath sequence reorderings, which seems to have a beneficial effect.
Within the time limit of ten minutes, 16 of the instances have been solved by at least one of the methods (APRR always among them) to optimality, detecting a converter-free assignment. Figure 4.7(d) shows one of the remaining cases where the lower bound was not matched during the iterations. In three of these instances, the gap between the best found solution and the lower bound is less than three, showing the generated solutions to be close to optimal. However, this gap could not always be fully closed and thus gives rise to look for further advancements of the methods.

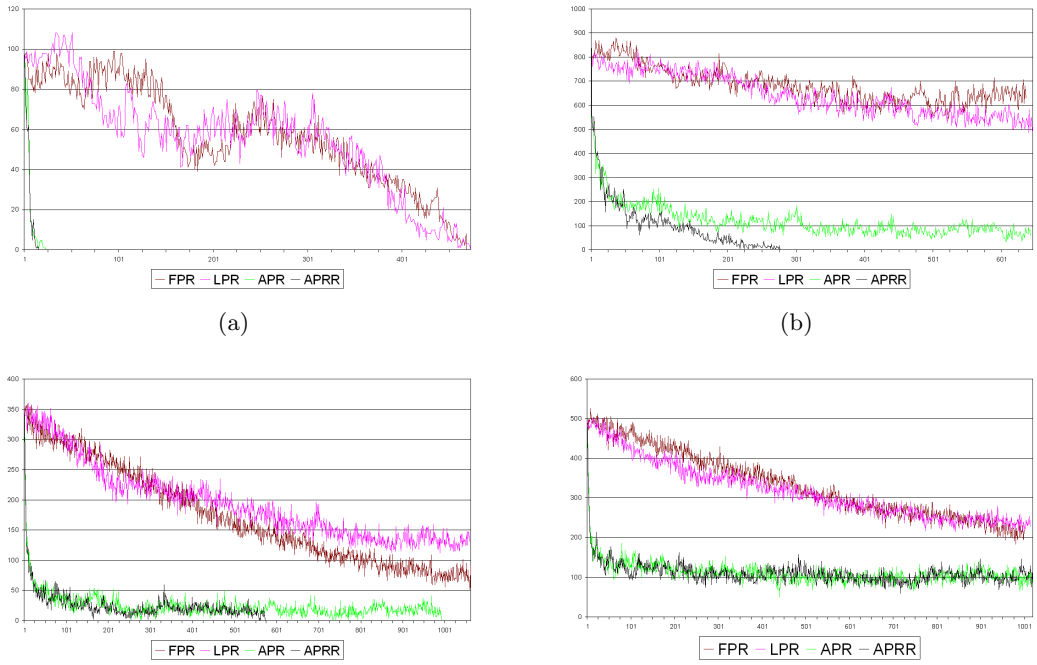


Figure 4.7: *Converter number progressions of the iterative methods for some selected instances.*

4.4.1.4 Wavelength extraction

A promising idea for a refinement of solution methods is the principle of problem reduction by fixing of partial solutions. Many optimization methods, such as branch-and-bound, make systematical use of this principle. For Mcwap, the following observation offers a heuristic approach in this direction.

Extracting wavelengths. Assume we have found a solution with a converter number $z < \frac{1}{2}|\Lambda|$. Since any converter affects exactly two different wavelengths, the solution contains some wavelengths that are never converted from or to. Such wavelengths are assigned exclusively to complete lightpaths. Restricting to (any subset of) these wavelengths and the corresponding lightpaths, a converter-free partial solution is obtained. As the objective value cannot be improved on such a partial solution, it is reasonable to fix it and to continue with the remaining instance, reflecting a 'critical part' of the problem where converters are still needed and thus further improvements can be expected. This process, called *wavelength extraction*, applies also to solutions with a number of converters larger than $\frac{1}{2}|\Lambda|$ as long as unconverted wavelengths occur to form an unimprovable problem part to be fixed. Note that the original solution guarantees existence of feasible solutions for any reduced instances generated by extracting a subset of wavelengths together with their associated lightpaths from any feasible solution of the original problem at hand.
By a wavelength extraction, the space of producible solutions is shrunken which can, in the worst case, exclude all optimal solutions, too. The probability for such a case keeps low as long as only unconverted wavelengths are extracted. Moreover, the extraction of complete wavelengths from an instance (solution) is just one reduction possibility, among many others. As special feature, this approach maintains structural properties from the original instance, such as uniform availability of wavelengths. In this light, the proposed wavelength extraction approach provides a good compromise between sacrifice of accuracy and gain of tractability by a considerable problem size reduction.

Improving heuristics. As a generic problem reduction principle, wavelength extraction can be applied in combination with the previously described heuristics. Whenever a solution with unconverted wavelengths is detected, (a subset of) unconverted wavelengths with the associated lightpaths can be removed, executing a heuristic recursively on the reduced instance. A preliminary study of this methodology has been carried out in Koster and Zymolka [96]. From 14 instances in total, in four cases an iterative heuristic run over 6000 seconds CPU time could not find a solution matching the lower bound. So, wavelength extraction was repeatedly applied for the best found solutions, removing all unconverted wavelengths and running the iterative heuristic again for the same time. One instance did not allow for progress this way, but two further instances have been solved to optimality within two iterations, and for the last instance, the gap between the best known solution and the lower bound was reduced by a total of 83% after three iterations. These results motivate to consider a more elaborate way to exploit wavelength extraction within the heuristics.
Since the iterative heuristics generate a complete solution in each iteration, wavelength extraction can be applied as a built-in feature. For this, we keep track of

the (number of) converted wavelengths in the actual assignment and initiate an extraction step according to some guiding criteria. Such extractions can be applied recursively. For any recursive call, we set a stop limit for searching the reduced instance, by a time bound, a maximum iteration number, or both in a competing manner. If no improvement was found when reaching the limit, we return to the larger instance of the invoking run and continue its iterative processing, seeking for further (partial) assignments promising to be extracted. Otherwise, a better partial solution found somewhere during the iterations can always be extended to the associated complete solution by adding the formerly fixed assignments. This way, a more elaborately guided search of the solution space is realized, exploiting problem size reductions in a useful manner.

A computational study of iterative heuristics with integrated wavelength extraction is presented in the next chapter. Here, we close the discussion of upper bound algorithms by heuristics and turn to their counterpart, lower bounds for Mcwap.

4.4.2 Lower bounds

$\mathcal{NP}$-hard problems do not allow for efficient computation of optimal solutions in general. Hence, the derivation of good lower bounds is of crucial importance. Such an additional information does not only allow to rate the quality of any solution for the problem, but can also be a valuable guide for solution algorithms, as already stated in the preceding section.
In this section, we study the computation of lower bounds for Mcwap. Having an integer linear program formulation at hand, a standard approach to obtain a dual bound consists of solving its linear relaxation. Moreover, a recent combinatorial lower bound from Koster [91] is discussed.

4.4.2.1 Linear relaxation values

For Mcwap, two integer linear programming models are presented in Section 4.3. Their comparison in Lemma 4.15 and Lemma 4.16 shows that the assignment formulation is dominated by the path packing formulation in terms of the linear relaxation value. Thus, we focus on the path packing model (4.3) and propose a column generation method to solve its linear relaxation. For the general outline of the method, we restrict to the uniform fiber spectra case and describe appropriate adaptions to general instances afterwards. As a first step, we describe a model transformation by which the size of the linear program is (in theory) substantially reduced.

Model transformation. In the original form, the path packing formulation (4.3) on page 151 is tremendously large since it holds an individual variable for each possible path packing. This is necessary due to the requirement that the selected path packings provide exactly the subpaths used for lightpath partitionings. On one hand, letting the path packings provide more subpaths than strictly required does not invalidate any solution, as turning equalities (4.3c) into '$\leq$'-inequalities yields a relaxation of the original model. In fact, any solution of the relaxed model could just offer multiple wavelengths for some partitioning subpaths to generate

the individual lightpath assignments. For this relaxation, on the other hand, path packings that are not maximal with respect to set inclusion become dispensable, since any solution including such path packings can be easily transformed to a solution with same objective value using exclusively inclusion-maximal path packings (by filling up the selected packings). Hence, the x_ϕ-variables can be restricted to the subset of inclusion-maximal path packings. As a consequence, the described transformation implies a considerable model size reduction.
Let $\hat{\Phi} \subset \Phi$ be the set of all path packings that are maximal with respect to set inclusion, i.e., $\forall\ \hat{\phi} \in \hat{\Phi},\ \phi \in \Phi\ :\ \hat{\phi} \subset \phi \Rightarrow \hat{\phi} = \phi$. Then the alternative path packing model for Mcwap reads:

$$\hat{z}_P = \min \sum_{p \in P_1} \sum_{s \in S_p} y_p^s \qquad \left[- \sum_{p \in P_1} v_p \right] \tag{4.4a}$$

$$\text{s.t.} \quad \sum_{\substack{s \in S_p: \\ \ell \in L(s)}} y_p^s = v_p \qquad \forall\ p \in P_1,\ \ell \in L(p) \tag{4.4b}$$

$$\sum_{\substack{p \in P_1: \\ s \in S_p}} y_p^s \leq \sum_{\hat{\phi} \in \hat{\Phi}} m_{\hat{\phi}}^s x_{\hat{\phi}} \qquad \forall\ s \in S \tag{4.4c}$$

$$\sum_{\hat{\phi} \in \hat{\Phi}} x_{\hat{\phi}} \leq |\Lambda| \tag{4.4d}$$

$$y_p^s, x_{\hat{\phi}} \in \mathbb{Z}_+ \qquad \forall\ p \in P_1,\ s \in S_p,\ \hat{\phi} \in \hat{\Phi} \tag{4.4e}$$

Both the objective (4.4a) and the lightpath partitioning constraints (4.4b) are carried over unchanged from the original model (4.3). Constraints (4.4c) result from relaxing constraints (4.3c) to inequalities, in order to allow for overprovision of subpaths, and the additional restriction to inclusion-maximal path packings, which applies for constraint (4.4d) as well. As usual, $\hat{z}_P^*$ denotes the linear relaxation value of (4.4).
Due to the variable set reduction, the transformation does not yield a relaxation of (4.3). So, it is not immediately clear how the solutions of both models are related to each other, in particular in view of their linear relaxations. The following lemma shows that the optimal values of both models still coincide.

Lemma 4.19 *The path packings models* (4.3) *and* (4.4) *have the same optimum value both as integer linear programs and as linear relaxations, i.e.,* $z_P = \hat{z}_P$ *and* $z_P^* = \hat{z}_P^*$ *for all instances.*

Proof. We first consider the linear relaxations and show that any feasible solution of (4.3) can be transformed to a feasible solution of (4.4) with the same objective value, and vice versa.
Consider an arbitrary fractional solution of the linear relaxation of (4.3). Any path packing ϕ with $x_\phi > 0$ can be filled up by adding single-link subpaths for any link with unoccupied wavelengths to an inclusion-maximal path packing $\hat{\phi} \in \hat{\Phi}$ with $\phi \subset \hat{\phi}$. Setting $x_{\hat{\phi}} := x_\phi$ for all these packings, we get a solution of (4.4) with same objective value, as the y-variables have not been changed.
Unfortunately, the reversed transformation to turn strictly satisfied inequalities (4.4c) into equalities does not work that simple. Removing paths from a packing $\hat{\phi}$ decreases the right-hand side of such an inequality by integer multiples of the

corresponding variable $x_{\hat{\phi}}$ which might fail to close the difference to the left hand side exactly (or even violate the former inequality). Instead, we modify a solution of (4.4) iteratively as follows. For each subpath $s \in S$, define the corresponding inequality slack

$$\Delta_s := \sum_{\phi \in \Phi} m_\phi^s x_\phi - \sum_{\substack{p \in P_1: \\ s \in S_p}} y_p^s \geq 0 \; .$$

Let $s \in S$ be an arbitrary subpath with $\Delta_s > 0$. Then there exists a variable $x_\phi > 0$ (initially in fact $\phi \in \hat{\Phi}$) with $m_\phi^s > 0$. Let ϕ' be the path packing obtained from ϕ by decrementing the multiplicity for s by one, i.e., $m_{\phi'}^s := m_\phi^s - 1$ for s and $m_{\phi'}^{s'} := m_{\phi'}^{s'}$ for all $s' \in S \setminus \{s\}$. Then we reset the variable x_ϕ to $x_\phi := \max\{0, x_\phi - \Delta_s\}$ and $x_{\phi'}$ to $x_{\phi'} := x_{\phi'} + \min\{x_\phi, \Delta_s\}$. This setting increases $x_{\phi'}$ by the same value as x_ϕ decreases, namely by Δ_s if $x_\phi \geq \Delta_s$ and by x_ϕ otherwise. By the same (positive) value, the transformation reduces the slack of the inequality for s (either to zero or to $\Delta_s - x_\phi > 0$) due to the definition of ϕ', while all other inequality slacks remain unchanged. This procedure is repeated as long as $s \in S$ with $\Delta_s > 0$ exists. In the end, we get $\Delta_s = 0$ for all $s \in S$, a feasible solution for the linear relaxation of (4.3). Thereby, the solution value is unaffected by any step since all y-variables keep their value.
For the integer linear programs, notice that $\Delta_s \in \mathbb{Z}_+$ for all subpaths $s \in S$. Applying the same transformations as for the linear relaxation solutions, all x-variable value changes occur in integer amounts and thus maintain integrality of the solutions. Hence, equivalence of both integer linear programs follows by the same argumentation. ■

As a result, the transformed model, though much smaller, does not loose strength and provides the same lower bound for Mcwap as the original model. Hence, formulation (4.4) is favorable for computational purposes.

Column generation approach. Despite the reduction, $\hat{\Phi}$ remains a set of exponential size, and thus the transformed path packing formulation is still too large to be handled explicitly. Therefore, we propose a column generation method to solve its linear relaxation. For a basic introduction to column generation and related techniques, we refer to Chvàtal [32] and Desrosier and Lübbecke [37].
For (4.4), we apply column generation for the x-variables, whereas all y-variables are considered explicitly. Let $\overline{\Phi} \subset \hat{\Phi}$ denote the subset of path packings whose associated variables are already included in the current linear program. Having determined the optimal solution of this restricted linear program, we have to solve the so-called *pricing problem* to decide whether further variables are required.
To formulate the pricing problem for (4.4), we introduce the dual variables π_ℓ^p, π^s, and π^Λ for the constraints (4.4b), (4.4c), and (4.4d), respectively. A primal-dual pair $((x, y), \pi)$ is optimal for the linear programming relaxation of (4.4) if and only if $c - A^T\pi \leq 0$, where c is the primal objective function and A is the coefficient matrix of (4.4). For a path packing $\hat{\phi} \in \hat{\Phi}$, we have $c_{\hat{\phi}} = 0$, and the coefficients in A corresponding to (4.4b) are zero as well. So, the optimality condition reads in our case

$$-\sum_{s \in S} m_{\hat{\phi}}^s \pi^s \leq \pi^\Lambda \quad . \tag{4.5}$$

Note that $\pi^s \leq 0$ by (4.4c), whereas $\pi^\Lambda \geq 0$. If $((x, y), \pi)$ is in fact an optimal pair for the unrestricted problem, (4.5) holds for every $\hat{\phi} \in \hat{\Phi}$. To verify this, we search for a still not contained path packing $\hat{\phi} \in \hat{\Phi} \setminus \overline{\Phi}$ that maximizes the left hand side of (4.5). If the maximum value is less than or equal to π^Λ, then no improving columns exist, and the linear relaxation is solved to optimality. Otherwise, we have found a path packing $\phi \in \Phi$ that violates (4.5) and can be added to the linear program to improve the relaxation.
Representing the multiplicity function of a path packing, we introduce the following variables:

$m^s \in \mathbb{Z}_+$ denotes the number of times subpath $s \in S$ is contained in a path packing.

Then the pricing problem reads:

$$z = \max \sum_{s \in S} (-\pi^s) m^s \tag{4.6a}$$

$$\text{s.t.} \sum_{s \in S : \ell \in L(s)} m^s = k_\ell \qquad \forall\, \ell \in L \tag{4.6b}$$

$$m^s \in \mathbb{Z}_+ \qquad \forall\, s \in S \tag{4.6c}$$

Using Observation 4.1, inclusion-maximal path packings are characterized my meeting the provided wavelength multiplicities k_ℓ on all links $\ell \in L$ exactly, as demanded by equalities (4.6b). In fact, any path packing that occupies a link $\ell \in L$ less than k_ℓ times can be filled up by adding the subpath s with $L(s) = \{\ell\}$ as often as required to reach k_ℓ. In the other direction, any path packing that fully occupies all link capacities obviously cannot be extended by further paths and thus is inclusion-maximal. Hence, equalities (4.6b) together with non-negativity and integrality conditions (4.6c) ensure that the set of all feasible solutions $(m^s)_{s \in S}$ corresponds to the set of multiplicity functions of all inclusion-maximal path packings, and objective (4.6a) is equivalent to maximizing the left hand side of the optimality condition (4.5).
We briefly remark that the pricing problem for the original formulation (4.3) is quite similar. As only difference, the equalities (4.6b) are replaced by $\leq$-inequalities, expressing that the wavelength multiplicities must not be exceeded by the selected subpaths, which already characterizes any feasible path packing of subpaths. Note that this pricing problem forms a relaxation of (4.6). Moreover, both pricing problems have the same optimum value as long as $(-\pi^s) \geq 0$ for all $s \in S$, since then each optimum path packing as solution for the relaxed problem with inequalities in (4.6b) can be filled up to an inclusion-maximal path packing without decrease in the objective function value.
In case $k_\ell = 1$ for all $\ell \in L$, the pricing problem (4.6) reduces to a maximum weighted set partitioning, which is known to be $\mathcal{NP}$-hard (see Garey and Johnson [52]). Hence, the pricing problem modeled by (4.6) is as well $\mathcal{NP}$-hard in general. However, the pricing problem need not be solved to optimality in every iteration of the algorithm. It suffices to find a feasible solution with value strictly larger than π^Λ, e.g., by heuristics such as built-in primal heuristics of general purpose mixed-integer program solvers (which we in fact use in our implementation). Only if this fails, we have to solve (4.6) to optimality in order to guarantee for an exact evaluation of the

optimality condition (4.5) for the path packing formulation (4.4).
In general, a path packing represents a multi-set (of subpaths). From an abstract perspective, problem (4.6) can then also be interpreted as maximum weighted *multi-set partitioning* and the relaxation with inequalities for (4.6b) as maximum weighted *multi-set packing*, representing the multi-set generalizations of the well-known set partitioning and set packing problem, respectively. In this light, multi-set partitionings and multi-set packings form generic structures which provide an interesting field for further research. As a first step, a special variant of multi-set packing with restriction to pairwise subpath conflicts has been studied as *stable multi-sets* in Koster and Zymolka [92, 95]. Similar investigations for multi-set partitioning and multi-set packing could be helpful in improving the solving process for the pricing problems and, due to their generic structure, for many other problems as well.

Non-uniform fiber spectra. In view of the generalization of the path packing formulation to non-uniform fiber spectra, the pricing problem can be easily adapted to generate columns for a particular wavelength subset Λ^i, as introduced in Section 4.3.2. To this end, we have only to replace the wavelength multiplicities in conditions (4.6b) by those for the considered subset, i.e., instead of k_ℓ, the right-hand sides are given by k_ℓ^λ for an arbitrary $\lambda \in \Lambda^i$. As a result, we obtain w different pricing problems, one for each wavelength subset. In each iteration, all these problems have to be solved in order to check whether there is an improving column for each wavelength subset individually. Since this adaption of the pricing procedure is straightforward, all further modifications and discussions can be applied on each of the individual pricing problems as well, and we continue to describe only the case of uniform fiber spectra.

Path number bounds. The solution space of pricing problem (4.6) comprises all inclusion-maximal path packings that do not exceed the wavelength multiplicities on the links. However, such a path packing can be of limited use if it contains a subpath $s \in S$ more often than it occurs as partial path of the given lightpaths. The corresponding packings, though included in (4.4), can in principle be replaced in any solution by packings in which the unusable paths are substituted by an appropriate multi-set of single-link subpaths (which are always usable for partitionings). In order to avoid pricing of columns with useless subpaths, additional bounds on the number of times a subpath can be packed are introduced.
For a subpath $s \in S$, the maximum utilization multiplicity is given by

$$b^s := \sum_{p \in P_1 : s \in S_p} v_p \quad ,$$

the total number of lightpaths that contain s as subpath. Hence, we extend the formulation by restricting the multiplicities of multiple link subpaths in the packings as

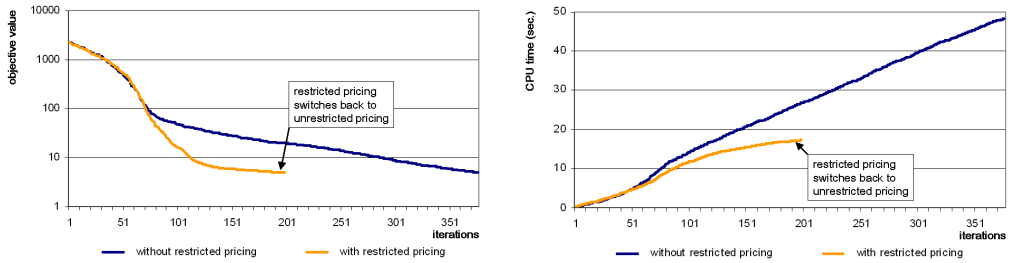
$$m^s \leq b^s \qquad \forall\, s \in S \;:\; |L(s)| \geq 2 \tag{4.6d}$$

and incorporate these conditions whenever we refer to the pricing model (4.6).

Restricted pricing. The bounding of subpath multiplicities inspires a further idea to improve pricing of valuable packings. Selecting a packing containing subpaths that have to be used as strict partial paths for partitioning of the lightpaths

directly implies a need for converters. To avoid this implication, a packing has to provide only full lightpaths as subpaths, each at most v_p times. In the wavelength assignment, the corresponding wavelength is then assigned exclusively to entire lightpaths. Such packings are most likely to be part of optimal solutions, in particular in case only few converters are needed in total.
For exploiting this observation, we apply the following strategy for pricing, referred to as *restricted pricing*. We start with the generation of path packings that are exclusively composed of entire lightpaths, called *restricted path packings*. For this, we bound the variables m^s by v_p if $s = p$ for a lightpath $p \in P_1$, and by 0 otherwise. Note that this only requires to modify conditions (4.6d) on the right-hand side. Moreover, single-link partial paths remain selectable and guarantee that maximal restricted path packings can be obtained this way. As soon as no further restricted path packings as improving columns are found, the variable upper bounds in the pricing problem are lifted from 0 or v_s to b^s, continuing with the usual column generation until reaching the optimum. For $z_P^* = 0$, restricted pricing could suffice to construct an optimal solution, and no proof of optimality is required when 0 is finally reached. In case $z_P^* > 0$, a change to unrestricted pricing is typically necessary at some earlier point to provide for the optimal solution also packings with strict partial paths (if not all of these occur also as full lightpaths). However, usually few final iterations with unrestricted pricing are needed for completing the procedure.
Using restricted pricing turns out to be a very profitable idea for solving the linear relaxation more quickly, as a computational evaluation in Koster and Zymolka [94] has indicated. Out of a test set of 80 instances, a converter-free and thus optimal solution was found by an iterative heuristic in totally 57 cases. For the remaining 23 instances, the lower bound obtained from the linear relaxation of (4.4) was computed both without and with restricted pricing. The results reveal the benefit of applying restricted pricing as long as possible. Overall, reductions above 50% in the total number of generated columns are typical, CPU time reductions of more than 80% not seldom. Figure 4.8 illustrates characteristic progressions of usual column generation and that by use of restricted pricing. In this example, restricted pricing was applied for the first 194 of totally 198 iterations. In the beginning, both strategies show a similar behavior. When approaching the LP solution (being $z_P^* = 5$ in this case), restricted pricing sheers out of the common progression track, providing faster pricing as well as faster progress towards the optimum value. At this phase, the advantage of restricted pricing takes effect. While the usual procedure grinds the final solution out by collecting many unrestricted and thus semi-helpful path packings at a constant rate, the restricted pricing has soon generated most necessary ingredients to compose the optimal fractional wavelength assignment. This is also indicated by the lower number of non-zero path packing variables x_ϕ in the optimal solution, in this case 179 compared to 236 by the usual pricing strategy. If restricted path packings cannot be found anymore, switching back to unrestricted pricing is just necessary at the very end for a couple of iterations to prove optimality.

Initialization. By use of column generation, we need not setup the complete program for solving the linear relaxation. In every iteration, the set of path packings considered explicitly is extended, exploiting the dual information of the actual solution. It is, however, unclear with which columns the linear program should be initialized as to minimize the number of iterations and computation time. We de-



(a) *LP value.* (b) *CPU time.*

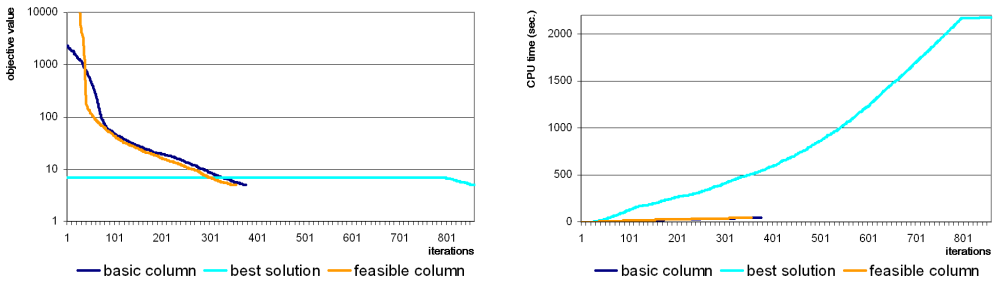
Figure 4.8: *Iterative progress of LP value (in logarithmic scale) and accumulated CPU time (in seconds) without and with the restricted pricing strategy in a typical case.*

scribe three promising strategies that have been evaluated and compared.

The path packing formulation (4.4) allows to initialize the master problem with a single path packing. For every link $\ell \in L$, we insert the subpath consisting only of this single link k_ℓ times. Obviously, the resulting path packing is feasible and inclusion-maximal. We refer to this path packing as the *basic column* in the linear program as well as to the strategy using only this column for initialization. This column can be taken $|\Lambda|$ times to obtain a feasible (integer) solution in which every lightpath is converted at every intermediate node. Hence, the initial solution value is the worst possible (integer) one as every path $p \in P_1$ contributes $(|L(p)| - 1)v_p$ to the objective.

Alternatively, the linear program can be initialized with a feasible wavelength assignment obtained by any heuristic method. Given such a solution, we have to construct $|\Lambda|$ path packings consisting of all subpaths to which the same wavelength is assigned and maximize them by adding single-link subpaths as often as needed to meet the provided wavelength multiplicities. In the strategy *best solution*, we use the best known solution for this construction. This way, the initial value of the master program becomes the number of converters in the solution and thus the best possible objective value known so far.

Both strategies described above start with a feasible (integer) solution by provision of a number of suitable path packings. A drawback of such an initialization is the necessity to find new columns that can be appropriately combined with the already included columns. To gradually construct a feasible (fractional) solution from scratch, the linear program has to be initialized without any path packing. However, in that case the first master problem is infeasible and typically no dual information is generated by the linear programming solver, which makes it impossible to solve the first pricing problem. A remedy lies in the usage of the basic column in a different way. We add a column to the formulation that is equivalent to the basic column, except for the objective coefficient and the contribution to the spectrum bound inequality (4.4d). This so-called *feasibility column* gets coefficient 0 in the spectrum bound constraint (4.4d) and thus does not represent a wavelength. Moreover, the column gets a huge objective penalty. This guarantees that the first master problem is feasible by selecting the feasibility column (at most) $|\Lambda|$ times. So, the first LP value will be very high. In the next iterations, columns are generated that reduce utilization of the feasibility column more and more until a (fractional) solution is found making the feasibility column obsolete. Initializing the linear program this



(a) *LP value.* (b) *CPU time.*

Figure 4.9: *Iterative progress of LP value (in logarithmic scale) and accumulated CPU time (in seconds) for all initialization strategies in a typical case.*

way constitutes the strategy *feasibility column*.
Compared to the basic column strategy, the feasibility column strategy progresses slightly different since (4.4d) initially is not satisfied with equality in the feasibility column case, whereas it typically is in the basic column case. Hence, the dual variables are likely to have different values and thus produce different columns to add. After the initial differences, both strategies perform similar.
A computational study on the performance influence by these initialization strategies for solving the linear relaxation is also reported in Koster and Zymolka [94]. Reusing the same test set as above, the linear relaxation was additionally solved in several ways, by alternative application of different initialization strategies and restricted pricing for a competing evaluation.
The results disclose two interesting observations. At first, the best solution strategy behaves very irregular on the test set. It turns out that this strategy is better compared to the others whenever the value of the best known solution used for initialization in fact equals the LP value, and much worse otherwise. This effect can be explained easily. In those cases that the LP value equals the best known solution value, no improving columns exist to decrease the value of the initial LP. This conclusion is drawn by the best solution strategy algorithm very quickly, for the test set confirming optimality after at most five iterations. The other strategies are comparable fast, but process more iterations for still having to generate the columns of an optimal solution. If in contrast the best solution value differs from the optimal LP value, starting with such a solution hampers the progress, due to being forced to improve it step-wise by compatible columns. In comparison, the other strategies can more simple and direct frame improved LP solutions and find the optimum much faster. For such a case, Figure 4.9 illustrates a typical behavior of the strategies during the column generation process, displaying the iterative progress of the LP value and the accumulating computation time.
Second, we realized that for the best solution strategy in particular those instances with a small (non-zero) gap between LP and best solution value are the most time consuming ones in comparison with the other strategies. This second effect is more difficult to explain, since it is in fact counter-intuitive. Experiments with less good solutions used as initial columns revealed that they often result in better CPU times than with the best solution. A possible explanation is that the progression can be more easily achieved if the solution is not that close to the LP value. Figure 4.9(b) confirms this, as the slope of the CPU time is far more steep for the best solution

strategy than for the other strategies, indicating that it is more difficult to integrate the new columns (the CPU time for pricing is typically marginal in comparison to the reoptimization of the master LP). First if the LP value starts to drop (compare with Figure 4.9(a)), the CPU time per column is comparable to the other strategies, but this usually happens much later.
So, against intuition, the best solution strategy is not favorable except for cases with suspicion to have an optimal solution already at hand. Otherwise, we experienced that the feasibility column strategy is most favorable on average, closely followed by the basic column strategy. These first evaluations have been carried out with the usual pricing. In combination with the always accelerating restricted pricing strategy, the best results are obtained by the basic column strategy, whereas the feasibility column strategy takes more time. This indicates that the restriction of path packings is most beneficial if achieving improvement by transition to another feasible solution requires generation of good fitting columns. In presence of a feasibility column, the effect looses strength.

Summary. As conclusion, the most profitable strategy combination is an initialization with the basic column and use of restricted pricing as long as it provides potential columns. Without restricted pricing, the strategies basic column and feasibility column perform comparable, usually following a similar progression curve after a different starting phase.
A further observation from the obtained results is that the path packing model has proven to be strong. The associated LP relaxation has often a non-zero optimum, which states a good lower bound and often directly confirms optimality of the best known solution generated heuristically. Hence, the formulation qualifies particularly as a suitable fundament for the development of an exact method. Before we turn to this, an alternative lower bound by a combinatorial approach is presented.

4.4.2.2 Combinatorial lower bound

Recently, Koster [91] suggested a combinatorial lower bound for Mcwap. Originally, the bound is derived for the special case of uniform fiber spectra and so resumed first, before we add the adaption to the general case and further observations.
In a clever way, Koster combines two former results and makes the conclusion applicable by excising small parts with characteristic topology from the network.

Excising stars. Consider an arbitrary Mcwap instance $(\mathcal{N}, P, \Lambda, K)$ with uniform fiber spectra. The network $\mathcal{N} = (N, L)$ can have any physical topology. However, focusing on a single node $n \in N$ and its incident links $L(n) \subset L$ (with the node's neighbors $N(n) \subset N$) forms a star subnetwork. We call such a partial network $\mathcal{N}_n := (\{n\} \cup N(n), L(n))$ an *excised star around* n. In a straightforward way, the Mcwap instance can be reduced to any such excised star. We replace each path by its subpath in $\mathcal{N}_n$ or remove it if it traverses no or just a single link in $L(n)$, obtaining the adapted path multi-set P_n of two-hop paths, and adapt K accordingly to $K_n : \Lambda \times L(n) \to \mathbb{Z}_+$ with $K_n(\lambda, \ell) := K(\lambda, \ell)$ for all $\lambda \in \Lambda$ and $\ell \in L(n)$. Let $(\mathcal{N}_n, P_n, \Lambda, K_n)$ denote such a reduced instance, referred to as *excised* Mcwap *star instance (around* n*)*.
In an excised Mcwap star instance, the center node n is the only place where

converters can be necessary. By the construction, a lower bound for such an excised Mcwap instance gives a lower bound on the number of converters required at n in the original instance, too. Since a node does not occur as center of multiple stars, lower bounds for excised Mcwap star instances around different nodes are independent of each other and thus can be accumulated to a total lower bound for the original instance. Besides this cumulation capability of the associated lower bounds, the investigation of star subnetworks is also profitable by the close relation of Mcwap to familiar coloring problems on such topologies.

Lower bound for excised stars. In Section 4.2, stars have been identified as topology with special properties in view of Mcwap. Figure 4.2 on page 146 indicates the close relation of Mcwap on star networks to instances with paths of hop-length at most two and the reversible transformation to edge coloring instances. Koster exploited the latter correspondence, capturing multi-fiber Mcwap with individual link fiber numbers by extension to k-edge coloring. This way, excised stars for any Mcwap instance can be handled. As second ingredient, he observed that a particular lower bound for k-edge coloring by Nakano et al. [125] can be adapted to provide an implication for the associated Mcwap star instances. This bound approximates the number of edges in a k-edge coloring instance that cannot be colored with a *given* number of colors. Because any edge represents a (part of a) lightpath by construction, uncolorable edges correspond to lightpaths that cannot be assigned a single wavelength and thus need a converter.

Let $(\mathcal{N}_n, P_n, \Lambda, K_n)$ be an arbitrary excised Mcwap star instance. Since the proper dimensioning condition (4.1) from page 137 carries over to excised subinstances, single-link paths in P_n can always be assigned a wavelength in a postprocessing and are left out in the following, assuming w.l.o.g. that $|L(p)| = 2$ for all paths $p \in P_n$. The corresponding k-edge coloring instance is then defined by the multi-graph $G_n = (V_{L(n)}, E_{P_n})$ where $V_{L(n)}$ contains a vertex v_ℓ for each network link $\ell \in L(n)$ and E_{P_n} holds an individual edge $v_{\ell_1} v_{\ell_2}$ for each path $p \in P_n$ with $L(p) = \{\ell_1, \ell_2\}$. The vertex weights k_{v_ℓ} are given by the associated link fiber numbers k_ℓ, w.l.o.g. with $k_\ell > 0$ for all $\ell \in L(n)$. Then, any k-edge coloring of G_n with at most $|\Lambda|$ wavelengths corresponds to a wavelength assignment for $(\mathcal{N}_n, P_n, \Lambda, K_n)$ without converters.

Let $W \subset V_{L(n)}$ be an arbitrary vertex subset, and define $k(W) := \sum_{v_\ell \in W} k_\ell$. For any k-edge coloring of G_n, the number of edges with the same color in the induced multi-subgraph $G_n[W] = (W, E_{P_n}[W])$ is at most $\lfloor \frac{1}{2} k(W) \rfloor$ (the maximum size of a so-called k-matching). Hence, any feasible k-edge coloring with (at most) $|\Lambda|$ colors leaves at least a number of

$$b_n(W) := \max \left\{ 0,\, |E_{P_n}[W]| \,-\, |\Lambda| \left\lfloor \frac{1}{2} k(W) \right\rfloor \right\} \tag{4.7}$$

edges uncolored. Koster argues that condition (4.1) yields $|E[W]| \leq \frac{1}{2} |\Lambda|\, k(W)$ and thus $b_n(W) = 0$ for any subset W with $k(W)$ even. The same result holds for $|W| \leq 2$. Therefore, these subsets can be omitted. Since (4.7) gives a valid lower bound on the number of uncolorable edges or, equivalently, the number of converters in the corresponding Mcwap subinstance restricted to the links associated with W for any remaining selection of W, it is possible to maximize over these subsets which

gives

$$B(G_n) := \max_{\substack{W \subset V_{L(n)}: \\ k(W)\ odd}} b_n(W) = \max \left\{ 0, \max_{\substack{W \subset V_{L(n)}: \\ k(W)\ odd}} \left(|E_{P_n}[W]| \ - \ |\Lambda| \ \left\lfloor \frac{1}{2} k(W) \right\rfloor \right) \right\}$$

as lower bound on the converter number for $(\mathcal{N}_n, P_n, \Lambda, K_n)$. Accumulating these bounds for each node $n \in N$ yields finally

$$B := \sum_{n \in N} B(G_n) \tag{4.8}$$

as lower bound for the original Mcwap instance $(\mathcal{N}, P, \Lambda, K)$.
Koster proved some interesting properties of the lower bound B defined by (4.8). On one hand, a comparison to the lower bound $\hat{z}_P^*$ obtained from the linear relaxation of (4.4) shows:

Theorem 4.20 (Koster [91]) *For any* Mcwap *instance with uniform fiber spectra on a star network, the lower bounds B and $\hat{z}_P^*$ for the minimum converter number are equal, i.e., $B = \hat{z}_P^*$.*
In the general case of Mcwap *with uniform fiber spectra on arbitrary networks, it holds that $\hat{z}_P^* \geq B$.*

So, B is theoretically weaker than $\hat{z}_P^*$, and Koster gives an example where in fact $\hat{z}_P^* > B$.
On the other hand, the combinatorial nature of B provides some advantages. As additional information, the values $B(G_n)$ allocate unavoidable converters to particular nodes in the network. Further research might investigate to exploit such non-zero bounds for individual nodes to improve guidance of solution algorithms. Moreover, B can be determined efficiently. Koster reformulated $B(G_n)$ by means of a submodular function. Minimizing such functions has been investigated by Grötschel et al. [57, 58] and yields:

Theorem 4.21 (Koster [91]) *For any* Mcwap *instance with uniform fiber spectra, B according to* (4.8) *can be computed in strongly polynomial time.*

Moreover, Schrijver [149] and Iwata et al. [73] gave combinatorial methods for minimizing submodular functions. For computational reasons, however, Koster proposed a simpler Greedy-type algorithm to approximate the values $B(G_n)$ from below by considering selected subcollections of subsets W and used this method for a bound comparison.

Non-uniform fiber spectra. We now adapt B to Mcwap with non-uniform fiber spectra. For this, reconsider the definition of $b_n(W)$ in (4.7). Note that in the previous derivation for uniform fiber spectra, the colors from Λ are considered individually, but all in parallel due to having the same total multiplicity $k(W)$ for any vertex subset W. Without the latter property, an analogous argumentation holds when using color-wise adapted entities.
We construct the edge-coloring multi-graph G_n as before, but deal now with individual vertex weights $k_{v_\ell}^\lambda := k_\ell^\lambda$ for each color $\lambda \in \Lambda$, i.e., restrict to edge colorings of G_n where the number of edges with color λ incident to a node n does not exceed k_ℓ^λ.

Each such (complete) coloring of G_n again corresponds to a wavelength assignment for $(\mathcal{N}_n, P_n, \Lambda, K_n)$ without converters. For an arbitrary vertex subset $W \subset V_{L(n)}$, we define $k_\lambda(W) := \sum_{v_\ell \in W} k_\ell^\lambda$ for all $\lambda \in \Lambda$. Then, no feasible edge coloring of G_n w.r.t. the vertex weights $k_{v_\ell}^\lambda$ can have more than $\left\lfloor \frac{1}{2} k_\lambda(W) \right\rfloor$ edges colored with λ in the induced multi-subgraph $G_n[W]$. This holds for any color $\lambda \in \Lambda$, and over the complete spectrum, the maximum number of colorable edges in $G_n[W]$ sums up to $\sum_{\lambda \in \Lambda} \left\lfloor \frac{1}{2} k_\lambda(W) \right\rfloor$. Hence, the estimation (4.7) for the least number of uncolorable edges in $G_n[W]$ is replaced by

$$b_n(W) := \max \left\{ 0,\, |E_{P_n}[W]| \,-\, \sum_{\lambda \in \Lambda} \left\lfloor \frac{1}{2} k_\lambda(W) \right\rfloor \right\}$$

and, if inserted further into the definition of $B(G_n)$, finally yields the desired combinatorial lower bound B for Mcwap with arbitrary fiber spectra.

Influence of terminating lightpaths. We add a further observation regarding the lightpaths terminating in the star center which have been neglected for the bound estimation. For the explanation, we restrict again to the special case of uniform fiber spectra, which carries over straightforward to the general case. Consider an arbitrary node $n \in N$ as center of an excised Mcwap star instance $(\mathcal{N}_n, P_n, \Lambda, K_n)$. Each path $p \in P_n$ in the adapted path multi-set has two hops and thus occupies a channel on two links from $L(n)$, and note that $|P_n| = |E_{P_n}|$. Moreover, let $\overline{P_n} := \{\, p \in P \mid n \in \{o_p, d_p\} \,\}$ be the multi-set of all lightpaths from the original instance (including single-link lightpaths) that terminate in node n. Each such path occupies a single channel on one of the links $L(n)$ incident to n. For reduction to an arbitrary vertex subset $W \subset V_{L(n)}$ of G_n, let further $\overline{P_n}[W] := \left\{\, p \in \overline{P_n} \;\middle|\; L(p) = \{\ell\} \,:\, v_\ell \in W \,\right\}$ define those paths from $\overline{P_n}$ using a link with associated vertex v_ℓ in W. Then the policy of sufficiently dimensioned links, as formulated in (4.1) on page 137, can be summed up over all links corresponding to vertices $v_\ell \in W$ and guarantees that

$$2\,|E_{P_n}[W]| \,+\, |\overline{P_n}[W]| \,\leq\, |\Lambda| \sum_{v_\ell \in W} k_\ell \,=\, |\Lambda|\, k(W)\;, \tag{4.9}$$

i.e., the restriction of all lightpaths from the original solution to the links corresponding to vertices in W does not occupy more channels in total than provided by the sum of installed link capacities. Inequality (4.9), appropriately transformed, can now be inserted into the combinatorial estimation (4.7) of unavoidable converters at node n and yields for the only relevant case that $k(W)$ is odd:

$$\begin{aligned} b_n(W) &= \max \left\{ 0,\, |E_{P_n}[W]| \,-\, |\Lambda| \left\lfloor \frac{1}{2} k(W) \right\rfloor \right\} \\ &\leq \max \left\{ 0,\, |\Lambda| \frac{1}{2} k(W) - |\Lambda| \frac{1}{2} \left(k(W) - 1\right) - \frac{1}{2} |\overline{P_n}[W]| \right\} \\ &\leq \max \left\{ 0, \frac{1}{2} \left(|\Lambda| \,-\, |\overline{P_n}[W]| \right) \right\} \end{aligned}$$

Hence, we can deduce that $b_n(W) = 0$ whenever the number of lightpaths terminating in n along the links associated to W is at least as high as the spectrum size $|\Lambda|$.

In other words, a combinatorial indication of unavoidable converters at a node n is only possible if the remaining lightpaths traversing n altogether occupy a number of channels on the involved links which is close to the totally provided capacity. Without need of conversion, lightpaths terminating in n push the amount of installed capacities, thus have a contra-productive influence on the lower bound estimation.

Evaluation. Koster [91] carried out a computational study on the quality of B, reusing the test set of the linear relaxation evaluations. This evaluation restricts to uniform fiber spectra cases. From the initial 80 instances, no converter-free solution was found by heuristics in 23 cases, and 13 of these have shown a zero lower bound $\hat{z}_P^* = 0$ which yields $B = 0$ as well by Theorem 4.20. For the remaining 10 instances, the proposed Greedy heuristic was applied to compute a lower bound (on B, in fact, and thus indirectly on the Mcwap instance as well). These computations terminated after less than a millisecond each and revealed as result that the obtained values were equal to $\hat{z}_P^*$ without exemption.
So, the proposed combinatorial bounding method proves comparable throughout the practical test instances of this evaluation. Clearly, this cannot be expected in general, as already indicated by Koster's counter-example. Therefore, using the linear relaxation value $\hat{z}_P^*$ of formulation (4.4) as lower bound is anyway advisable, too. As already suggested, model (4.4) suits also for development of an exact approach, discussed next.

4.4.3 An exact approach

A standard exact solution approach for integer linear programs is provided by the well known branch-and-bound method (see, for instance, Nemhauser and Wolsey [127]). In a divide-and-conquer manner, the scheme explores a dynamically generated search tree of partial problems, which are successively subdivided whenever the associated linear relaxation does not show an integer optimum solution. Thereby, a bounding methodology is used to identify search tree areas whose investigation becomes obsolete during the process. For this, the optimal value of the linear relaxation of a subproblem typically serves as dual bound, whereas the value of the best found integer solution provides the primal bound to compare with. In combination with a pricing procedure for solving the linear programs by column generation, the resulting method is finally called branch-and-price.
In this section, we complete the necessary ingredients to define an exact branch-and-price method for solving the path packing formulation (4.4) as integer linear program. The model and the applied column generation procedure have already been presented. As last essential part, we discuss applicable branching rules to deal with fractional solutions of the generated linear programs. For our Mcwap model, this aspect turns out to be challenging due to the influence of master program modifications on the pricing step.

Branching rules. In any branch-and-bound based approach for solving integer linear programs, the provision of well-designed branching rules is of crucial importance for the solution process. In the inspiring approach to vertex coloring, Mehrotra and Trick [115] propose a combinatorial branching by the simple distinction whether two non-adjacent vertices get the same color or not. This elegant idea provides the

advantage that branching just changes the underlying graph, but does not require to add branching constraints to the integer linear program and thus does not affect the column generation procedure at all. Unfortunately, this way of branching does not carry over to Mcwap where wavelengths can be available multiple times and thus conflicts do not restrict to pairs of lightpaths.
General purpose ILP solvers typically apply generic branching schemes, such as branching on a single integer variable with non-integral value or branching on a weighted integer variable sum with actual fractional sum value (also known as knapsack branching). Furthermore, each of these schemes comes along with a specified rule to break ties, for instance by maximum fractionality or by random choice. For a particular branching step, the applied rule dictates which additional constraints have to be inserted into the actual linear program to cut off a current non-integral LP solution and to specify the subproblems to consider further. As long as these subproblems are solvable by standard LP solvers, any branching rule can be applied. In our case, however, column generation is used to solve the LP due to its exponential number of variables. In combination with column generation, suitable branching rules for the master problem have to be selected more careful since the induced constraints added to an LP for branching can also influence the pricing problem (and its solvability). A further issue is the completeness of a branching scheme, i.e., whether one of the involved rules applies to each feasible fractional solution or not. In the following, we propose a couple of branching rules and discuss their properties in regard of these aspects.

Branching on individual variables. Branching on individual integer variables with non-integral value is one of the most prominent types of branching rules. Such a branching possibility is simple to detect and simple to realize, as usually only variable bounds must be adapted.
The path packing formulation (4.4) contains two types of integer variables, the y-variables selecting the partial paths to cover each lightpath, and the x-variables specifying the path packings to provide these selected partial paths (and, implicitly, the wavelengths assigned to them). For LP solving, the x-variables are priced out by column generation, and so branching on these variables has undesired implications on the pricing problem. In fact, assume the current (subproblem) LP solution contains a variable with fractional value $x^*_{\hat{\phi}} \notin \mathbb{Z}$ for a path packing $\hat{\phi} \in \hat{\Phi}$. A branching on this variable generates two subproblems, one by adding the constraint $x_{\hat{\phi}} \leq \lfloor x^*_{\hat{\phi}} \rfloor$, and the other by adding the constraint $x_{\hat{\phi}} \geq \lceil x^*_{\hat{\phi}} \rceil$. Each of these constraints generates a further dual variable whose value must be appropriately taken into account within the corresponding pricing when solving the respective master subproblem. (Otherwise, pricing in the branch with $x_{\hat{\phi}} \leq \lfloor x^*_{\hat{\phi}} \rfloor$ could generate exactly the same column as for $x_{\hat{\phi}}$, and the associated variable with value $x^*_{\hat{\phi}} - \lfloor x^*_{\hat{\phi}} \rfloor$ would reproduce an equivalent LP solution as that intended to be cut off—the begin of a never ending story without happy end.) Unfortunately, detection of whether such a dual value contributes to the objective of the pricing problem or not requires to identify whether the actual pricing problem solution equals exactly the multiplicity function of $\hat{\phi}$ or not, a rather extensive task as will be discussed below in more detail. Anticipating these results, we propose not to branch on individual x-variables in order to avoid the exacerbating complication of the pricing.

In contrast, all y-variables are explicitly included in the master program, and one easily observes that branching on such a variable does not influence the pricing. Hence, we propose as first branching rule:

Branching rule 1: Branching on y-variables

Given a feasible fractional solution (y^*, x^*) of the linear relaxation of (4.4) such that there exist $p \in P_1$, $s \in S_p$ with value $y_p^{*s} \notin \mathbb{Z}$, branch by dividing the current problem into two subproblems defined by adding

$y_p^s \leq \lfloor y_p^{*s} \rfloor$ in one branch, and

$y_p^s \geq \lceil y_p^{*s} \rceil$ in the other branch.

For breaking ties, select an arbitrary candidate variable with most fractional value, i.e., branch on a variable whose value minimizes $|y_p^{*s} - \lfloor y_p^{*s} \rfloor - 0.5|$.

Clearly, no feasible integer solution of (4.4), but the actual fractional LP solution is cut off by this standard branching rule. Since the branched variables occur in the objective, such branchings have often an impact on the optimal subproblem LP value in (at least) one of the branches. For instance, any branching on a variable y_p^s with $s = p$ directly implies usage of more respectively less conversions for lightpaths along route p in the subproblems. In case no other lightpaths need conversion, the LP optimum in the branch with $y_p^s \leq \lfloor y_p^{*s} \rfloor$ will strictly increase and, in the best case, might thereby become prunable.
Unfortunately, Branching rule 1 is not complete since there exist feasible fractional LP solutions with all y-variables integer, e.g., whenever the LP value $\hat{z}_P^* = 0$ in objective (4.4a) (where evaluations always include the term in brackets), all y-variables have integer values according to the only feasible setting $y_p^{*s} = v_p$ if $s = p$ and $y_p^{*s} = 0$ otherwise, independent of whether the LP solution is completed by fractional or integer x-variables. As a consequence, exclusive application of Branching rule 1 does not suffice to provide an exact solution method for Mcwap, and further branching rules involving the x-variables are inevitable.

Branching rules involving x-variables. Whenever a variable $x_{\hat{\phi}}$ is involved in a constraint added to the master program for branching purposes, the pricing has to take the corresponding dual variable value of this constraint into account. Basically, such a dual variable becomes part of the optimality condition (4.5) for $\hat{\phi}$, being added to the left hand side with the appropriate constraint coefficient as weight. Hence, the weighted dual variable value becomes part of the pricing problem objective if and only if the solution of (4.6) equals the multiplicity function of $\hat{\phi}$. This implies that the pricing problem has to be adapted in order to identify those solutions that correspond to path packings contained in a master program branching constraint.
An exact identification of an individual path packing $\hat{\phi}$ is cumbersome since it is to ensure that $m^s = m^s_{\hat{\phi}}$ holds for all $s \in S$. This can be linearly modeled by use of binary indicator variables and appropriate constraints for their correct setting as shown below, but introduces a number of additional variables and constraints in the order of $\mathcal{O}(\,|S|\,)$ to the pricing model. Since to be added for each individual path packing to identify, these substantial extensions of (4.6) yield an exploding pricing

problem which soon becomes too complex for an efficient (since often repeated) solution and thus impracticable.
A more promising alternative is to apply knapsack branching by use of a particularly designed partition of $\hat{\Phi}$ into categories whose members can be identified more efficiently for pricing. For this, branching is realized by adding a master program constraint that bounds an (actually fractional) sum over all x-variables for path packings that belong to the selected category, including those that have not yet been priced out, to the next integral values. A simple case for such a partition of $\hat{\Phi}$ is to consider the subsets $\hat{\Phi}^s_\beta := \left\{ \hat{\phi} \in \hat{\Phi} \;\middle|\; m^s_{\hat{\phi}} \geq \beta \right\}$ and $\hat{\Phi} \setminus \hat{\Phi}^s_\beta$, i.e., to distinguish the path packings according to whether subpath s is contained at least β times or not. Provided the actual LP solution has a fractional sum value over the x-variables for all $\hat{\phi} \in \hat{\Phi}^s_\beta \cap \overline{\Phi}$, the following branching rule can be applied:

Branching rule 2: Branching on a single subpath multiplicity

Given a feasible fractional solution (y^*, x^*) of the linear relaxation of (4.4) such that there exist $s \in S$ and $\beta \in \mathbb{Z}_+$ with $\sum_{\hat{\phi} \in \overline{\Phi} \cap \hat{\Phi}^s_\beta} x^*_{\hat{\phi}} =: \alpha \notin \mathbb{Z}$,
branch by dividing the current problem into two subproblems defined by adding

$$\sum_{\hat{\phi} \in \hat{\Phi}^s_\beta} x_\phi \leq \lfloor \alpha \rfloor \quad \text{in one branch, and} \tag{4.10}$$

$$\sum_{\hat{\phi} \in \hat{\Phi}^s_\beta} x_\phi \geq \lceil \alpha \rceil \quad \text{in the other branch.} \tag{4.11}$$

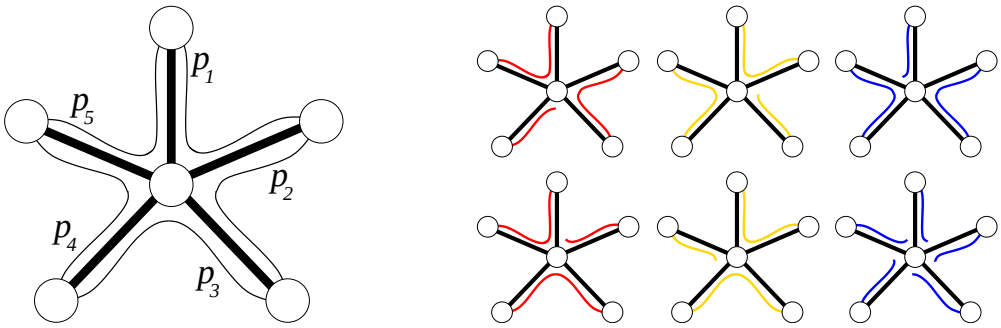
For breaking ties, select an arbitrary candidate set $\hat{\Phi}^s_\beta$ with most fractional value α, i.e., select $s \in S$ and β such that the corresponding sum value α minimizes $|\alpha - \lfloor \alpha \rfloor - 0.5|$.

Again, no feasible integer solution of the path packing formulation is discarded by such a branching, whereas the current fractional LP relaxation solution is cut off. Regarding the pricing of further columns, the following adaption has to be done. Let $\pi_{\hat{\Phi}^s_\beta}$ be the dual variable corresponding to the added constraint (4.10) resp. (4.11). In the objective (4.6a) of the pricing problem, the value of this variable, with coefficient 1 when representing (4.10) and coefficient -1 otherwise, must be added if and only if the solution of (4.6) corresponds to the multiplicity function of a path packing $\hat{\phi} \in \hat{\Phi}^s_\beta$. For characterizing this situation, we introduce the indicator variable $t_{\hat{\Phi}^s_\beta} \in \{0, 1\}$ to express whether $\hat{\phi}$ with $m^s_{\hat{\phi}} := m^s$ in (4.6) belongs to $\hat{\Phi}^s_\beta$ (by $t_{\hat{\Phi}^s_\beta} = 1$) or not. Then the objective (4.6a) is extended by the additional term $\pm \pi_{\hat{\Phi}^s_\beta} t_{\hat{\Phi}^s_\beta}$, and the correct setting of $t_{\hat{\Phi}^s_\beta}$ is ensured by adding the following conditions to the pricing model (4.6):

$$\beta \, t_{\hat{\Phi}^s_\beta} \leq m^s \tag{4.12a}$$

$$1 + (m^s - \beta) \leq (b^s + 1) \, t_{\hat{\Phi}^s_\beta} \tag{4.12b}$$

$$t_{\hat{\Phi}^s_\beta} \in \{0, 1\} \tag{4.12c}$$



(a) *5-star* MCWAP *instance.* (b) *The path packings used in the solution.*

Figure 4.10: *Example for incompleteness of Branching rule* 2. *For the single-fiber* MCWAP *instance (a) with uniform form fiber spectra of size* $|\Lambda| = 3$ *and five lightpaths* $p_1, \ldots, p_5$*, a fractional solution of* (4.4) *is given by use of the path packings depicted in (b) each with variable value* $\frac{1}{2}$ *and all others being 0, while setting* $y_p^s = 1$ *whenever* $s = p$ *and* $y_p^s = 0$ *otherwise. This feasible solution yields* $\alpha \in \mathbb{Z}$ *for any choice of* $s \in S$ and $\beta \in \mathbb{Z}_+$ *and thus does not allow to branch by Rule* 2 *(neither by Rule* 1*, too).*

Inequality (4.12a) is redundant for $t_{\hat{\Phi}_\beta^s} = 0$, but in case $t_{\hat{\Phi}_\beta^s} = 1$ implies that in fact $m^s \geq \beta$ must hold. In the other direction, constraint (4.12b) guarantees that $t_{\hat{\Phi}_\beta^s}$ is set to 1 whenever $m^s \geq \beta$, and otherwise reduces to a redundant non-negativity condition, too. Thereby, the maximum utilization multiplicity b^s of subpath s, already applied in the bounds (4.6d), used for the right-hand side coefficient in (4.12b) obviates $t_{\hat{\Phi}_\beta^s}$ to take values larger than 1. Condition (4.12c) restricts $t_{\hat{\Phi}_\beta^s}$ to take binary values only and thus to serve as an indicator. In combination, this model extension satisfies $t_{\hat{\Phi}_\beta^s} = 1 \;\Leftrightarrow\; m^s \geq \beta$ and thus, as intended, integrates the dual variable value $\pi_{\hat{\Phi}_\beta^s}$ correctly into the pricing problem. For any further generated column, the value $\pm t_{\hat{\Phi}_\beta^s}$, depending on which branch is considered, delivers also the coefficient of the column in the corresponding branching constraint (4.10) respectively (4.11).
As a consequence, application of Branching rule 2 for the master problem generates two subproblems whose corresponding pricing models grow by a binary variable and two inequalities, increasing the complexity for generating further columns just moderately. Clearly, iterative branchings by this rule yield iterative pricing model extensions, too, and solving these models therefore becomes more and more complex. Moreover, the rule does not provide a complete branching scheme, as proven demonstratively by the example in Figure 4.10.
Branching rule 2 can be modified and extended in various ways. As a simple example, branching on the selection of all path packings $\hat{\phi}$ with the property $m_{\hat{\phi}}^s \leq \beta$ for a specified subpath s can be realized similarly since the corresponding path packing subset equals $\hat{\Phi} \setminus \hat{\Phi}_{\beta+1}^s$. So, one has just to replace β by $\beta + 1$ and $t_{\hat{\Phi}_\beta^s}$ by $(1 - t_{\hat{\Phi}_\beta^s})$ in the pricing model extension to capture this case.
A more interesting variation is to bound a (weighted) sum of subpath multiplicities by a constant value for characterization of a path packing category. We restrict the explanation to cases with uniform weights. For such a case, a subpath subset $S' \subset S$ is specified together with a multiplicity bound $\beta \in \mathbb{Z}_+$, and we define the

path packing category

$$\hat{\Phi}_\beta^{S'} := \left\{ \hat{\phi} \in \hat{\Phi} \;\middle|\; \sum_{s \in S'} m_{\hat{\phi}}^s \geq \beta \right\} .$$

Clearly, each category with $|S'| = 1$ equals that defined by $\hat{\Phi}_\beta^s$ with $S' = \{s\}$, and hence the new categorization scheme generalizes the previous one. Moreover, we obtain a further branching rule:

Branching rule 3: Branching on a subpath multiplicity sum

Given a feasible fractional solution (y^*, x^*) of the linear relaxation of (4.4) such that there exist $S' \subset S$ and $\beta \in \mathbb{Z}_+$ with $\sum_{\hat{\phi} \in \overline{\Phi} \cap \hat{\Phi}_\beta^{S'}} x_{\hat{\phi}}^* =: \alpha \notin \mathbb{Z}$,
branch by dividing the current problem into two subproblems defined by adding

$$\sum_{\hat{\phi} \in \hat{\Phi}_\beta^{S'}} x_\phi \leq \lfloor \alpha \rfloor \quad \text{in one branch, and} \tag{4.13}$$

$$\sum_{\hat{\phi} \in \hat{\Phi}_\beta^{S'}} x_\phi \geq \lceil \alpha \rceil \quad \text{in the other branch.} \tag{4.14}$$

For breaking ties, select an arbitrary candidate set $\hat{\Phi}_\beta^s$ with most fractional value α, i.e., select S' and β such that the corresponding sum value α minimizes $|\alpha - \lfloor \alpha \rfloor - 0.5|$.

As before, both constraints are violated by the actual relaxation solution, whereas each feasible integer solution satisfies exactly one of them. Furthermore, using $\sum_{s \in S'} m_{\hat{\phi}}^s \leq \beta$ as category characterizing property results in the subset $\hat{\Phi} \setminus \hat{\Phi}_{\beta+1}^{S'}$ of all path packings and can thus also be realized by a corresponding adaption of Branching rule 3, as described above for the single subpath branching case.
In view of the pricing problem, a similar extension as for Branching rule 2 applies. Now we use the indicator variable $t_{\hat{\Phi}_\beta^{S'}} \in \{0, 1\}$ which replaces $t_{\hat{\Phi}_\beta^s}$ in all terms including the objective adaption, and we further add constraints (4.12) where just m^s is substituted by $\sum_{s \in S'} m^s$, and b^s by $\sum_{s \in S'} b^s$. Hence, branching by Rule 3 extends the pricing model for the subproblems only by a single binary variable and two inequalities, too. Regarding these inequalities, it is probable that those containing more m^s variables yield stronger constraints. Therefore, a further (or alternative) criterion for breaking ties in the branching rule is to choose a largest set S' yielding a non-integral value α.
Unfortunately, it is open whether Branching rule 3 defines a complete branching scheme, especially for the uniformly weighted case. In the example of Figure 4.10, this rule allows to branch for instance by using the property $m^{p_1} + m^{p_3} \geq 2$, i.e., $S' = \{p_1, p_3\}$ and $\beta = 2$, which yields $\alpha = \frac{1}{2}$. Moreover, computational experiments on realistic instances have shown that such a branching could always be found. Nevertheless, as long as an existence of special instances and solutions which render application of Branching rule 3 impossible is not disproved, completeness of the branching rules presented so far cannot be claimed. Therefore, we present a final rule which provides a complete branching scheme.

Complete branching rule. Both category definitions used above, for $\hat{\Phi}^{s}_{\beta}$ and $\hat{\Phi}^{S'}_{\beta}$, use a single condition to specify the contained path packings. Now, we consider the more general case where multiple conditions are allowed. Let $r \in \mathbb{N}$ and consider mutually different subpaths $s_1, \ldots, s_r \in S$ with associated multiplicity bounds $\beta_1, \ldots, \beta_r \in \mathbb{Z}_+$. Then we define the corresponding path packing category

$$\hat{\Phi}^{s_1,\ldots,s_r}_{\beta_1,\ldots,\beta_r} := \left\{ \hat{\phi} \in \hat{\Phi} \;\middle|\; m^{s_1}_{\hat{\phi}} \geq \beta_1, \ldots, m^{s_r}_{\hat{\phi}} \geq \beta_r \right\} \tag{4.15}$$

and its counterpart as $\hat{\Phi} \setminus \hat{\Phi}^{s_1,\ldots,s_r}_{\beta_1,\ldots,\beta_r}$. The case $r = 1$ corresponds again to the category definition for Branching rule 2, and so we now obtain another generalization by the following rule:

Branching rule 4: Branching on multiple subpath multiplicities
Given a feasible solution (y^*, x^*) of the linear relaxation of (4.4) with non-integral part x^*, determine $r \in \mathbb{N}$, $s_1, \ldots, s_r \in S$, and $\beta_1, \ldots, \beta_r \in \mathbb{Z}_+$ defining $\hat{\Phi}^{s_1,\ldots,s_r}_{\beta_1,\ldots,\beta_r} =: \hat{\Phi}'$ for which the current variable sum $\sum_{\hat{\phi} \in \overline{\Phi} \cap \hat{\Phi}'} x^*_{\hat{\phi}} =: \alpha \notin \mathbb{Z}$ is non-integral, and
branch by dividing the current problem into two subproblems defined by adding

$$\sum_{\hat{\phi} \in \hat{\phi}'} x_\phi \leq \lfloor \alpha \rfloor \quad \text{in one branch, and} \tag{4.16}$$

$$\sum_{\hat{\phi} \in \hat{\phi}'} x_\phi \geq \lceil \alpha \rceil \quad \text{in the other branch.} \tag{4.17}$$

For breaking ties, select a candidate set $\hat{\Phi}^{s_1,\ldots,s_r}_{\beta_1,\ldots,\beta_r}$ with minimum number r of defining conditions and, among those, an arbitrary one that yields a most fractional value α.

As already expressed in the description, this rule can be applied to cut off any feasible solution of the linear relaxation of the path packing formulation (4.4) with non-integral x-variables, which is formally proven by the following proposition:

Proposition 4.22 *For any feasible solution (y^*, x^*) of the linear relaxation of (4.4) that contains non-integral variables x^*, there exist $r \in \mathbb{N}$, $s_1, \ldots, s_r \in S$, and $\beta_1, \ldots, \beta_r \in \mathbb{Z}_+$ defining $\hat{\Phi}' := \hat{\Phi}^{s_1,\ldots,s_r}_{\beta_1,\ldots,\beta_r}$ by (4.15) such that*

$$\sum_{\hat{\phi} \in \overline{\Phi} \cap \hat{\Phi}'} x^*_{\hat{\phi}} \notin \mathbb{Z} \,.$$

Proof. Consider an arbitrary variable $x_{\hat{\phi}}$ with value $x^*_{\hat{\phi}} \notin \mathbb{Z}$ which exists by prerequisite. The corresponding path packing $\hat{\phi} \in \overline{\Phi}$ is fixed in the following. Then we set $r := |S|$ for involving all subpaths $\{s_1, \ldots, s_r\} = S$ indexed arbitrary, but fixed, and further define $\beta_i := m^{s_i}_{\hat{\phi}}$ for all $i = 1, \ldots, r$.
Next, consider the associated subset $\hat{\Phi}' := \hat{\Phi}^{s_1,\ldots,s_r}_{\beta_1,\ldots,\beta_r}$ as defined in (4.15). For any $\phi \in \hat{\Phi}'$, we get $\hat{\phi} \subset \phi$ due to $S = \{s_1, \ldots, s_r\}$, and since $\hat{\phi}$ is inclusion-maximal by prerequisite, $\phi = \hat{\phi}$ follows. As a consequence, $\hat{\Phi}' = \{\hat{\phi}\}$, and the initial selection

yields

$$\sum_{\phi \in \overline{\Phi} \cap \hat{\Phi}'} x^*_\phi = x^*_{\hat{\phi}} \notin \mathbb{Z}$$

which proves the claim. ■

The proof shows that Branching rule 4 can in fact be applied to branch on an individual $x_{\hat{\phi}}$-variable with non-integral value. As indicated before, this requires (in the worst case) to use $r = |S|$ conditions in the category definition which yields a substantial expansion of the pricing model.
For associated adaption of the pricing problem, any branching according to Rule 4 needs for each of the r conditions to extend model (4.6) by a separate, individually parameterized constraint set (4.12) with the contained indicator variable expressing whether this condition is satisfied or not. Detecting whether all conditions are fulfilled concertedly needs a further indicator variable $t_{\hat{\Phi}'} \in \{0, 1\}$ set correctly by further adding the conditions

$$r\, t_{\hat{\Phi}'} \leq \sum_{i=1}^{r} t_{\hat{\Phi}^{s_i}_{\beta_i}}$$

$$\sum_{i=1}^{r} t_{\hat{\Phi}^{s_i}_{\beta_i}} \leq t_{\hat{\Phi}'} + r - 1$$

$$t_{\hat{\phi}'} \in \{0, 1\}$$

to the model. Finally, $\pm\pi_{\hat{\Phi}'} t_{\hat{\Phi}'}$ is inserted into the objective (4.6a) where $\pm\pi_{\hat{\Phi}'}$ refers to the dual variable value of the added master program branching constraint (4.16) respectively (4.17). Hence, each application of Branching rule 4 extends the pricing problem by $r + 1$ binary variables and $2(r + 1)$ inequalities. For minimizing this extent, the tie breaking selects at first a minimum value for r. In addition, some condition extensions might have been already inserted by earlier branching steps and can then be reused. In general, however, the rule can cause a considerable complication for pricing and should therefore be used only in case no other presented branching rule applies.

Branch-and-price method. Proposition 4.22 shows that Branching rule 4 in combination with Branching rule 1 provides a complete branching scheme for the path packing formulation (4.4). Though the other branching rules need not be used necessarily, they form preferable alternatives due to imposing least additional effort for pricing in comparison to Branching rule 4 in general.
Altogether, the presented set of branching rules completes the definition of a branch-and-price method for solving the path packing formulation (4.4) of Mcwap as integer linear program. For any further algorithm design issue such as search tree node selection, standard mechanisms can be applied. Theoretically, such a branch-and-price algorithm is an exact solution method with finite running time for any finite Mcwap instance. For computational experiments, we have also implemented the method, using depth-first-search with backtracking as search tree exploration scheme in order to reduce overhead for setting up the actual subproblem LP and the corresponding pricing problems. As expected, our computational studies revealed that the basic branch-and-price algorithm allows to optimize instances of limited

size only, for a rough figure at most having around five wavelengths and up to several hundreds of lightpaths if the number of different routes reduces this number by at least a factor of two to three. Practical instances from real-world networks are typically much larger, in particular with respect to the number of wavelengths, and thus further research on algorithmic improvements is needed to make the exact branch-and-price approach applicable to such instances, too. Nevertheless, the main strength of our path packing formulation consists of providing a strong linear relaxation lower bound which often proves optimality of a heuristically generated wavelength assignment involving converters, as our computational result presented in the next chapter confirm. Here, we close the discussion of the exact approach by briefly reporting on preliminary computational experiences which might indicate helpful insights for the development of further enhancements.

Preliminary computational results. In computational studies, we tested the branch-and-price algorithm (and several variations) on instances of varying size and structure. Regarding the branching rules, the most promising order turned out to be as naturally expected, using Branching rule 1 whenever possible due to having no influence on the generation of further columns, followed by the rules 2 and 3. In fact, Branching rule 4 was never needed since one of the other rules always applied (as long as the algorithm was run).
A helpful feature was to make use of a feasibility column as described in Section 4.4.2.1 for the initialization of the master program linear relaxation. Such a feasibility column, which is not included in any branching constraint as well, provides the advantage to guarantee for feasibility of the master program as long as successively added branching constraints do not form an infeasible subsystem. Note that adding a branching constraint to an optimized LP during the branch-and-price method makes the resulting master LP subproblem often infeasible as long as no further columns are generated. In such a situation, pricing would have to use advanced dual information which is more difficult to obtain. This is circumvented by a feasibility column which turns infeasibility by missing columns into large objective penalties and, in case the actual LP is nonetheless infeasible, implies that this search tree node can be pruned since adding any further columns would not turn the program feasible.
Regarding the computational progress of the method, a major observation is that optimal linear relaxation solutions tend to contain a large set of x-variables with (small) positive values. A closer look on such sets shows that sometimes variable subsets can be identified which sum up to an integer value for a yet not priced out path packing, i.e., form a 'dispersed' representation of the required column. We also observed a particular situation where such a column can be described as linear combination of other columns. As long as all these other path packings are feasible solutions of the pricing model, such a combined column will not be priced out, since it is no extreme point of the pricing problem polytope. In this case, only further branchings imposing pricing model extensions cutting off combinable path packings can make the desired column pricable. Such time-consuming subprocesses might be avoidable by efficient methods for identification of those x-variable subsets, guiding an accompanying generation of columns that are encoded in such a dispersed way.
Moreover, a special variant of Branching rule 3 with $S' = S$ (called full sum branching) can provide an interesting variation. Since all subpaths occur in the sum, the

bound β in fact limits the cardinality of the represented path packing. Note that such cardinalities are limited to at most $\sum_{\ell \in L} k_\ell$ and at least a minimum value that can be computed by solving (4.6) for objective $\min \sum_{s \in S} m^s$. We often observed an increase in the subproblem LP optima when applying full sum branchings. As a potential approach, initial insertion of some corresponding branching constraints generates a subproblem with a solution space restricted to path packing selections that fit into a certain cardinality pattern. Such a much smaller problem might be better solvable, and a clever strategy to determine suitable patterns could be helpful to fasten the search for good primal solutions.
Another direction for enhancements is to derive cutting planes which can be separated during the computation, extending the method to a so-called branch-and-cut-and-price algorithm. A simple class of cutting planes is obtained by use of the maximum utilization number of subpaths for a solution. Each path packing $\hat{\phi} \in \hat{\Phi}^s_\beta$ provides at least β times the subpath s, and thus more than $\lceil b^s/\beta \rceil$ path packings of this type will not occur in any solution. In particular with $b^s/\beta \notin \mathbb{Z}$, a non-trivial feasible inequality

$$\sum_{\hat{\phi} \in \hat{\Phi}^s_\beta} x_{\hat{\phi}} \leq \left\lceil \frac{b^s}{\beta} \right\rceil$$

is obtained for each subpath $s \in S$ and each value $0 < \beta \leq b^s$ (where some values can imply the same inequality, of course). Though these cuts, even if all were added initially in the root node, did not brought observable improvements, detection of further (and stronger) cuts might be more successful in this regard.
Finally, the problem and model formulations can be varied, too. We already applied such a modification when changing from equalities to inequalities in constraints (4.4c) (with the opposite change in constraints (4.6b)). Additionally, one could turn the spectrum bound constraint (4.4d) to an equality since overprovision of subpaths does not matter. Another idea is to release the restriction to inclusion-maximal path packings while constraints (4.4c) remain inequalities. This allows to reset pricing constraints (4.6b) to inequalities, possible making the pricing model easier to solve. Another idea is to fill up the instance with single-link lightpaths such that all wavelengths on each link must in fact be assigned. This way, inequalities (4.4c) become implicit equalities.
This report documents some of the ideas and variations we experimented with. These studies have shown that progress in solution efficiency for Mcwap is quite hard to gain, and additional ideas will be needed until large-scale instances become solvable this way. The discussed insights could be useful to guide further research in this direction.

Chapter 5

Computing survivable optical network designs

In this chapter, we report on a comprehensive computational study to evaluate the developed methods and ideas on realistic instances. The emphasis of this study is twofold. On the one hand, the results document the performance of the derived solving methods and the quality of the computed designs. On the other hand, the generation of solutions for alternative settings does not only confirm the flexibility of our approach, but also allows to examine and compare several options for optical networks and their design, ranging from conceptional to technological prerequisites—as needed for a substantiated decision support in practice.

We begin with a description of the instances and settings for the study. We introduce the set of topologies used, the corresponding traffic demands and survivability requirements, as well as the considered hardware equipment. Moreover, we present some characteristics of the resulting network design instances.
As first part of the study, we investigate optical network designs under varying settings. In comparison to a set of reference solutions, corresponding designs are computed for the evaluation of alternative survivability concepts, an extended hardware model, the opaque network scenario, and the special case of upgrade planning. These case studies continue and extend previously performed studies on optical network design in Zymolka et al. [178] and on survivability models in Gruber et al. [62], Koster et al. [97], and Bley et al. [24].
In the second part, we take a closer look on the wavelength assignment subtask on a widely expanded set of design solutions. We generate feasible wavelength assignments heuristically and compute lower bounds for the number of converters.
A detailed register of the relevant outcome data of all computations can be found in Appendix B. In the following discussions, we focus on aggregated presentations of the results for deducing and approving interesting observations.

5.1 Instances and computational environment

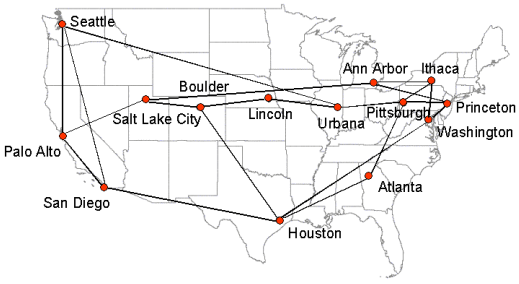
Practically oriented optical network design incorporates many details, alternatives, and options. The goal to incorporate as many of these issues as possible has motivated the development of highly flexible models and solution methods, as described in the previous chapters. For a significant computational evaluation of the methods and ideas, however, it is advisable to focus on the main aspects of interest and to minimize undesirable influence of minor details as much as possible. To this end, we avoid to use a confusing variety of different hardware devices, traffic matrices, or other parameter specifications and rather focus on a plain outfit of the test set instances.

Basically, the computational study is organized as follows. We use a set of five network topologies each with a prespecified static traffic demand and four different degrees of survivability to realize. This defines a total of 20 basic instances which are used for reference and comparisons. For these reference instances, we use a simple hardware model, no preinstalled equipment (i.e., greenfield planning), the transparent multi-hop case as technological scenario, and the general concept of Demand-wise Shared Protection (DSP) as survivability model. Variations of the latter fixings are individually explored in case studies, as discussed in the next section.

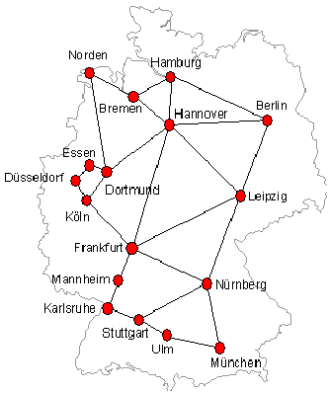
Two of the instances are supplied by our industrial cooperation partners, namely the large Germany network (pt50) provided by T-Systems [157] and the Austria network (pt20) from Telekom Austria [158]. The other three networks, the USA network (pt14) based on the famous NSF NET topology, the small Germany network (pt17), and the Europe network (pt28), have been published as realistic reference scenarios within the project MultiTeraNet (cf. Betker et al. [17] and the project website [124]) under participation of T-Systems and other partners from industry and academia. These reference networks and topologies have been used in other studies in the literature as well. Moreover, the considered topologies and traffic matrices can be found in the instance library `SNDlib` [152, 167].

In the following, we give an overview on the basic instance constituents, their characteristics, and the general settings for the design task and the computational environment. Thereby, parameter namings correspond to those used in the definitions in Chapter 2.

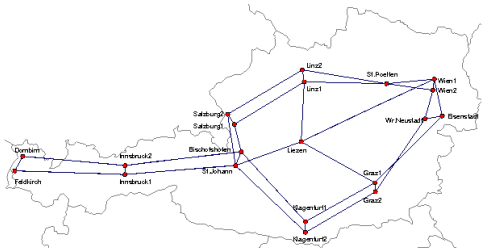
Network topologies. We have selected five realistic network topologies with different sizes, covering a range from 14 nodes and 21 physical links up to 50 nodes and 88 physical links. For all networks, the physical topologies are displayed in Figure 5.1, and the particular sizes with some further characteristics are listed in Table 5.1, focusing on purely topological properties. As shown in the second column, we replace the long network names by shorter identifiers in the following. Besides the number of nodes and physical links in the next two columns, the fifth column provides the link density of the topologies, i.e., the physical link number as percentage of the number of links in a complete graph. The sixth and seventh columns display the maximum and mean degree of nodes in the networks, and the last two columns give the same statistics for node pair connectivities. These connectivity values refer to all possible node pairs. In all networks, the minimum node degree is two, and the minimum connectivity is two as well.



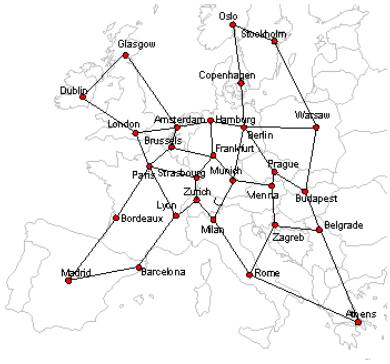
(a) *USA network (pt14).*



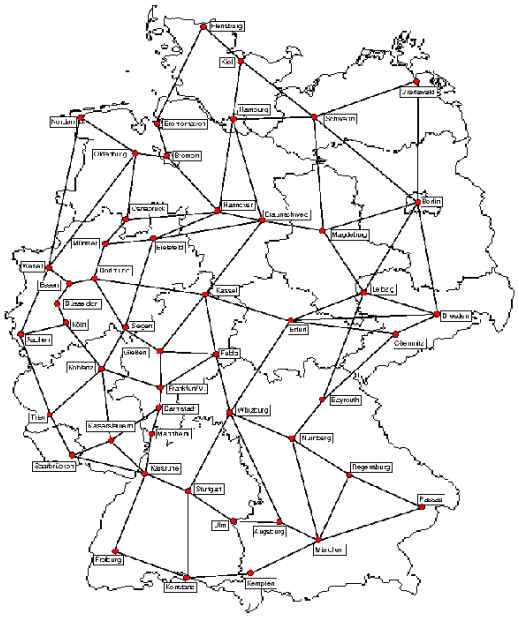
(b) *Germany small (pt17).*



(c) *Austria (pt20).*



(d) *Europe (pt28).*



(e) *Germany large (pt50).*

Figure 5.1: *The network topologies for the study.*

physical topology	topology identifier	$\|N\|$	$\|L\|$	density (L)	node degrees max	node degrees mean	connectivity max	connectivity mean
USA network	pt14	14	21	23.08%	4	3.00	4	2.74
Germany small	pt17	17	26	19.12%	6	3.06	4	2.18
Austria	pt20	20	33	17.37%	5	3.30	4	2.72
Europe	pt28	28	41	10.85%	5	2.93	4	2.33
Germany large	pt50	50	88	7.18%	5	3.52	5	2.78

Table 5.1: *Characteristics of the network topologies.*

All networks have a meshed physical topology, but a closer look exhibits some structural differences. The link density decreases with increasing network size. The USA network (pt14), though with a mean node degree of only 3.0, has a well linked structure yielding a high mean connectivity. In fact, over 72% of the node pairs are at least 3-connected. In comparison, the small Germany network (pt17) has a slightly higher mean node degree, but a much lower mean connectivity due to its layout, allowing for many node cuts of size two. Here, over 80% of the potential communication endnode pairs are only 2-connected. These two and the large Germany network do not have a significant shape, while the Austrian network (pt20) is obviously composed of some connected rings with a central hub. The larger ring is doubled for reliability reasons and forms, together with the hub, a quite dense and highly interconnected structure. In total, over 63% of the node pairs are at least 3-connected, and leaving out the four nodes in the attached tail towards Feldkirch, this value increases to 100% with a mean connectivity of 3.13. The European network (pt28), in contrast, has the topology with lowest mean node degree, showing a well interconnected central region which is surrounded by some simple rings to cover remote locations. Since most nodes on these rings have just two incident links, the mean node degree is below three for this instance, and in total about 67% of the node pairs are only 2-connected. Finally, the large Germany network (pt50) is the largest instance and has, at the same time, the topology with the highest mean node degree. Here, nearly 16% of the node pairs have a connectivity of at least four. Altogether, the selected topologies cover a mix of different sizes and structures of meshed networks as they occur in practice.

Demands and survivability requirements. Each topology comes with prespecified static traffic demands to establish in terms of optical connections. The particular traffic matrices are listed in Appendix B.1 beginning at page 249. Moreover, we distinguish four levels of survivability. Each level is defined by a percentage which specifies uniformly the portion of each commodity that has to be able to survive any single link or node failure. Thereby, fractionals are always rounded up to the next higher integer value in order to specify optical protection requirements connection-wise. The four considered levels are 0% (unprotected traffic), 50%, 75%, and 100% (full protection), and we abbreviately refer to these levels by s0, s50, s75, and s100, respectively. In this setting, the particular numbers of connections to protect for each commodity can be easily computed from the traffic matrices in Appendix B.1 and are thus not listed explicitly. Table 5.2 subsumes some characteristics of the commodities and survivability requirements for each network. Here, the second column lists the total number of commodities, the third column holds the commodity density as the number of node pairs to connect divided by the total

topology	$\|\mathcal{Q}\|$	total demand	density ($\mathcal{Q}$)	total asked to survive			
				s0	s50	s75	s100
pt14	91	2710	100%	0	1379	2067	2710
pt17	121	1021	89%	0	535	814	1021
pt20	112	702	59%	0	395	597	702
pt28	378	1008	100%	0	629	907	1008
pt50	662	2365	54%	0	1226	2097	2365

Table 5.2: *Characteristics of the traffic demands.*

number of different node pairs, the fourth column shows the total demand value summed up for all commodities, and the last four columns give the total number of protected connections for the different survivability levels.
The instances pt14 and pt28 have a full demand matrix, i.e., each node pair defines a commodity with traffic to establish. Nearly the same holds also for pt17 where traffic to all other nodes exists from each location except for one node, Norden, which has only a single commodity to Frankfurt (since Norden hosts just a second hub for transatlantic traffic being managed in Frankfurt). In pt20, the original traffic demands have been very small and thus are scaled up with a convenient factor to balance the traffic with the hardware capabilities (avoiding to use for this one case a separate hardware model, see below). Note that the scaling maintains the commodity structure and was chosen such that the property of a low traffic instance was preserved. Moreover, pt20 and pt50 do not have full demand matrices, both hold a little bit more than half of all possible commodities. Finally, none of the instances have a regular traffic distribution. With varying relations, there is a mix of commodities of small and large demand values, reflecting the interaction between metropolitan regions and areas with a more rural character. In total, the test set combines different sizes of topologies and traffic loads and thus provides various comparison possibilities.
In view of the survivability concept comparison and the relevance of network connectivities for DSP, we refine the topological information from Table 5.1 with respect to the particular traffic relations. For each instance, Table 5.3 shows in columns two to five the number of commodities whose endnodes have the corresponding node connectivity. In addition, column six gives the mean connectivity over all commodities, while the mean over all commodities weighted with their demand value is displayed in the last column. Hence, the latter values refer to the mean connectivity per optical connection to establish. These statistics reveal the flexibility of survivable routings for the instances. Note that the weighted means are always larger than the mean values, indicating that commodities with higher connectivity tend to have larger demand values, and the particular increase reflects the extent of this effect.

Hardware model. For generating comparable results, using exactly the same hardware setting for the instances is preferable, but hampered by several aspects. As already mentioned, one such issue is the balance of traffic flow sizes and hardware capabilities. Operators would neither use great numbers of small capacity devices when larger modules are available (and typically more cost-efficient due to economies of scale), nor employ oversized devices for accommodating small flows and thus end up with large amounts of spare resources. Therefore, the traffic in pt20 has been increased, as already mentioned. This allows to use basically the same set of

topology	commodities of connectivity				connectivity	
	2	3	4	5	mean	weighted mean
pt14	25	65	1	0	2.74	2.79
pt17	97	23	1	0	2.21	2.32
pt20	41	55	16	0	2.78	2.91
pt28	255	121	2	0	2.33	2.35
pt50	188	330	130	14	2.95	3.03

Table 5.3: *Connectivity characteristics of the commodities.*

hardware modules with uniform capacities for all instances. In addition, we focus on a small variety of devices in order not to distort main observations and conclusions. Table 5.4 lists all hardware modules with their properties and cost values. Unless stated otherwise, this hardware model is applied throughout all computations in our study. With a single fiber type, physical and supply links coincide, and we use the same link lengths in both cases. The applied cost model has been provided by T-Systems and reflects realistic cost relations, while the individual cost values have been scaled by an appropriate factor.

A further aspect is the limitation of the maximum optical transmission distance, in particular in relation to the link lengths as unchangeable characteristic of the topologies. The considered networks cover geographical regions of different sizes. Both Germany networks pt17 and pt50 as well as the Austria network pt20 are limited to relatively small countries, whereas the USA network pt14 and the Europe network pt28 span continents. As a consequence, the topologies contain physical links with lengths in different magnitudes, as illustrated in Figure 5.2. So, assuming a uniform hardware model with a common maximum optical transmission distance for all instances would provide predictable but undesired effects in view of the number of required signal regenerations. Instead, we make the following distinction. In the geographically small networks, the modules are equipped for long-haul transmissions with an optical reach up to 1200 km, while ultra long-haul transmissions in the large area networks enable a maximum distance of 8000 km to be bridged optically. Reflecting the applied technologies, the transmission-relevant equipment, namely amplifiers (and thus fibers), WDM systems, regenerators, and wavelength converters, cause different costs in both settings. This also shifts the total cost ratio for links

module type	module name	install at	capacity	cost	pt17,pt20,pt50	pt14,pt28
fiber	`Fiber`	links	1 WDM system	per km:	0.05	0.05
				each 70 km:	6.00	9.00
WDM system	`WDM40`	links	40 wavelength channels		24.00	36.00
switch	`OXC64`	nodes	64 ports (bidir.)		193.20	193.20
switch	`OXC128`	nodes	128 ports (bidir.)		312.40	312.40
regenerator	`REG`	nodes	1 optical channel		2.00	3.00
converter	`CONV`	nodes	1 optical channel		2.00	3.00
					reach: 1200 km	reach: 8000 km

Table 5.4: *Installable hardware modules.*

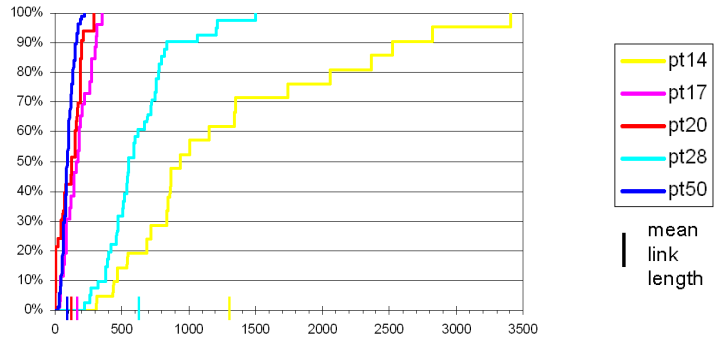


Figure 5.2: *Overview on link lengths in the topologies, illustrated by the accumulated link length distribution function* $F(\omega) = |\{\, \ell \in L \mid \omega_\ell \leq \omega \,\}| \,/\, |L|$ *as percentage of links with length at most* ω *km, and an additional mark on the mean link length in each network.*

and nodes in the designs. In the large networks, longer links and the more expensive transmission equipment yield higher cost for installing link capacities in comparison to the smaller networks, whereas the node capacity costs are equal in all of them. Hence, a larger portion of total cost is to be expected in the larger networks for link capacities.
Note that except for fibers, Table 5.4 displays for all modules the resulting total cost per device. According to our cost model, these costs can be (and are) composed of various cost components, like a base cost plus port cost for the OXCs, for instance. For the fibers, these components are listed individually, as the fiber cost is length-dependent and thus yield link-individual costs. Thereby, no fixed cost for installation of fibers is charged. Moreover, the cost per km accounts over the full length, while the segment cost is only incurred whenever exceeding a (multiple of) 70 km segment length. So, for a fiber $f \in \mathcal{F}$ with cost C^f_{km} per km and segment cost C^f_{seg}, the total cost C^f_ℓ for installation on a supply link $\ell \in L$ with a length of ω_ℓ km, $\omega_\ell \in \mathbb{R}_+ \setminus \{0\}$, is computed by $C^f_\ell = \omega_\ell C^f_{km} + (\lceil \omega_\ell / 70 \rceil - 1) C^f_{seg}$. For example, a link of length 210 km in pt17 costs 22.5, while 211 km yield a total cost of 28.55 for a fiber.

Computing environment. For the study, all computations have been carried out on Linux-operated PCs with a 3.2 GHz Pentium 4 HT processor and 2 GB main memory. The described methods are implemented in C++. In addition, we make use of the following external software libraries. We employ LEDA version 4.1 [114] for graph data structures as well as basic graph algorithms. ILOG CPLEX version 9.130 [71] together with ILOG CONCERT technology version 2.1 is applied for LP solving and small auxiliary subproblem ILP solving. Moreover, as described in Chapter 3.3, the dimensioning and routing subproblem is handled by the tool DISCNET developed at ZIB and marketed by the spin-off company atesio (see also the website of atesio [9]). DISCNET also involves the use of LEDA and CPLEX (in the above versions).

With these settings and prerequisites, our computational studies are carried out and described in the following sections. As in any experimental work, we are aware of the fact that the described observations and drawn conclusions are clearly restricted to the particular instances and parameterizations chosen for each scenario

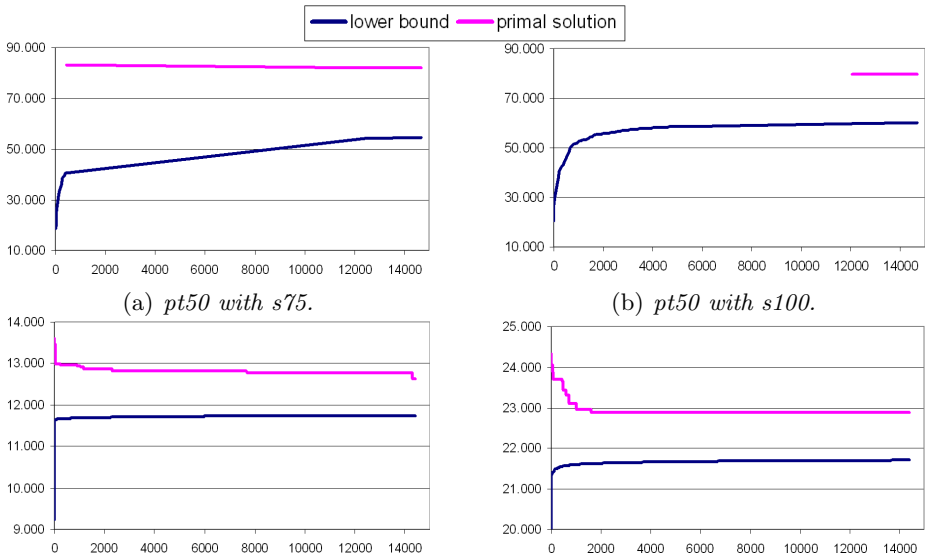
and thus need not reflect general relations and properties. Nevertheless, the reported experiences might serve as indicators for interesting issues and prepare extended examinations for further research.

5.2 Computations for optical network design

In the first computational study, documented in this section, we evaluate the performance of solving optical network design problems and compare the obtained solutions for different settings. Regarding the performance, we focus on the provable solution quality. Since finding global optima cannot be guaranteed by our methodological approach (in a timely manner), we run the algorithms with time limits which prevents an evaluation in terms of computation time. The comparisons for different scenario settings are organized in case studies, where each variation is considered individually in relation to the set of reference solutions described first. This way, we consider alternative protection schemes, the case of upgrade planning, an extended hardware model, and the opaque architecture.

Algorithmic outline. All computations follow the same algorithmic outline, sketched in Figure 3.3 on page 106. The method consists of two consecutive steps, the dimensioning and routing optimization without distinction of wavelengths, and a subsequent wavelength assignment in case of transparent networks.
For the dimensioning and routing subtask, we apply Discnet with the modeling by use of capacity-cost lists. These lists are generated in a preprocessing step by dynamic programming, as presented in Section 3.3.1, and handed over to Discnet by a file interface. The main procedure of Discnet, the branch-and-cut method with integrated heuristics to generate integer flows (see Section 3.3.2), is then run with a time limit of four hours CPU time, returning the best solution(s) found. Finally, a postprocessing step tries to improve the obtained routings and accomplishes solutions for the transparent multi-hop case accordingly by placement of required regenerators using the physical topology based method described in Section 3.3.3.2. The subsequent wavelength assignment is performed by a short heuristic run. For this, we apply the iterative sequential wavelength assignment algorithm as described in Section 4.4.1.3 and depicted in Figure 4.6 on page 161 with the APRR reordering scheme and a time limit of 15 minutes.

Total computation times. Due to the fixed time limits for the main algorithms, the total computation times for the instances vary only marginally, and the fluctuations can be easily explained. So, we only give a brief overview on these values.
The time limit for the dimensioning and routing part has been carefully set such that substantial progress during the optimization is not cut off and a feasible solution is found for all (reference) instances. So, the larger instances have been naturally responsible for the actual selection. For illustration, Figure 5.3 shows the branch-and-cut progress for some selected instances. The largest topology pt50 sometimes awaits a longer period until a feasible design is completed, as for the full protected case in Figure 5.3(b), and usually shows also significant progress in that time, see Figure 5.3(a). For the smaller topologies, further improvement steps, as visible in Figure 5.3(c), occur from time to time, while a fast initial convergence of primal and



(a) *pt50 with s75.* (b) *pt50 with s100.*

(c) *pt17 with s0.* (d) *pt17 with s75.*

Figure 5.3: *Dimensioning and routing optimization progress for some selected instances, showing the primal and dual bound values over the CPU time in seconds.*

dual bound can also be followed by a quite stable continuation in the following, as in case of Figure 5.3(d).
In all computations, the branch-and-cut algorithm in the first phase occupied the full time span of four hours plus a small period to complete examination of the last considered search tree node before exceed of the time limit is detected. Hence, no guarantee to have found an optimal solution of the dimensioning and routing subtask can be given for the test set, though some instances show to be close to optimality. The fluctuation in computation times is reasoned by the wavelength assignment algorithm in the second step. Finding a converter-free solution allows to terminate immediately, which regularly happened within the 15 minute time window. All other instances have run the full quarter of an hour and, thus, show a total computation time of at least 15300 seconds (= 4.25 hours), but rarely run much longer than this. In fact, exceeding the limit sum by more than 150 seconds happens only for the largest pt50 instances which is only due to the considerable rise of the LP size in this case, causing longer CPU times for solving individual branch-and-cut nodes.

5.2.1 Reference solutions

At first, we report on the results obtained for the reference setting, using the specifications from Section 5.1 and DSP as protection scheme. These solutions allow to look on the achievable design qualities and to estimate the extent of quality loss entailed by the applied decomposition approach.

Solution qualities. Our solution methodology delivers a feasible network design with specified total cost $\hat{C}$ and a cost lower bound $\hat{L}$ for any possible design satisfying the requirements. So, as a measure for the outcome quality, we use the optimality gap derived from these two values as that portion of the solution cost that can at

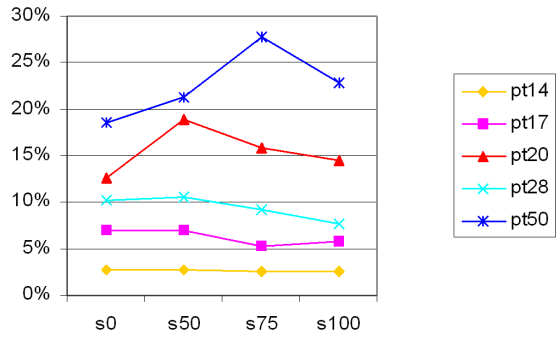


Figure 5.4: *Optimality gaps for all reference instances.*

most be saved by an (unknown) optimal solution, i.e., defined as $\frac{\hat{C}-\hat{L}}{\hat{C}}$. Note that the optimality gap is limited by 100% this way. An alternative measure often used in the literature is to divide by the lower bound $\hat{L}$, relating any gap to the optimum approximation rather than to a solution of unknown quality. However, this variant fails when the lower bound is zero, which happens often for the converter number in wavelength assignment. In order to use a uniform measure, we therefore apply the above described method, expressing optimality gaps as portion of the solution value.

For the reference scenario, Figure 5.4 plots these values instance-wise for all four survivability levels considered. The picture shows a clear ranking of the networks in terms of solvability. Surprisingly, this order does not always follow the instance size relation, since all solutions for pt28 show a lower optimality gap than those for pt20. Note that pt20 is not only the smaller network, but carries also less traffic. This issue, indeed, might give a hint that networks with low amount of traffic hamper the derivation of helpful lower bounds, yielding bad optimality gaps though having already good designs at hand. In network design, lower bounds are usually drawn on estimations of unavoidable traffic throughput on (generalized) cuts in the network and thus can suffer from high connectivity in combination with low traffic. Another possible explanation is based on the network connectivities which, except for pt14, increase in the order of decreasing solution qualities, cf. Tables 5.2 and 5.3. Remind that DSP includes for each commodity the decision about the total number of connections to establish, which drives the extent of how much to spread the final routing. Higher connectivities in meshed networks allow for a fast growing variety of alternative routings to take into account, which might slow down gap closing. This effect intensifies in larger networks, and thus pt14 might profit from its smallness which restricts routing variety with a (more than) compensating counter-effect on tractability. However, an evident reason for the observed translation of pt20 cannot be identified. All other instances show the expected behavior, where reaching a certain solution quality for growing instance sizes usually requires longer executions of the optimization procedure, typically with an exponential increase.

Comparing the solution qualities along different survivability requirements in a network does not show a clear tendency. Here, any kind of slope can be observed throughout the instances, whatever the compared levels are. Hence, solvability does not correlate much with the portion of protected traffic.

Altogether, the obtained solution qualities are satisfactory in view of the restricted computation duration. With this affordable effort, 70% of the generated designs

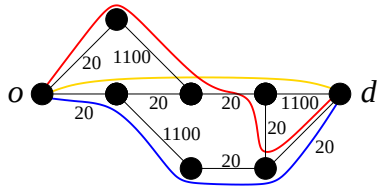
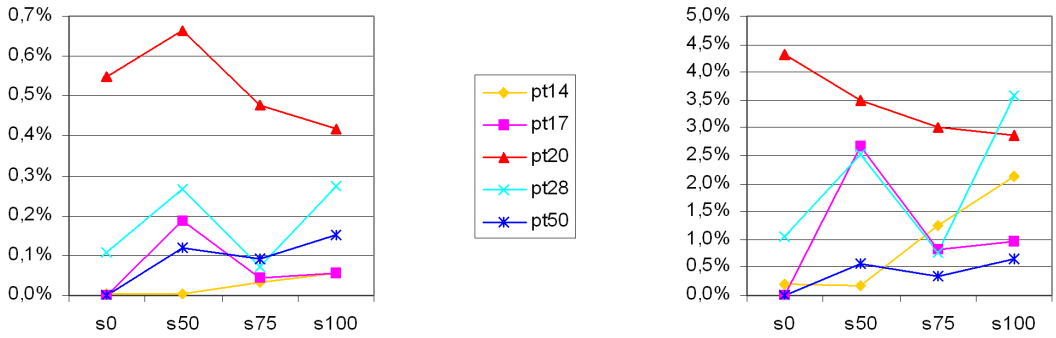


Figure 5.5: *Example for the invalidity of half a shortest cycle length as lower bound for the length of connections protected by* DSP*. For a commodity q between nodes o and d requesting $v_q = 3$ connections of which $v_q^* = 1$ is to be protected with* DSP*, the displayed routing is feasible. No connection is longer than 1180 km, whereas the shortest cycle has a length of 3400 km.*

are closer than 15% to optimality, and among these, again over 70% show even a gap below 10%. Besides the computed solutions as ready-to-install optical network designs, the accessory determination of such a quality guarantee provides valuable assistance for planners in practice. Hence, the proposed decomposition methodology forms a suitable approach to handle this complex task.

Decomposition loss. Regarding the solution approach, a further interesting performance characteristic is the quality loss suffered by the applied decomposition. An exact evaluation would require to compare the optimal network design cost with the cost of the best solution achievable with the decomposition method. As both values are unknown and very hard to find, we have, second best, to use a worst case estimation. For this, we relate the cost incurred by aspects excluded from the core optimization to the total design cost and to the difference of final solution value and total cost lower bound, i.e., the cost corresponding to the obtained optimality gap. With the decomposition, the relaxed aspects concern the lightpath length limitations, satisfied supplementary by adding required regenerators, and the simultaneous determination of a conflict-free wavelength assignment, which is carried out subsequently with possible inclusion of wavelength converters. Having a lower bound on the number of unavoidable converters or regenerators, the corresponding cost can be added in our setting to any total cost lower bound determined for the dimensioning and routing subproblem, thus reducing the maximum decomposition loss, too. Such non-trivial lower bounds for converter and regenerator numbers are indeed rarely found. For conversions, the lower bounds presented in Section 4.4.2 refer to a prefixed routing and do not apply as a general instance bound which must hold for any feasible routing. In fact, if installable wavelength capacities are unbounded, this bound is always zero. For the number of unavoidable regenerators, a simple lower bound can be obtained if a shortest path for a commodity already exceeds the optical reach. Any connection routed for this commodity will employ at least the number of regenerators needed for a shortest path. The least length estimation can be improved to half the length of a shortest cycle for any connection protected by 1+1 path protection. The protection scheme DSP surprisingly prohibits the use of half the shortest cycle length as minimum connection length estimation, as proven by the example in Figure 5.5. However, such an implication of unavoidable regenerators does not occur in any instance, whose maximum shortest path and cycle lengths (over all node pairs) are listed in Table B.1 on page 252 in the appendix. Therefore existence of network designs at the cost of the derived dimensioning and



(a) *Total cost portion* $\Gamma/\hat{C}$. (b) *Gap cost portion* $\Gamma/(\hat{C}-\hat{L})$.

Figure 5.6: *Solution quality losses caused by the applied problem decomposition, where the total cost* Γ *for converters and regenerators is plotted in relation to (a) the total solution cost* $\hat{C}$*, and (b) the optimality gap* $\hat{C}-\hat{L}$.

routing lower bound without further need of regenerators and wavelength converters cannot be excluded, and the total cost for all employed wavelength converters and regenerators, denoted by Γ, in fact provides the worst case decomposition loss.
For the reference instances, Figure 5.6 plots the maximal decomposition loss as portion of the total solution value, i.e., $\Gamma/\hat{C}$ as loss in absolute quality, and as portion of the optimality gap, i.e., $\Gamma/(\hat{C}-\hat{L})$ representing the loss relative to the proven design quality. The results show that the decomposition does not worsen the solution qualities too much. For the absolute quality depicted in Figure 5.6(a), the pt20 instances break ranks again and give the weakest results, though even here the maximum total cost portion of the loss is less than 0.7%, whereas all other instances show less than 0.3% potential total cost savings encumbered by the decomposition. Remarkably, a network ranking reveals the same order as for the solution qualities in Figure 5.4 except for pt50 with quite good results in terms of decomposition loss. This surprising performance of pt50 becomes even more significant when changing to the relative loss measure in Figure 5.6(b). Here, pt50 outperforms all other instances except for the half-protected case (s50) of pt14. However, the former quality ranking of the other networks disappears, although the relations among the results for each individual network (with a single exception in the half-protected pt20 case) kept their shape, only being stretched or compressed in scale. So, the influence of the decomposition approach on the design qualities shows unpredictable behavior, but to an overall small extent.
These observations allow for several conclusions. The plots reveal that the quality loss by decomposition is quite independent of the instance size. Besides the good performance of the largest network pt50, a further indication is given by the fact that the total cost portions over most networks are lying close together. From this, only the medium-size network pt20 deviates. Note that this network holds the least traffic, and surprisingly, a network-wise tendency of lower total cost portions for higher traffic amounts can be detected. Hence, higher traffic requirements seem to result in sparser requirements for conversion. We guess that conflicts most often involve very few lightpaths and can be dissolved when more lightpaths and thus wavelengths are present. For instance, consider the three specific connections in a star subtopology as shown in Figure 4.3(a) on page 153, where a converter is unavoidable. Note that in the same example, twice the wavelength availabilities allow to accommodate twice

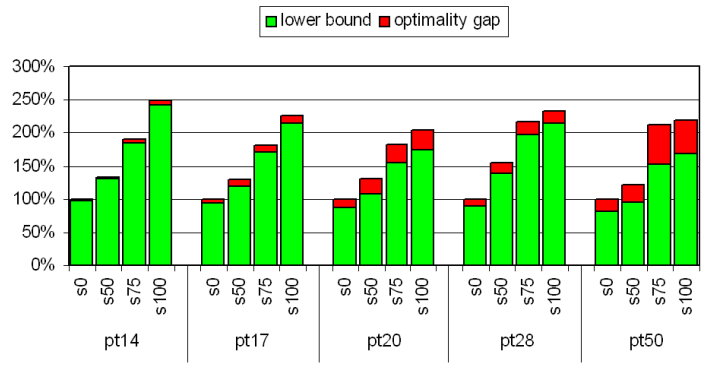


Figure 5.7: *Growth of the total cost with increasing protection level, normalized to the s0 design cost for each network. In addition, each total cost bar is split into the lower bound (bottom) and the optimality gap (top) parts.*

the lightpaths without conversion, and the triple size example again requires just a single converter, whereas the other costs increase. So, the number of converters and thus the conversion cost portion scales well with growing wavelength multiplicities and lightpath numbers in these instances. In a similar manner, instances with higher wavelength multiplicities, even if carrying more lightpaths, could better allow to avoid conflicts and thus to reduce the total portion of lightpaths needing conversion. This effect can reason the ranking observed in Figure 5.6(a), and we conclude that the study results follow our conjecture by which a larger amount of traffic correlates with a potentially decreasing cost portion for needed converters.
While such relations can be identified in the total cost portions, the mixture of values for the relative loss indicates that no correlation of gap sizes and decomposition losses exists. Hence, we can assume that the problem decomposition has a random, but limited influence on the quality of the computed solutions. The worst case cost portion as the maximum quality loss suffered by exclusion of regenerator placement and wavelength assignment is below 5% over all instances and thus occurs as reasonable price for the achieved increase in solvability. The dimensioning and routing part is obviously responsible for most of both the total cost and the solution quality. Clearly, both aspects depend on the cost relations in the hardware model, with comparatively low costs for regenerators and converters. However, these relations reflect current market prices, and most practitioners expect the relations to be stable or to become even more advantageous by further progress in regeneration and conversion technologies. As long as this holds, the decomposition loss will remain very low, making this solution approach well suited for real-world application.

Cost of survivability. After considering solution qualities, we turn to the particular design cost results. While a comparison of designs for different networks is of minor significance, an evaluation of the cost increase for higher protection levels is of interest. Figure 5.7 illustrates these cost growths, normalizing all values to the unprotected design cost in each network. In addition, the lower bound and the optimality gap portions are separated to take the design qualities into account within the discussion.
The plots show that full protection produces typically more than twice the cost of a design with unprotected traffic. At first sight, one might wonder whether dou-

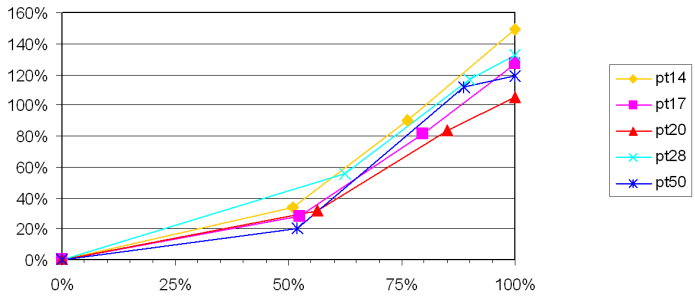


Figure 5.8: *Cost increase in relation to the total cost of the unprotected design against the portion of in fact protected connections at the levels s50, s75, and s100 for each network.*

bling a network's hardware does not provide a cheaper alternative. Such a doubling, indeed, implies to set up a completely separated network, without common use of node equipment stations, fiber strands, or any other infrastructure endangered to fail. Due to these requirements, a surrogate network 'copy' would cause much more than the original network's cost.

When comparing the particular cost growths for the networks, some interesting properties can be observed. In most cases, only a small additional cost is incurred when changing from unprotected traffic to the s50 level. This results from the fact that DSP typically needs to establish only a small number of additional backup connections for protection of up to half of the traffic and the required diversification of the routings (cf. Table 2.1 on page 91), though typically increasing mean connection lengths, can be realized without occupying a lot of further resources. Here, DSP profits most from its flexibility to select for each commodity those routings which combine in the best possible way from the entire network's perspective.

For the further changes to higher protection levels, varying slopes for the cost growths occur. While nearly the same cost increase for each additional 25% traffic protection is incurred in the smaller networks, the larger ones show a much higher step from s50 to s75 than from s75 to s100. Thereby, note that the total costs and the corresponding lower bound values behave similarly, i.e., the increase from one level to the next is of comparable size, though the optimality gaps tend to grow with higher protection. From this, we conclude that the achieved solution qualities have only marginal influence on the observed effects.

An explanation for the growth differences is, though intuitively suggested, hardly provided by the topology sizes, and most arguments in this direction contradict other properties, like the fact that the additional cost for s100 increases follow another ranking. Moreover, the observed additional cost coincide neither with traffic volume nor connectivity relations of the instances. For example, both pt14 and pt50 have a similar total number of demanded connections and comparable connectivity values, but discriminative slope shapes in Figure 5.7. Instead, structural properties of the commodities provide an explanation, also for the exceptional high cost growth for s50 in pt28.

Table 5.2 on page 191 lists for each protection level the total numbers of connections that are in fact guaranteed to survive single node or link failures. Due to rounding of fractionals, these numbers are *at least* the level percentage of the total connection number and can form a substantially larger portion. Actually, the

growths of these numbers are highly correlated with the observed cost steps. For illustrating this correlation, Figure 5.8 puts the two issues into relation. For each network, the cost increase from the unprotected design is plotted over the portion of connections that is really protected at the individual protection level. Obvious similarities over all networks appear in this measure. The low incline from s0 to s50 has already been reasoned above. For pt28, the higher cost increase at level s50 results from effectively protecting more than 60% of the traffic, which indicates that many commodities with small odd demand value are involved (cf. Table 2.1). At the next level s75, both pt28 and pt50 show the highest overprotection, with more than 13% added to the requested 75% protection. This explains the larger step from s50 to s75 for these two networks as well as the subsequent much smaller increase for completing protection for all connections in s100. Especially in pt50, s75 leaves very few connections unprotected, and thus the reliability completion becomes achievable with a small number of further backup connections at marginal surplus costs.

Node vs. link cost. Moreover, the relation of node and link costs is of interest in view of the applied hardware cost models. Remind that we differentiate the optical reach on the area size covered by the networks and therefore charge different transmission hardware costs, while the node hardware costs are uniform. For all reference instances, Figure 5.9 depicts the proportional breakdown of the total cost into the node cost and the link cost.
First of all, the results confirm the expectation stated with the cost model description in Section 5.1. The small area networks show a higher node cost portion than the US and the European network, where much longer fibers and the need for the more expensive ultra long-haul transmission hardware result in a higher fraction for the link costs. In fact, the node costs dominate the total design expenses in the German and Austrian topologies. In the European network, both cost portions are nearly balanced, and in the furthest expanded US network, the link costs become dominating.
Within each network, the cost relations vary only marginally for different survivability levels. For the mean portion of link cost within each network, the maximum deviation of a single value is 2.15%, and the mean deviation is 0.65%. Note also that the traffic volume for the smaller networks pt17, pt20, and pt50 is medium, small, and large, whereas very similar node cost portions are found for all of these

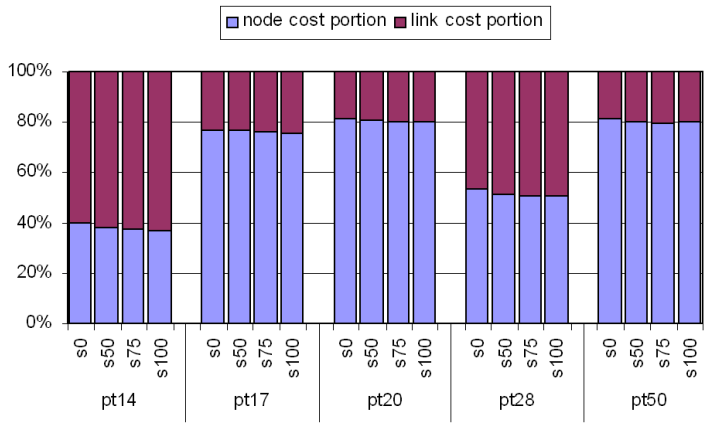


Figure 5.9: *Total design cost portions for node and link equipment.*

instances. From this, we conclude that neither the protection level nor the traffic volume drive the cost relation between link and node equipment, but this depends basically on the applied cost model relations and the network topology. The cost model constitutes the basic proportion, potentially shifted by topological properties. Considering pt17 and pt28, the lower connectivity of these instances seems to push the cost relation towards a better balance in comparison to the better interconnected networks sharing the same cost model.
Finally, the results show that both link and node equipment can form the major part of the total cost in designing realistic networks. As a consequence, optimization restricted to either of these hardware categories might produce misleading results when planning to set up a complete architecture. Hence, a precise evaluation of alternative designs can only be provided by a model involving all major cost contributors.

5.2.2 Survivability model alternatives

As first setting alternative, we compare the results for using different survivability models. Substituting DSP, we consider the two special variants 2-DSP and max-DSP as well as 1+1 path protection which is most commonly applied in practice.

Model implementation. Both DSP variants, as described in Section 2.3.2, are implemented as a special diversification model parameterization. 1+1 path protection is realized with the diversification routing model by an input modification as follows. Each commodity is split into a set of (sub-) commodities. For each connection asked to survive any single link or node failure, an individual subcommodity is set up demanding full protection. All remaining connections are grouped into a last subcommodity which is established unprotected. This way, each protected optical connection obtains finally a dedicated disjoint backup connection, and thus the union of all subcommodity's routings forms a 1+1 path protection solution for the original commodity.
We remark that our realization of 1+1 path protection differs from the implementation usually applied in practice. There, all working and backup connections for the protected part of each demand are typically routed on a single shortest cycle (computed by Suurballe's algorithm [18, 156] w.r.t. to certain length metrics), where the working connections use the shorter of both origin-destination paths. For additional unprotected connections, Dijkstra's shortest path algorithm is used to carry out the routing. In contrast to this, our model does not restrict on uniform (pairs of) paths, but allows each connection of a demand to select a route (and its backup route) individually and independently of the other connections of that demand. Since these selections are decided within the cost minimization, the implemented 1+1 path protection model can be seen as optimization variant, and thus our results represent conceptually a kind of lower bound for the Suurballe/Dijkstra-based variant. In fact, any solution feasible for the scheme typically applied in practice is obviously feasible in our model as well, but not vice versa. The cost lower bound we compute is in particular a cost lower bound for the more restrictive concept, too. Hence, the following discussion regarding conceptional performance carries over to any Dijkstra/Suurballe-based 1+1 path protection scheme as well (with at least the same significance).

Design costs. At first, we turn to the performance comparison of the concepts in terms of the total design costs and quality guarantees. For this evaluation, Figure 5.10 puts all these values into relation within a comprehensive overview. The bars represent the solution costs and are subdivided into the lower bound value and the additional cost corresponding to the optimality gap. This way, the ranges on top of each bar mark the cost interval in which optimal solutions must lie. All values for each instance are normalized to the unprotected DSP solution's total cost. Note also that the ordinate scale begins at 70% of the total cost.

First of all, the results for the unprotected case are extremely similar among instances of each network. Hence, these benchmarks indicate that differences in the computation models and their implementation have only marginal influence on the outcome. Note further that non-overlapping gap ranges directly prove the higher bar's concept as worse even for its optimal solution (although not known), as its dual bound is already higher than a primal solution value for the other concept. This holds for 1+1 path protection throughout all half-protected scenarios and also for the s75 instances in pt14 and pt17 which are solved to high quality. For the remaining instances, a tendency to similar behavior can be observed, but the indication loses strength as the solution quality degrades. However, 1+1 path protection never generated the best solution throughout all instances. Hence, we can conclude that DSP and its variants outperform 1+1 path protection.

Interestingly, DSP outperforms all other concepts in the s75 and s100 instances of pt14 and nearly all in s75 of pt17, while the size of its optimality gap shows the highest increase with growing network sizes. On the one hand, these observations confirm that DSP has the best potential for a superior concept, but is, on the other hand, also most difficult to solve. This is best demonstrated in pt20 and pt50 for the two highest survivability levels. Both of these characteristics of DSP are reasoned by the same fact. Any demand routing feasible for any other concept is feasible for DSP as well, but not vice versa, thus optimal DSP solutions will never be worse than those for the other concepts. In fact, the idea of DSP is especially to decide about an appropriate routing for each demand from a general perspective, with choice of many alternatives including varying numbers of connections. This additional degree of freedom, in turn, makes it harder to solve the model.

Comparing the variants 2-DSP and max-DSP with fixed parameters, one observes that a more extensive spreading of the demand routings by max-DSP, minimizing the total number of required connections, is of varying benefit. In some cases, such as s100 for pt14, pt20, or pt50, the max-DSP approach seems to be helpful, but becomes contra-productive in other scenarios, like for s75 and s100 in pt17 and pt28. To interpret this effect, it does not suffice to argue only by the provided network connectivities which are larger in the max-DSP favoring networks, as a more diversified routing usually requires less, but longer routing paths and thus consumes capacity from more links and nodes. So, the observed relations indicate that the additional capacity consumption needed by prolonged connections is overcompensated by the reduced connection number, whenever the meshed networks are not too narrow. Otherwise, exploiting a maximum connectivity can in fact require to take overlong detours, and more but shorter connections become the better choice. The network pt28 provides a good example where a well interconnected center is surrounded by some rings covering remote areas. For some commodities, a maximum spread forces to route some connections along these rings which therefore have to

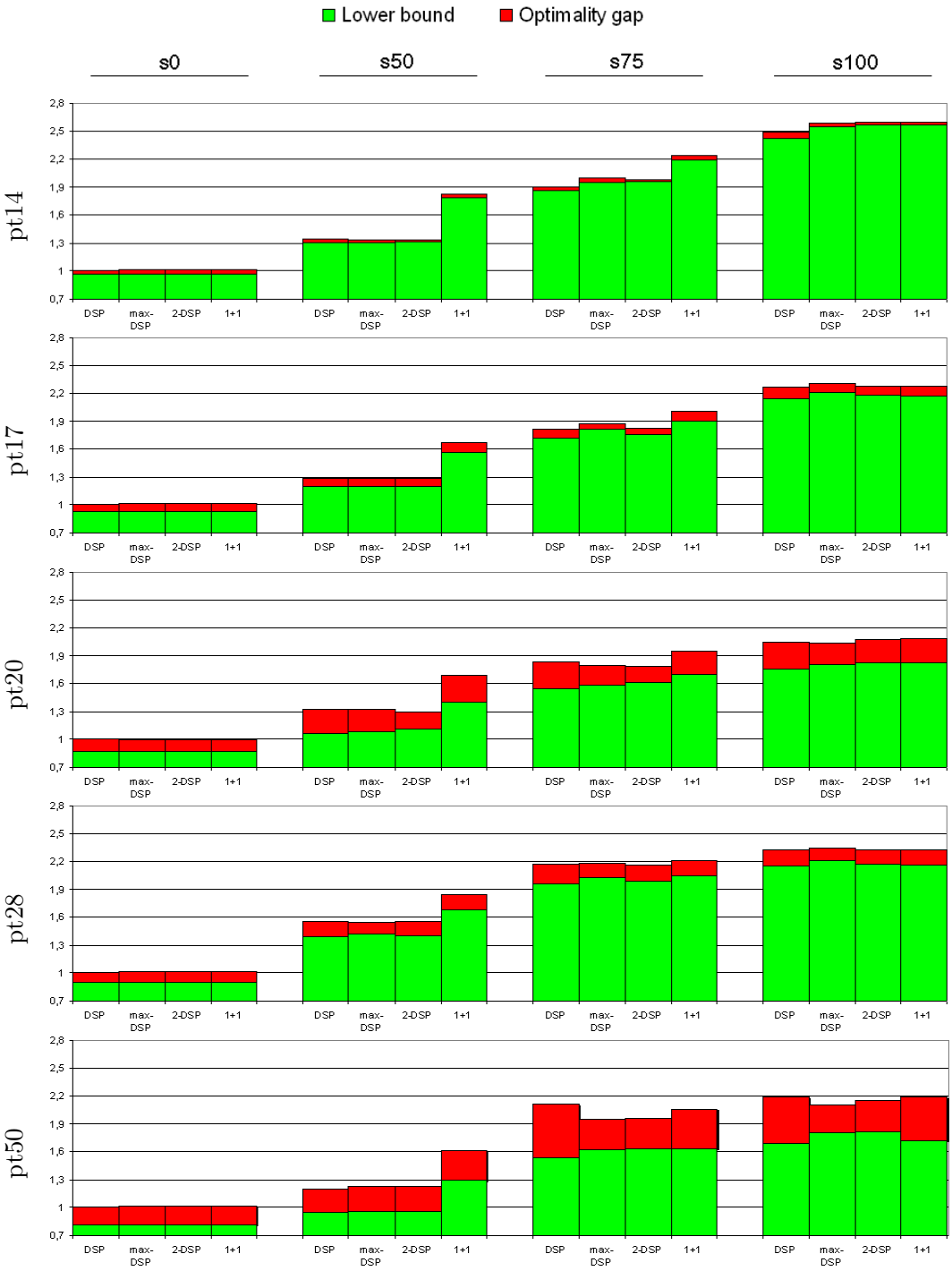
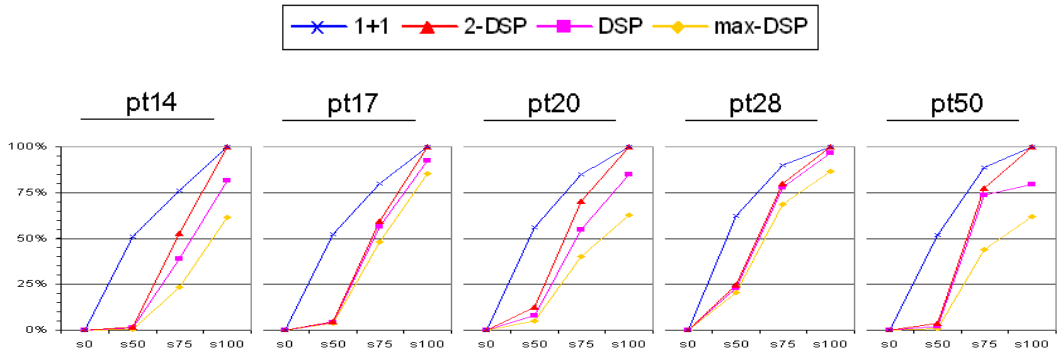


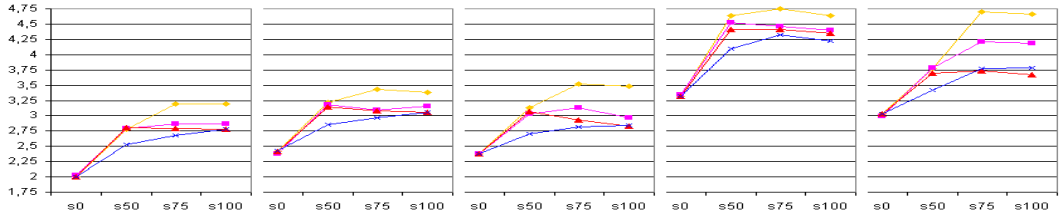
Figure 5.10: *Comparison of solutions and their qualities for the survivability models, subdividing the total cost bars into the lower bound value (bottom) and the cost corresponding to the optimality gap (top). For each instance, all values are scaled with the total cost of the unprotected* DSP *solution. Note further that the ordinate scale begins at 70% of the solution cost, distorting the gap proportion to provide better comparison capabilities.*

provide the requested capacity on all their hops, resulting in higher total cost than by application of 2-DSP.
Furthermore, note that in the solutions of the s100 scenario for the three better connected networks pt14, pt20, and pt50, the concepts max-DSP, 2-DSP, and 1+1 path protection show in this order growing design costs. This holds also for the corresponding lower bound costs except for pt50 with 1+1 path protection. In the two other networks pt17 and pt28, exactly the contrary holds. From this, we conclude that in fact higher connectivity favors the savings by less connections in more diversified routings, whereas routings using more connections are more favorable in low connectivity networks. Moreover, in all but the largest network, DSP is comparable to or gives the best result due to having the choice between the solutions for all other concepts.
In general, the plots reveal also that the survivability concept selection for the two higher protection levels yields highest result differences for pt14, pt17, and pt50, while the costs and bounds are much more similar for pt20 and pt28. As possible explanation, the low amount of traffic in pt20 and the particular network structure in combination with medium traffic size in pt28 seem to equalize the competing effects, whereas dominating properties in the other scenarios enhance the (dis-) advantages of particular concepts. For instance, the size and connectivity of pt50 allows to use spare resources from installed capacities by many demand routings without large detours, favoring a more diversified routing with less connections in total.
Finally, we observe that the cost for 1+1 path protection in s50 or s75 is often not much exceeded by the DSP cost of the next higher protection level. This holds in particular for the smaller networks with high solution qualities. In pt28 and pt50, the larger cost difference from s50 with 1+1 to s75 with DSP results from the overprotection in the latter cases, as already discussed in the previous study. The shift in abscissa direction of the measure points in Figure 5.8 represents the extent of this overprotection, which has to be taken into account when putting cost values (or differences) over different survivability levels into relation. In the smaller networks, however, the demanded levels of protection are matched quite good, making these relations most representative. From these observations, we can altogether conclude that DSP allows to protect much more traffic at low additional cost in comparison to 1+1 path protection.
To summarize, the cost performance comparison of the survivability concepts reveals that DSP is overall most favorable, but also most difficult to solve. This scheme has particular strength for not fully protected demands and thus suits best for a concerted planning of protected and unprotected traffic. Moreover, networks with higher connectivity provide structural advantages to those concepts that accept longer routing paths, but save in the total number of connections, being next evaluated in more detail.

Total numbers and lengths of connections. The total number of connections required to realize a certain demand with particular reliability requirements and their hop-lengths are also important issues for network operation. Clearly, the operational effort to setup and support a lightpath configuration grows with the number and length of connections which have to be configured, tracked for their operability, and handled appropriately when failures happen. In addition, network operators prefer short transmission paths in general.



(a) *Number of established backup connections in relation to the total demand.*



(b) *Mean hop length of all connections.*

Figure 5.11: *Overviews on (a) the number of backup connections established by use of different survivability concepts as portion of the total demand, and (b) the mean hop length of all connections in each instance.*

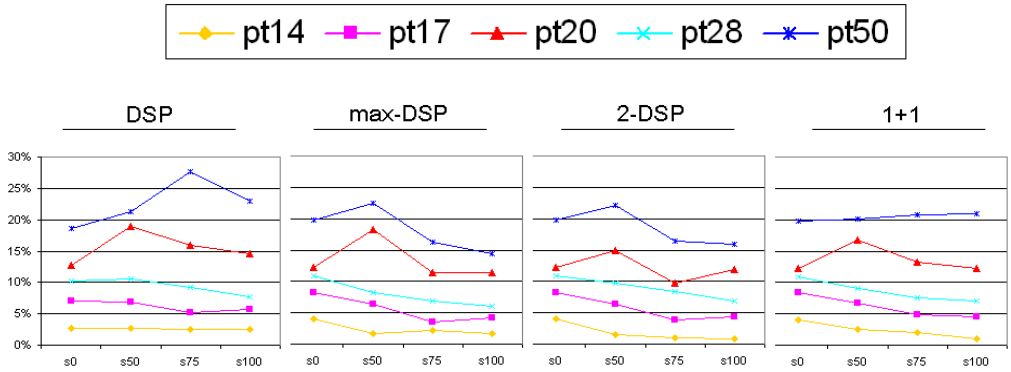
Figure 5.11 displays in the upper part, as portion of the corresponding total demand, the numbers of backup connections established by use of the compared survivability concepts. Note that the number of protected connections, as listed in Table 5.2, equals the number of additional backup connections for 1+1 path protection, where each protected working connection gets a dedicated, disjoint backup connection. Hence, the corresponding line marks also the total amount of protected traffic (and indicates also the extent of overprotection).

First of all, the plots confirm the conceptionally enforced relations. The values for 2-DSP do never exceed those for 1+1 path protection and match them in the full protected cases. Next, the backup connection numbers for DSP lie always between those for 2-DSP and max-DSP which form the two opposite extreme variants of the general concept. Remind that max-DSP in fact uses the minimum number of connections per commodity for the prescribed survivability level. Moreover, the distances between each pair of extreme values, namely those for 1+1 path protection and max-DSP, behave as expected. In relation, these distances grow strictly with the mean (weighted) connectivity of the networks, i.e., increase in the order pt17, pt28, pt14, pt20, and pt50. They are also largest for s50 and shrink with growing protection requirements.

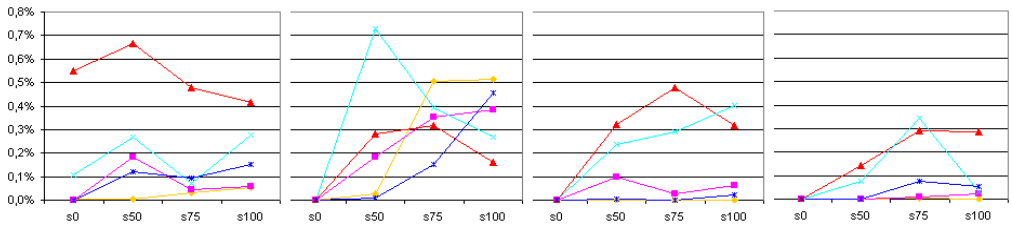
Regarding the particular values, DSP and its variants generate considerably lower total numbers of connections to establish than 1+1 path protection, in particular when the commodities are just to be secured partially. As mentioned earlier, the DSP schemes typically add very few backup connections in the s50 case. For pt28 and pt20, the values are larger due to having many commodities with odd demand values. With further increasing protection requirements, the connection numbers

then most often grow faster than the provided protection, except for max-DSP in most cases. However, DSP keeps on saving many transmission lines as far as the available connectivities allow. Hence, at the price of higher recovery effort compared to 1+1 path protection, DSP provides also advantages for the traffic management in regular operation due to holding less connections in total.
A further observation concerns the values of the 'intermediate' concepts 2-DSP and DSP. While their curves nestle more to the max-DSP curve in the least connected networks pt17 and pt28, the shapes spread more and more for increasing connectivity. The spreading is quite regular for pt20 and pt14, but somehow wavering for pt50, which probably results from the worse solution qualities. When focusing on DSP in relation to its extreme variants, the values are closer to 2-DSP for lower connectivities, then form a fair medium curve for pt20 as well as pt14, and show irregular behavior for pt50. Altogether, these relations indicate that DSP, for growing alternative variety, finds better solutions when balancing diversification extent and number of (backup) connections, rather than to favor a single extreme.
In view of the connection lengths, the bottom row in Figure 5.11 displays the mean hop lengths of connections, ranging between 1.99 and 4.75 in the study instances. The mean hop lengths are comparable in the three smaller networks and become higher in the two larger networks. Especially pt28 contains long connections due to its topological structure. Since extended ring subnetworks collect the remote locations for connection to the meshed network center, integration of protection often forces use of extended detours for establishing backup connections. This effect in pt28 accompanies the general trend to longer connection means in larger networks, as observable in the plots.
When comparing the connection numbers and their mean lengths, a further observation can be made. As expected, max-DSP generates least, but often extraordinary long connections, even if the total numbers are close together, as for pt17. Mostly for pt28 and pt17, this effect turns max-DSP into the worst performing DSP variant. As opposite concept, the mean lengths for 2-DSP are quite comparable with that for 1+1 path protection. For s50, the difference is larger, since 2-DSP has to spread the much less established connections more, whereas very similar lengths occur for higher protection levels. Between its extremes, the ranking of DSP is remarkable. In pt50, DSP varies its tendency in terms of connection numbers with growing protection level, but steers a middle course in the mean hop lengths. In the other networks, DSP selects medium connection numbers, but prefers most often shorter routing alternatives, being (much) closer to 2-DSP. From this, we conclude that DSP in general tends more to use shorter connections with less resource consumption while making clever use of the additional routing variability by selecting a suitable number of connections. Hence, the potential cost savings achievable with DSP are usually realized with in the mean comparable connection lengths and thus without suffering operational complications.

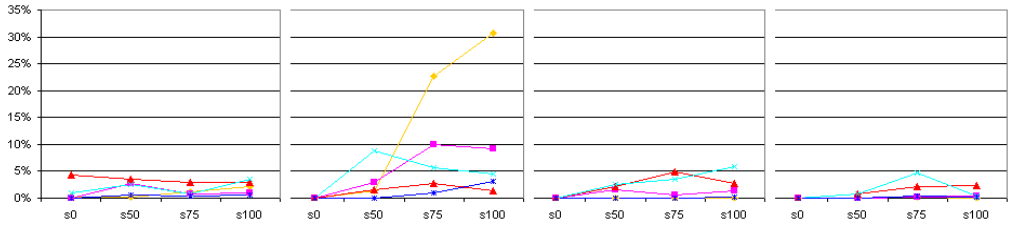
Solvability and decomposition loss. After comparing the network designs obtained by the different concepts strategically and operationally, we come back to previously discussed evaluations. At first, we consider solvability and the decomposition influence for the corresponding tasks. As in the previous section, Figure 5.12 visualizes for each instance and each concept the optimality gap and the decomposition loss as portions of both the total cost and the gap cost. The DSP values are



(a) *Optimality gaps.*



(b) *Absolute decomposition loss as portion of total cost.*



(c) *Relative decomposition loss as portion of gap cost.*

Figure 5.12: *Measures for solvability and decomposition sensitivity of the models for the alternative survivability concepts.*

repeated for comparable presentation in uniform scales.
The figures show some new effects, but also confirm previous observations. First of all, the network ranking regarding solvability of the instances is maintained by the solutions for all concepts. This certifies that none of the instances offers properties of dominating advantage for an individual concept, and thus the curves represent the relations of difficulty of the corresponding design problems. Basically, all concepts perform similar on the large scale, and in particular the plots of max-DSP and 2-DSP have a striking similarity. However, a closer look shows some clear differences. In fact, DSP yields the worst gap in 15 of the 20 instances, among these nearly all with protection. So, DSP is in tendency most difficult to solve, as claimed before. In view of the best qualities, the concept ranking is max-DSP (8 instances), followed by 2-DSP (6), DSP (4), and finally 1+1 path protection (2), which on the other hand never produces the worst quality. A clearly favored concept in terms of solvability can therefore not be determined.
These conclusions are underpinned by the various mean values over the compared concepts listed in Table 5.5. Except for s0, DSP shows the highest means in all scenarios, while the other concepts perform similar throughout the test set. In contrast

	DSP	max-DSP	2-DSP	1+1	mean
pt14	2,63%	2,41%	1,85%	2,39%	2,32%
pt17	6,23%	5,62%	5,72%	6,14%	5,92%
pt20	15,53%	13,31%	12,13%	13,59%	13,64%
pt28	9,43%	8,02%	8,96%	8,62%	8,75%
pt50	22,63%	18,24%	18,52%	20,40%	19,95%
mean	11,29%	9,52%	9,43%	10,23%	10,12%
s0	10,25%	11,00%	11,00%	11,03%	10,82%
s50	12,09%	11,42%	10,90%	10,99%	11,35%
s75	12,13%	8,11%	7,87%	9,68%	9,45%
s100	10,68%	7,54%	7,96%	9,20%	8,85%

Table 5.5: *Mean optimality gaps for the survivability concepts.*

to DSP, the comparison of means for different protection levels for the alternative concepts indicates growing qualities with increasing protection level over 50%. In total, our solution method achieves a mean optimality gap of 10.12% over all instances. As the DSP results with 11.29% are not deviating too much from this, design tasks using this concept can still be seen as well solvable by the presented approach.

Next, we turn to the worst case decomposition losses in solution quality. For the absolute values, overall less than 0.75% of the total costs result from the exclusion of regeneration and conversion issues in the dimensioning and routing optimization. Figure 5.12(b) indicates lowest affection for 1+1 path protection comparable to 2-DSP and marginally worse results for the other DSP variants. As main reason, the first two concepts allow for shorter connections and thus require less regenerator devices, with regenerator number means (over all corresponding instances) of 0.6 (1+1) and 1.35 (2-DSP) compared to 5.3 (DSP) and even 29.95 for max-DSP. Regarding converters, the concepts are affected in a restrained manner by the length effects. Here, the mean converter numbers are 9.4 (1+1), 12.65 (2-DSP), 16.7 (DSP), and 21.95 (max-DSP), apparently closer together than for regenerators. Nevertheless, these mean numbers are quite small compared to the total demand in the instances, and thus we can conclude that the decomposition approach proves effective for all considered concepts.

Putting decomposition losses into relation to the solution quality, a single peak for max-DSP in the heaviest loaded network pt14 catches the eye. This plot exposes the concept's weakness to require often overlong connections. In combination with high traffic volume, an extraordinary large number of regenerations becomes necessary and, as not involved in the optimization objective, increases the provable optimality gap considerably. This exceptional portion is further driven by the high solution quality achieved for the pt14 instances. In the other cases, however, a gap portion of 5% is rarely exceeded, independent of the concept or topology. From this, we conclude that for any of these scenarios, the decomposition provides an effective and suitable approach, too.

Finally, we just remark that the comparison of link vs. node cost portions gives for any concept nearly the same values as in the DSP case. These proportions have been displayed in Figure 5.9, and due to mean value deviations of not more than 0.65% for the other concepts, additional plots can be omitted. Interestingly, these similarities show that the different cost models used in the study do not favor any

of the survivability schemes concerning the decomposition loss.

5.2.3 Upgrade planning

In this case study, upgrade planning for optical networks is examined. With a predefined network configuration, the task is to find minimum cost extension for accommodating new traffic requirements. Since reference scenarios are typically not provided with preinstalled equipment and real-world networks of providers are highly confidential, we apply a special procedure to generate scenarios for an upgrade planning study.

Scenario generation. Most often, upgrade planning of networks is demanded in case of changed traffic forecasts derived from latest transmission measurements and longer period development experiences. Another reason for need of adaption can be varying traffic properties which require to resettle connections or entire commodities. Such a property, for instance, is the portion of protected traffic (which might jump up after customers recognize or suffer consequences of actual failure occurrences). When network capacities have already been occupied tightly, realization of such traffic changes, even if not varying the volume, can often be achieved only in combination with an extension of the configuration.
In the previous studies, we have computed designs based on graded survivability levels. Driven by the cost optimization, the obtained network configurations contain a very low amount of spare capacity. These networks therefore offer a suitable basis for upgrade planning when taking increasing reliability requirements into account. The following study again focuses on a single protection scheme, namely DSP as used in the reference case, and considers the upgrade of each but the full protected scenario to each higher level of survivability, denoted by sl-u with l as protection level of the originating design and u as new protection level to achieve.

Upgrade vs. greenfield planning costs. Since greenfield planning solutions for all survivability levels have also been computed as discussed previously, it is interesting to evaluate whether the designs obtained as upgrades differ fundamentally. Before comparing the total network costs as first indicator, we take a brief view on the solution qualities for upgrade planning. Analogous to the previous plots in Figures 5.4 and 5.12(a), Figure 5.13 displays the optimality gaps for each scenario as portion of the total costs. Note that this representation limits the gaps to at most 100%.

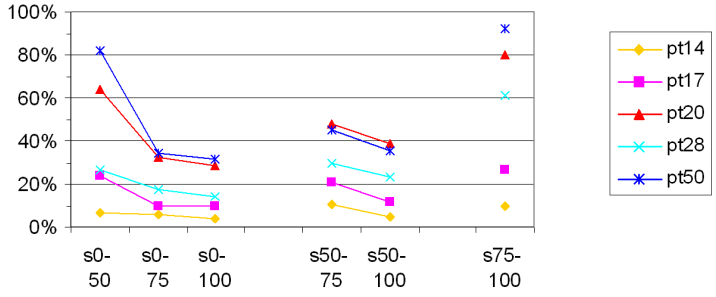


Figure 5.13: *Optimality gaps for the network design upgrades.*

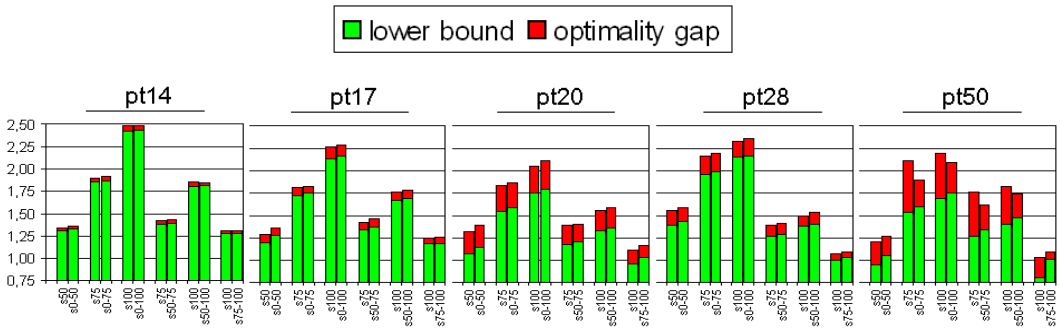
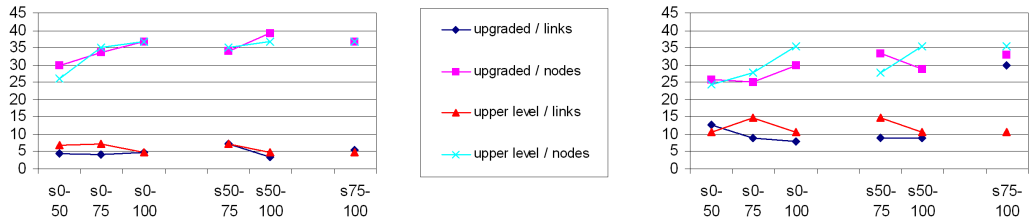


Figure 5.14: *Comparison of design costs and qualities, represented as before by the total cost split into lower bound and optimality gap, for greenfield planning as left bar and upgrade planning as right bar in each pair of side by side bars, where all values are normalized by the total cost of the upgrade's originating design.*

At first glance, the values are surprisingly high in comparison to the former solution qualities reported. In presence of existing capacities, the solution method seems to perform much worse than in case of greenfield planning. Such a (rash) conclusion, indeed, might be misleading, and in fact partially even the opposite statement holds. Before showing this in detail, a reason for the observed effect is not hard to identify. Let us for a moment fix the difference between a lower bound and an associated solution value. The higher the total cost is, the smaller the optimality gap will be, though corresponding to the same absolute difference. In turn, the gap size increases with decreasing solution value, and exactly this happens when offering many existing capacities. Note also that the gap sizes strictly decline with growing difference in the percentage of protected traffic in the originating and the targeted survivability level. This confirms the presented argument, as providing more protection needs to add more backup connections which consume further resources and this way drive higher total cost for additionally required capacities. For this, a strong indication is that the largest gaps occur for those upgrades where only few more backup connections in the higher level instance have been routed within the greenfield planning. The latter property holds in particular for s0-50 and s75-100. Hence, the larger optimality gaps need not be a sign of having found bad designs.
In fact, the upgraded designs turn out to be comparable or sometimes even better than the greenfield planned ones (for the targeted survivability level). The key observation is that the absolute cost values corresponding to the optimality gaps are of similar size in each scenario. For such a comparison, the total cost of the originating design has to be added to both the lower bound and the total cost of the solutions with upgrade planning. Figure 5.14 places these results, as usual split into lower bound and optimality gap costs, opposite to the greenfield planning design values. For each pair of side by side bars, all cost values have been normalized with the total cost of the originating design for the upgrading. This value corresponds to the '1' on the ordinate scale, and thus the bar part above one represents the upgrade solution lower bound and total cost similar to the previously applied manner.
The picture illustrates concisely the above statement. Both the total cost and the optimality gap range are typically only marginally shifted in comparison to the greenfield planned solutions, with the upgraded designs being at most 6.4% more expensive over all scenarios. In pt14 and pt50, upgrading turns even out to gen-



(a) *pt28.* (b) *pt50.*

Figure 5.15: *Mean numbers of unused node ports and link channels in upgraded designs and the upper level greenfield planned networks for two selected networks.*

erate better solutions in some cases, saving up to 10.6% of the greenfield planning total costs. For the largest topology, this is most probably due to the lower solution qualities achieved for the originating designs, and the savings in pt14 are not more than 0.4% and thus minor deviations. Nevertheless, provision of preinstalled devices makes it unnecessary to consider all module combinations for up to the capacity available for free and thus reduces the number of configurations which have to be taken into account. The corresponding complexity reduction can allow to solve the planning task to higher quality and this way yields better designs in the end. Since the cost comparison is based on including the originating configuration cost, occurrence of such an effect presumes that no preinstalled capacities are wasted, i.e., an existing network configuration can be fully exploited for the changed traffic, too, which is examined next.

Capacity utilization. When the lightpath configuration in a preconfigured network has to be adapted to changed traffic, decisions taken in the previous network design might turn out as disadvantageous. In this view, underequipped links and nodes are noncritical as upgrades are still possible, whereas oversized capacities are annoying for the waste of resources. Avoiding such a waste best possible is therefore desirable when optimizing network design.
In the studied instances, capacity utilization is of satisfactory degree in the majority of cases. Mostly, the average numbers of unused channels per link range around a value of 5, indicating an overall high occupation of the installed link capacities. Only for pt50, these values are between 10 and 15. At nodes, spare port numbers are naturally higher due to the link-oriented node dimensioning and a coarser granularity of installable capacities. Here, overdimensioning of typically between 20 and 35 ports occurs. Interestingly, both holds for upgraded designs as well in nearly all cases. Figure 5.15 plots the mean numbers of unused link channels and useless node switch ports for upgraded networks in comparison to the greenfield planned designs for the targeted protection level for two selected networks.
In pt28, the corresponding values match nearly perfectly and show that capacity utilization is quite good, independent of whether planned as upgrade or on the greenfield. Such close congruences, however, occur rarely, whereas the pt50 plots better reflect a typical behavior. Nevertheless, the amounts of unused capacity remain comparable for both planning types and, altogether, we can conclude that capacity utilization is at a satisfactory level.
These conclusions are also confirmed when coming back to capacity waste in case of upgrading. In fact, completely unused modules crop up in no more than four

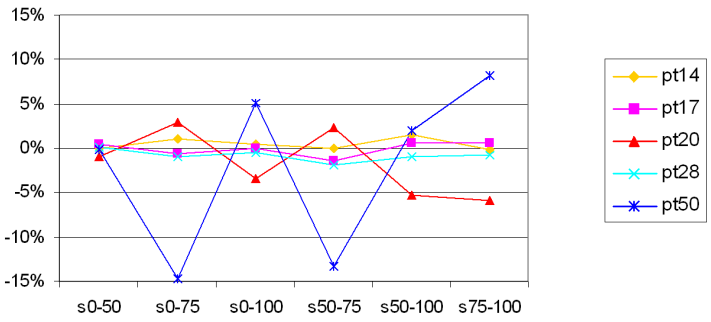
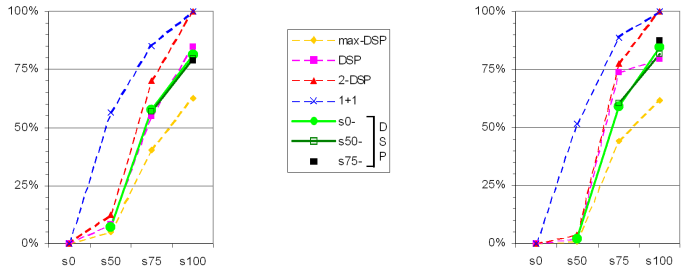


Figure 5.16: *Difference in the number of established connections for the upgraded network in comparison to the greenfield planned design of the targeted survivability level, related to the total demand.*

upgrade instances. Switches at nodes are never redundant, while two WDM systems (with the carrying fibers, of course) are not used in the s0-50 case for pt14, pt17, and pt50, and just a single case, s75-100 in pt50, breaks ranks with in total 30 obsolete WDM systems. The latter exception results most probably from the worse solution quality generally achieved for the largest topology. Overall, however, we can conclude that upgrade planning is quite well able to exploit freely given capacities in the generated scenarios. We remark that such a conclusion heavily depends on the studied instances, and we are aware of the possibility that our instance generation process, asking just for increased protection, certainly has an influence on the outcome. More comprehensive investigations on upgrade planning performance in terms of capacity utilization are therefore left to further research, being beyond the scope of this thesis, where we rather turn to other fundamental observations.

Total number of connections. The general concept DSP has the choice between a variety of possible routings for each commodity, differing in the number of connections to establish and their necessary diversification. For each commodity, the decision of which alternative configuration to establish is taken such that the final design over all routings comes to an as low as possible total cost. Presence of freely available capacities can clearly influence these decisions. For verification of such an influence, we consider the total number of connections as indicator for a basic change in routing selection. Figure 5.16 illustrates the difference in the total number of connections between the upgraded designs and the corresponding greenfield planned network configurations for the targeted survivability levels.
The results show that in most cases very similar routing decisions are taken. The difference in total connection numbers for pt14, pt17, and pt28 is less than 1.9% of the total demand. Only for pt20 and pt50, larger differences occur, and interestingly these changes show exactly opposite behavior. A possible explanation of these effects combines two properties of the greenfield planned designs, the achieved solution quality and the total connection number growth for increased survivability level.
The deflection sizes shown in Figure 5.16 for the instances correlate roughly with the solution quality ranking from Figure 5.4 on page 196. Where greenfield planned designs are proven to be (at least) close to optimality, the upgraded designs tend to become very similar in the DSP selected parameterization, as long as changed prerequisites do not enforce or favor a distinct solution. As a consequence, larger



(a) *pt20 instances.* (b) *pt50 instances.*

Figure 5.17: *Number of established backup paths as portion of the total demand in network upgrades for the topologies pt20 and pt50, extending the plots for greenfield planning solutions of the targeted survivability levels.*

variations are predominantly to be expected for scenarios with higher optimality gaps, which leave more room for improvements by alternative protection routings. These claims are also affirmed by a more detailed comparison of the number of established backup connections. As extension of the corresponding plots for different protection schemes in Figure 5.11 (upper row), Figure 5.17 illustrates these numbers for the upgraded designs.
In pt20, the added values do not deviate much from the connection numbers found in the greenfield planned designs. For pt14, pt17, and pt28, the corresponding plots behave similarly, being even closer to the values for DSP in Figure 5.11. For pt50, in contrast, a very different curve emerges, forming a much better balanced intermediate way between 2-DSP and max-DSP. From this, we conclude that network upgrades, due to their reduced complexity, allowed to identify more beneficial routings for these instances and so generated better final solutions than with greenfield planning for the targeted scenario. Note that this does not contradict to the alternating sense of the differences. In fact, the number of backup connections in s75 for pt50 occurs quite high and increases with considerably flattened slope to the s50 case with a surprisingly small backup connection number. Exactly these extremes are leveled out in all upgrade design solutions.

Node vs. link cost. Finally, we briefly remark that the relation of node cost vs. link cost again does not deviate considerably from the previously presented results. Though individual values are shifted of up to 6.36%, a graphical illustration has striking similarity with Figure 5.9 on the large scale. From this, we conclude that also presence of preinstalled equipment has no major influence on which of these cost portions is dominating, as long as no irregular originating configuration has been prescribed, e.g., having tightly occupied links in combination with far oversized node equipment.

5.2.4 Extended hardware model

This case study considers optical network designs optimized for an extended hardware model. The extension introduces few additional modules for the sake of comparability with the reference scenarios. We first describe the added devices and their properties before discussing the obtained results.

module type	module name	install at	capacity	cost pt17,pt20,pt50	cost pt14,pt28
WDM system	WDM20	links	20 wavelength channels	14.00	21.00
WDM system	WDM10	links	10 wavelength channels	8.00	12.00
switch	OXC256	nodes	256 ports (bidir.)	522.80	522.80
				reach: 1200 km	reach: 8000 km

Table 5.6: *New modules in the extended hardware model.*

Hardware extension. In general, any modification on the available variety of installable devices and their properties can influence network design results. A change of the hardware model has therefore to be chosen carefully in order to maintain comparability with reference results. In this study, we focus on scalability of transmission and switch capacities by a slight refinement of the module type sets.
We introduce two new WDM system types and one new OXC switch type, while keeping the entire previous hardware model without modifications. Table 5.6 lists the new modules which are added to those subsumed in Table 5.4 on page 192. The additional link equipment allows for installing capacities in finer granularities, with expectation to reduce spare resources in total. Thereby, the wavelength spectra are specified such that $\Lambda_{\texttt{WDM10}} \subset \Lambda_{\texttt{WDM20}} \subset \Lambda_{\texttt{WDM40}}$ holds. At the nodes, a larger switch type is inserted for providing further economies of scale. All costs extrapolate basic relations where larger modules provide regular discounts on twice the cost of half-sized modules, for the full cost in case of WDM systems, and for the base cost in case of OXCs. The consequences of these clearly arranged extensions are next studied in detail.

Costs and qualities. As in the previous studies, the optimization performance in terms of the objective, minimizing total costs, and the obtained solution qualities are primary evaluation issues. In the usual representation, Figure 5.18 illustrates the total costs, split into lower bound and optimality gap costs, for the reference instances on the left besides the results with extended hardware model on the right of each pair of bars. For each network, all values are normalized with the total cost of the unprotected reference scenario costs.
The plots show as expected that the extended hardware model enables to realize cost savings compared to the reference instances. The total cost reductions range between 3.8% and 14.7%, and the lower bounds decrease to a comparable extent as well. Once again, only s75 for pt50 gives a special case with significant increase of the optimality gap. For this instance, the design cost decrease due to the hardware extension is quite regular compared to the other instances, whereas the lower bound has reduced dramatically. Since experiments with longer running time have detected much stronger lower bounds, this indeed seems to be just an exceptional case.
Altogether, we conclude for this case study that the provided capabilities to save costs by installation of finer granularities (at graded costs) and economies of scale could have been well exploited for finding cheaper designs, while having nearly no impact on the quality performance of the optimization method.

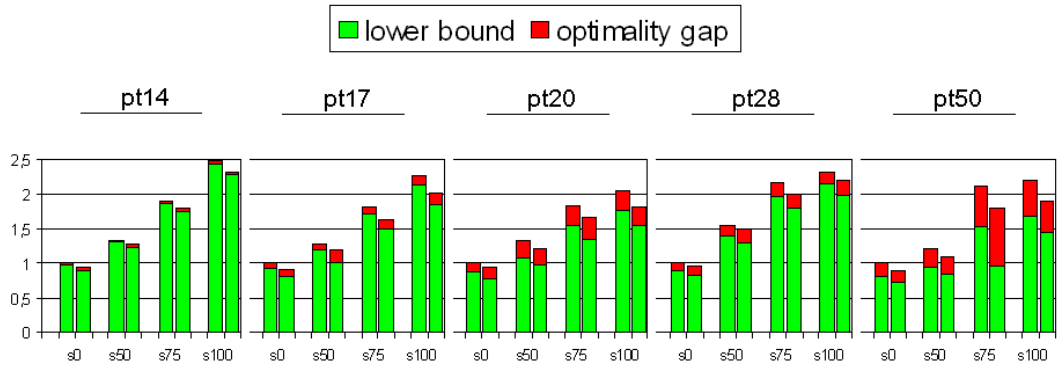


Figure 5.18: *Comparison of design costs and qualities in usual representation for the reference instances as left bar side by side to the corresponding extended hardware model design solution as right bar in each bar pair. Network-wise, all values are normalized by the total cost of the corresponding unprotected reference design.*

Link capacity utilization. Besides cost savings, the hardware extension is also expected to have an effect on the spare capacities in the designs. As mentioned in the previous study, the inherent oversizing of nodes due to link-oriented dimensioning generates indifferent spare capacities which depend only in second order on the traversing traffic. Therefore, we focus on links where these issues are much more closely related and the refined installation granularities let expect an observable effect. For this, Figure 5.19 plots the change in average spare channels per link for the instances of this study in comparison to the reference instances.
The figure reveals that the wide majority of instances in fact contains less spare channels on links in total. In few cases, slightly increased averages per link appear, which are evidently acceptable when finding cheaper designs in the end. Neglecting these exceptions, the average numbers of spare channels per link from the reference solutions decrease by a mean of about 50% when offering the refined module set. The largest rise in link utilization is achieved for pt50, where up to 10.5 less spare channels per link are observed. Notice that a spare channel reduction of one on 40 links is equivalent to saving of a complete `WDM40` system. Typically, the total number of installed link systems increases with the extended hardware, but the total number of provided link channels is often considerably smaller than for the reference instances.
Another perspective on transmission capacity utilization is provided by the portion

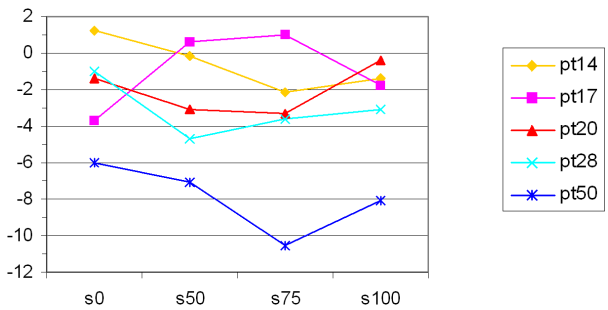
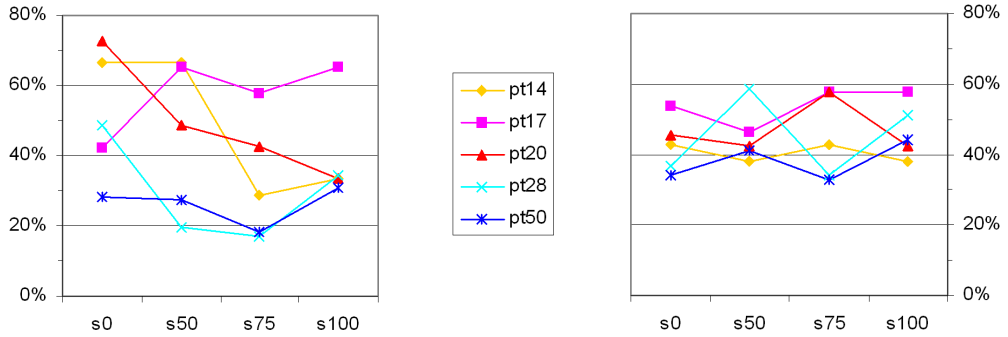


Figure 5.19: *Absolute difference in the number of average spare channels per link for the hardware extended instances in comparison to the reference instances.*



(a) *Reference instances.*

(b) *Instances with extended hardware.*

Figure 5.20: *Portion of links with fully utilized capacity, i.e., no spare channels, for (a) the reference instances and (b) the designs upon additional hardware.*

of optimally utilized links, i.e., the number of fully loaded links over the total number of links. Figure 5.20 plots these values for the reference and the hardware extended instances.
Intuition suggests that less total spare channels basically correlate also with larger portions of fully occupied links, forming a shift of the reference instance values. Surprisingly, the diagrams reveal another observation. The full link portions span a wide range from 17.1% to 72.7% in the reference instances. With extended hardware, the values are much closer to each other, concentrating around the mean of 44.9% which is just marginally higher than 42.4% for the left figure. The refined link system structure hence seems to yield a better balanced link capacity utilization.

Node vs. link cost. A short final remark again concerns the composition of total costs from link and node costs. Due to the hardware extension, these relations are again shifted only marginally in comparison to the reference instance splittings, as shown in Figure 5.9 on page 201, but do not change considerably on the large scale. In fact, all link cost portions increase by values between 0.6% and 5.2%, with a mean of 2.9%. From this, we conclude that the saving possibilities by the extended module set are slightly less favorable for links, where overdimensioning can be reduced by installation of finer granularities at the cost of the installation of more systems (and thus fibers), than at nodes just with an additional large switch offering economies of scale. Such a balance, however, depends predominantly on the specific modules and cost values assumed in the study and therefore does not state a result in general, but reflects a helpful analysis for particular settings planners in practice are faced with.

Summarizing, the expected effects by provision of a refined module set have been observed. Economies of scale and a finer capacity granularity yield cheaper network designs with less spare capacities and better balanced link utilization. Thereby, the performance of our solution method was unaffected by the instance variation. For wavelength assignment, however, inhomogeneous WDM systems offer different wavelength spectra and thus result in inhomogeneous wavelength availabilities on the links. Consequences on solvability and the number of required converters are discussed in Section 5.3 where we give a comprehensive report on evaluations for wavelength assignment.

5.2.5 Opaque scenario

As last study on varying network design settings, we consider optical networks based on opaque technology as alternative architecture. For this, another adaption of the hardware model is introduced.

Hardware model adaption. Opaque networks are characterized by the property that any optical connection undergoes an o-e-o-conversion at each traversed node. Since this function includes a possible exchange of the wavelength of operation, wavelength converters are not needed anymore and thus are not further involved explicitly.
The placement of transponders (performing the o-e-o conversion) can be based on different approaches: either individually on established connections only, or by means of fully equipped transmission or switch modules, i.e., placing a transponder at each port or transmission channel independent of its utilization in the current lightpath configuration. The first method clearly results in sparsest use of o-e-o conversions, but might become troublesome for lightpath reconfigurations to accommodate changed traffic requirements, even if no additional transmission or switching capacities are needed. Following our policy of designing networks flexible in preparation for (most) alternative routings, we rely on the second possibility. Moreover, guided by the applied kind of link-oriented node dimensioning, we consider full equipment of WDM systems rather than of switches. This way, any computed opaque network design still allows for any routing reconfiguration that does not exceed the installed link capacities.
The advantage of an opaque architecture consists of the possibility to employ electronic switching devices, introduced as DXCs in Chapter 1. Being widely used today, this switching technology is usually somewhat cheaper than optical switching, but is also bounded by physical limitations. Growing transmission bitrates offered by optics thus need DXC cascades to cope with a multiplied throughput, which might (over-) compensate the cost advantage. To evaluate such effects, we do not restrict on a particular bitrate with corresponding device prices as market snapshots (or estimations), but rather analyze break-even relations based on the formerly assumed OXC costs. For this, we leave the reference model switch modules and their prices unchanged in the case study and appraise discount factors on node equipment that would equal both architecture's total cost.
As a consequence, the hardware model is just adapted as follows. Each WDM system comes with a transponder at each offered transmission channel. As both transponders and transmission channels are bidirectional, the total number of transponders per WDM system equals its channel number and thus is predetermined. Therefore, the opaque hardware model holds just an increased WDM system cost of 104 ($= 24 + 40 \cdot 2$) in the networks pt17, pt20, and pt50 with smaller optical reach, and 156 ($= 36 + 40 \cdot 3$) in pt14 and pt28 with ultra-long haul transmissions. Installation of individual transponders does not occur anymore, making their explicit incorporation unnecessary as for converters. Hence, we end up with a reduced hardware model with shifted cost relations as basis for this case study.

Before discussing the results, a computation exception has to be remarked. The dimensioning and routing optimization with four hour time limit did not succeed

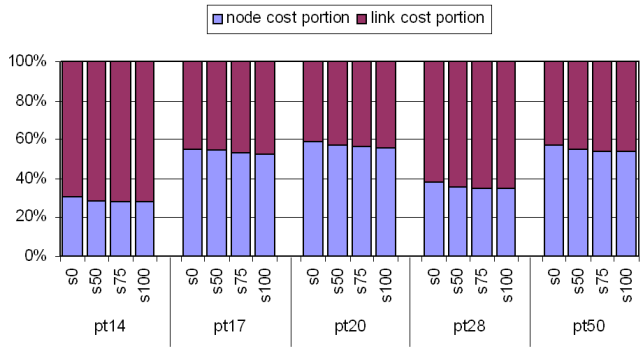


Figure 5.21: *Opaque network design total cost splitting into node and link equipment portions.*

to find feasible solutions for the opaque architecture in two cases, namely scenarios s75 and s100 for the largest topology pt50. So, we raised the time limit for these two instances to six hours of CPU time which turned out to suffice. The following report evaluates the finally generated network designs.

Node vs. link cost. The introduced shift in the module cost relations clearly carries over to the total cost portions for node and link equipment in the computed network designs. So, we first display these splits in Figure 5.21 which compare to those for the reference instances shown in Figure 5.9 on page 201.
Due to the hardware modifications, the enlarged link cost portions do not surprise. Now, link equipment expenses are at least roughly equal to that for nodes or even become the dominating part. Though the more expensive OXCs are still used as switches, the node cost portions from the reference instances reduce roughly by a percentage ranging between a fourth and a third, with a mean of 29.2%. Notice that replacing OXCs by DXCs saves just a portion of this part of the total costs.

Architecture costs. Turning to an architectural appraisal, Figure 5.22 presents the opaque optical network costs and the associated solution qualities in the usual way, repeating those for the transparent reference instances for providing better comparability.
The opaque designs show considerably higher total cost than the corresponding reference instances, which clearly results from the simply increased prices for some modules. As this modification varies only parameter values, but leaves the integer linear programming formulation otherwise unchanged, it does also not astonish that comparable solution qualities are achieved for both architectures. In fact, the optimality gaps for the opaque architecture are always smaller than for the corresponding reference case. This is basically reasoned by the opposite effect than in case of upgrade planning, where in comparison to the reference instances similar cost differences between lower bound and solution value yield much higher optimality gaps since being related to a much smaller solution value. However, no extraordinary gap differences occur, which would otherwise disturb an analysis of DXC cost limits for making the opaque architecture more favorable.
To compete with the transparent designs, DXCs have to provide sufficient cost advantages over OXCs. For this, Figure 5.23 displays the percentages of the opaque solution node costs at which break-even in total cost is reached.

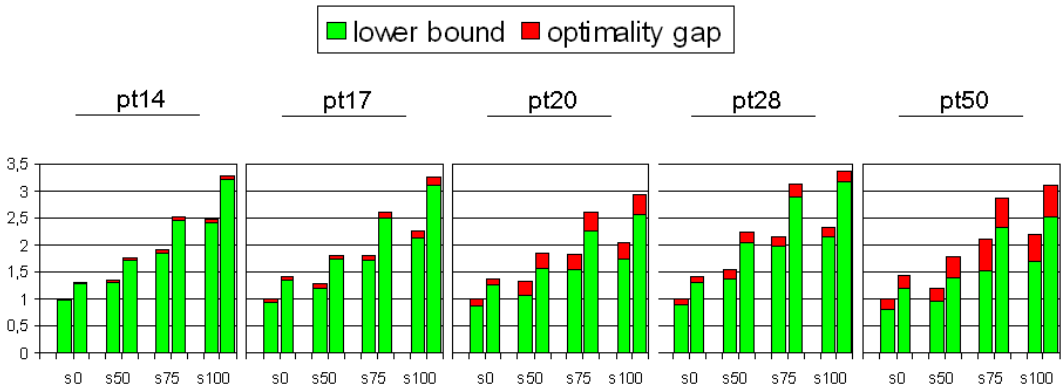
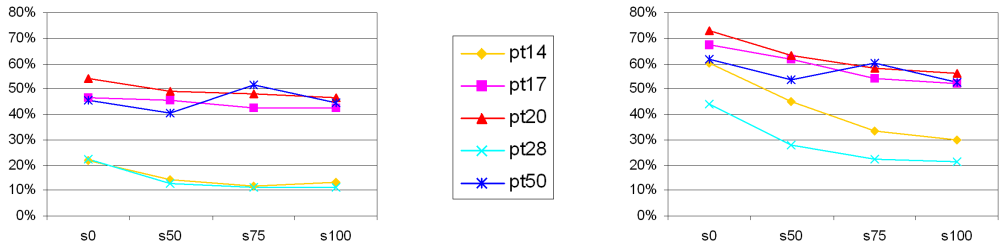


Figure 5.22: *Comparison of design costs and qualities in usual representation for the reference instances as left bar and the corresponding opaque architecture instance as right bar of each pair of side by side bars. For each network, all values are normalized by the total cost of the unprotected reference solution.*

When simply comparing the obtained solutions, the break-even portions are quite regular. In the small area networks pt17, pt20, and pt50 with shorter optical reach, a node cost reduction of around 50% suffices, i.e., the opaque design becomes favorable whenever DXCs are not more than approximately half as expensive as OXCs. In the ultra long-haul equipped networks pt14 and pt28, nodes are already of lower relevance for the total investments. Thus, a much lower portion of node costs, ranging between 10% and 20%, has to be achieved to even out the cost difference between the corresponding designs.
This comparison indeed is not complete, since transparent networks incur also fixed costs which have been left out of the optimization and, for better comparability of the method's outcomes, in the preceeding break-even analysis, too. These costs subsume equipment employed for the transformations from the accessing electronic client layer to the optical layer at the begin and back to the electronic layer at the end of each optical connection, as described in Chapter 1. In the opaque scenario, these functionalities are provided by the already involved transponders. When comparing transparent networks to such a different technological architecture, these fixed costs



(a) *Computed designs.*

(b) *With inclusion of transparent design fixed costs.*

Figure 5.23: *Portion of total node costs in the opaque instances marking the total cost break-even point to the transparent reference designs, for the computed solutions (on the left) and with additional inclusion of the fixed cost for the e-o and o-e transformations at the endnodes of each connection in the transparent reference designs (on the right).*

have to be taken into account as well and add a transponder cost for each working optical connections, but not for backup connections (which reuse those from disrupted connections in case of failures). For this, Figure 5.23(b) shows the break-even relations when the fixed costs are involved within the transparent designs. This adaption has two effects. First, the critical DXC cost percentages clearly rise, since a smaller design cost difference has to be equalized. And second, the break-even points are now decreasing with increasing protection level. As remarked above, the transparent design fixed costs do not depend on the protection requirements. Nevertheless, these requirements yield higher total costs and thus also higher differences between the architectures which have to be compensated by switch cost savings. As a consequence, the final break-even portions are not anymore (nearly) stable over the scenarios for a network, but depend also on the demanded network reliability.
Summarizing, an opaque architecture only pays off if the available node equipment is sufficiently cheap compared to OXCs, in particular for networks covering wide areas. The derived limits have no general validity and depend highly on the module price assumptions. However, such a break-even analysis is only possible if technological alternatives can be evaluated by comparable methods and can help planners to decide whether the one or the other architecture is favorable.

Similar to an examination of break-even opacity costs, the expenses dedicated to transparency in optical networks can as well be analyzed and compared to those for alternative architectures. For such a study, the relevant costs vary by the required numbers of regenerators and wavelength converters, as already considered for rating the decomposition loss of our solution methodology. Thereby, wavelength assignments have been computed with short heuristic runs to demonstrate practical applicability. Though the results obtained this way are already satisfying in terms of sparse conversion requirements, potential further improvements and a deeper analysis of this mathematically interesting problem are worth a deeper evaluation, as carried out next.

5.3 Computations for wavelength assignment

In this section, we report on the second computational study concerning the wavelength assignment subproblem. Basically, we reuse all but the opaque network design instances from the previous study, i.e., all results for transparent optical networks. For these 130 instances, solutions have been completed with initial assignments generated by short heuristic runs. We begin with a summary on these assignment results which already include a considerable number of converter-free and thus optimal solutions. Since (too) few challenging wavelength assignment instances are left for further evaluations, we extend the test set in the way described next. In total, we end up with 634 Mcwap instances (and solutions) of which 323 lack of an optimal assignment or its proof of optimality.
For the resulting instance set, we first report on the lower bounds on the number of converters computed by the methods described in Section 4.4.2. Such bounds indicate possibly still improvable assignments for which we next present results by application of the improvement heuristic with wavelength extraction from Section 4.4.1. This finally leaves only a fraction of all instances unsolved. A report on the

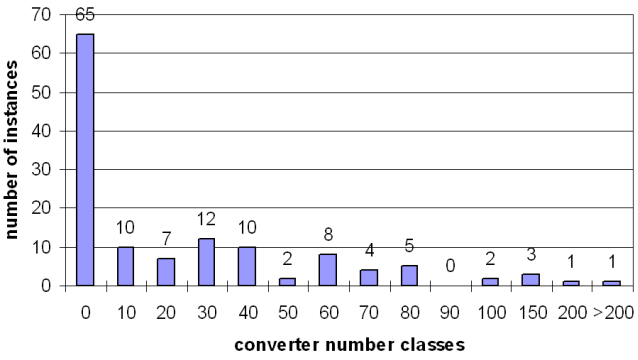


Figure 5.24: *Histogram of solution numbers from the previous computational study in Section 5.2 with converter numbers suitably classified in steps of 10 resp. 50 converters.*

outcome of the exact method for selected cases and possible directions for further improvements conclude the study.

5.3.1 Design computation results

In the first computational study, totally 130 transparent optical network designs have been computed and evaluated. Each of these solutions includes a feasible wavelength assignment obtained from an iterative heuristic run for at most 15 minutes. At first, we take a closer look on the converter numbers in these solutions.

Converter numbers. Figure 5.24 shows a histogram on the converter numbers in these assignments, each sorted into that class whose associated number is the smallest one not exceeded by its converter number (the particular value for each individual instance can be found in Appendix B).
The histogram reveals that for 65 and thus exactly half of all instances a converter-free solution has been found. Interestingly, this set contains all instances for pt14, all but one for pt17, and ten of the 15 involved s0 instances for the other networks, hence a strong incidence of smaller networks and the unprotected scenario. For a possible explanation, note that both of these properties, a low link number and no protection restrictions, facilitate an establishment of short connections with respect to the hop length. Such tendencies are also observable in the mean hop lengths plot of Figure 5.11(b) from the survivability model evaluation, involving 80 of the considered instances. This gives an argument in combination with the natural expectation that finding converter-free wavelength assignments is more probable for connections with shorter hop-length, also promoted by results like Corollary 4.12 or Theorem 4.13 which identify classes of instances with the guarantee of a converter-free solution, in particular due to having only lightpaths of hop-length at most two.
Regarding the 65 non-zero solutions, a dominance of relatively small converter numbers can be observed in Figure 5.24, and only few outliers appear. In fact, 60% (or absolute 39) of these instances contain at most 40 converters, while there are only 7 instances with more than 80 converters. The maximal class of size 12 refers to assignments with a converter number between 21 and 30, where also the average of 22.8 converters per assignment over all 130 instances lies. As further observation,

the designs based on the extended hardware model from Section 5.2.4 have a more than doubled average of 50.2 converters per assignment, and these only 20 instances include already 7 of 12 solutions with more than 60 converters. This indicates that the use of inhomogeneous WDM systems indeed tends to increase the need for converters.
When summing up values over all instances, the solutions in total contain 269,448 lightpaths and employ 2,961 converters, i.e., approximately one converter per 100 lightpaths. Focusing on those instances with converters, 130,679 lightpaths remain, and thus roughly one converter occurs per 50 lightpaths. Hence, we can conclude that the number of converters, even in the fast heuristically generated assignments, overall scales very well with the amount of traffic to route, being a further affirmation for the suitability of the proposed decomposition approach.

5.3.2 Instance set extension

Among the design results, half of the 130 solutions come with a converter-free and thus already optimal solution. As this leaves only 65 instances for further examination, we extend the test set by adding up to four alternative solutions for each design instance. We first explain how this extension is carried out, before we discuss the enlarged instance set and its properties.

Generation of alternative solutions. The proposed solution method for optical network design is not restricted to deliver a single solution only. In fact, the dimensioning and routing optimization returns the best found solutions in a pool of adjustable maximum size. Clearly, the postprocessing can be applied to any of these interim solutions, generating alternative designs for the original problem instance.
In our computations, the pool size parameter was in fact set to a value of five. For eight instances, however, less than five solutions have been found within the time limit of 4 hours for dimensioning and routing. Not surprising, all these instances belong to the largest network pt50. Here, just a single solution is obtained for the DSP reference s100 instance as well as the DSP upgrade planning s0-50 instance, three solutions for the DSP reference s75 instance as well as the DSP upgrade planning s0-75 instance, and four different solutions have been found for the 1+1 path protection instances with s75 and s100 as well as the DSP upgrade planning s0-100 and s75-100 instances. So, 16 solutions are missing of the potential maximum of 650, and thus we end up with a total of 634 different designs, each with an individual routing and initial assignment, as base of the extended test set for Mcwap evaluations.

Design improvements. Before we go more into detail on the initial assignments in these solutions, a remarkable observation is to be mentioned. All results evaluated so far refer always to the design generated from the best Discnet solution for the sake of providing a best possible comparability. During these computations, the other interim solutions of minor quality have been just stored and completed separately afterwards.
Surprisingly, the alternative designs not seldom yield a better solution in the end. In fact, for totally 31 of the 130 considered instances, the final minimum cost design is not obtained from the best dimensioning and routing solution, but in 14 cases

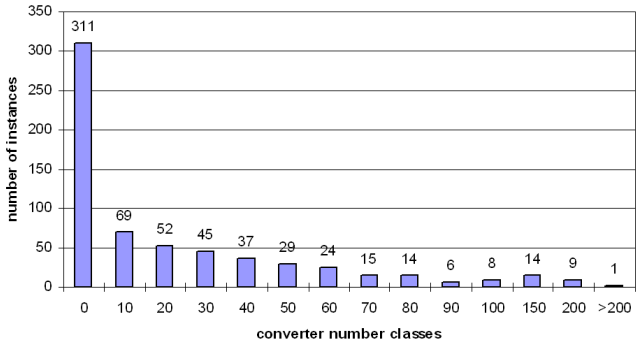


Figure 5.25: *Histogram of solution numbers in the extended set of designs reusing the converter number classification from Figure 5.24.*

from the second best, in eight cases from the third best, again in eight cases from the fourth best, and once even for the fifth best one. The decreasing multiplicities indicate that such an overtaking postprocessing performance cannot be expected for arbitrarily low ranked alternative solutions of dimensioning and routing. Moreover, the additional savings in the detected better solutions do not always result from a smaller number of employed converters and regenerators, which reduces in only 17 cases, remains unchanged in five cases, and even grows in nine cases. The routing improvement procedure described in Section 3.3.3.1 in fact contributes to the final design quality by detecting reroutings which allow for capacity reductions. For this, larger networks seem to offer more potential, as 21 advancements occur in the two largest networks pt28 and pt50. Regarding the distribution of the improved instances over the considered scenarios, no striking concentration appears.
In the 31 successful cases, the design costs further reduce on average by 0.65% with a maximum of 1.88% in relation to the primary solution, and the optimality gaps shrink on average by an absolute value of 0.51% with a maximum of 1.54%, which corresponds to a relative gap closing by 3.01% on average and 8.43% maximally. As a more demonstrative example, the mean optimality gap of 10.12% over all 80 instances from the survivability model evaluations, as reported in Table 5.5 on page 209, decreases to 9.97% when the primary solutions are replaced by the best alternative ones for the 21 instances of this set where such an improvement was found.
These observations show that the postprocessing procedures are in fact capable to enhance dimensioning and routing solutions for the final designs. In addition, we conclude that the proposed approach for optical network design optimization can profit from taking more than just the best dimensioning and routing solution into account for completion. Though not substantial, further solution quality improvements can be achieved this way.

Converter numbers in the extended solution set. Preparing the following study on McWAP, we finally evaluate the converter numbers in the extended set of 634 initial assignments. As before, these numbers are categorized for the histogram in Figure 5.25 which shows how many instances fall into any of the categories.
The plot reveals basic relations that are similar to those for the primary designs. Again, roughly 50% of the assignments are converter-free and thus optimal solutions. Among these, all but 4 solutions for the two smaller networks pt14 and pt17 as well as still 37 of the 75 unprotected designs for the other networks are present, contin-

uing the observed concentration for solutions with on average shorter lightpaths.
In the remaining 323 designs with converters, an exponential decrease in the number of solutions with higher conversion requirements comes to light in Figure 5.25. Note that the last three converter number classes are larger than the previous ones. This appearing regularity contributes further to the expectation of rare cases which need outstanding conversion effort, as motivated for the decomposition approach. From all solutions needing conversions, now even more than 80% contain not more than 40 converters, corresponding to a slightly lower average of 20.5 converters per design over all 634 instances.
In total 12,992 converters are placed on 1,423,480 routed lightpaths in the entire test set. Interestingly, already 4,556 converters are needed just for the 100 instances using extended hardware, from which only all those for pt14 and pt17 and an unprotected pt20 instance are converter-free. The remaining 59 solutions for this scenario thus contain an average of 77.2 converters, a strong indication that installation of inhomogeneous WDM systems drives larger need for conversions or makes it just more difficult to find solutions with low number of converters.

As a consequence, the test set extension brings up a considerable number of designs with an initial assignment of unknown quality. For these 323 designs, the initial number of converters serves as 100% gap correspondence when discussing quality improvements achieved by computation of lower bounds and attempts for solution improvements. In the natural order, we first determine lower bounds in order to see whether further proofs of optimality for solutions with converters are directly found this way and to identify the remaining designs with potentially improvable assignments.

5.3.3 Lower bounds

In Section 4.4.2, we present two approaches to obtain lower bounds for Mcwap: using the linear relaxation value of the dominating path packing formulation (4.4), and the combinatorial approach from Koster [91], adapted for non-uniform fiber spectra in Section 4.4.2.2. Both methods have been applied to all 323 instances with a so far non-zero optimality gap for the wavelength assignment. The results are discussed subsequently, beginning with the LP based approach.
In the following, we discuss the outcome of the computations. Thereby, we use some abbreviations to refer to networks, scenario settings, and solution indices for simple notation of individual designs. As familiar, the networks are denoted by their topology identifier and survivability levels as before by s0, s50, s75, and s100 as well as the introduced combinations used for upgrade planning. Moreover, the applied survivability schemes are indicated by 'max' for max-DSP, 'two' for 2-DSP, 'dsp' for the general DSP scheme, and by '1+1' for 1+1 path protection. Next, we differ the usual (reference) hardware setting marked by 'w1' and the extended hardware setting by 'w3', corresponding to the number of WDM system types involved. And finally, the up to five solutions for each instance are numbered sol1,sol2,... according to increasing cost of the solution obtained from Discnet.

Lower bounds by the LP optimum. For solving the linear relaxation of the path packing formulation by column generation most efficiently, several strategies

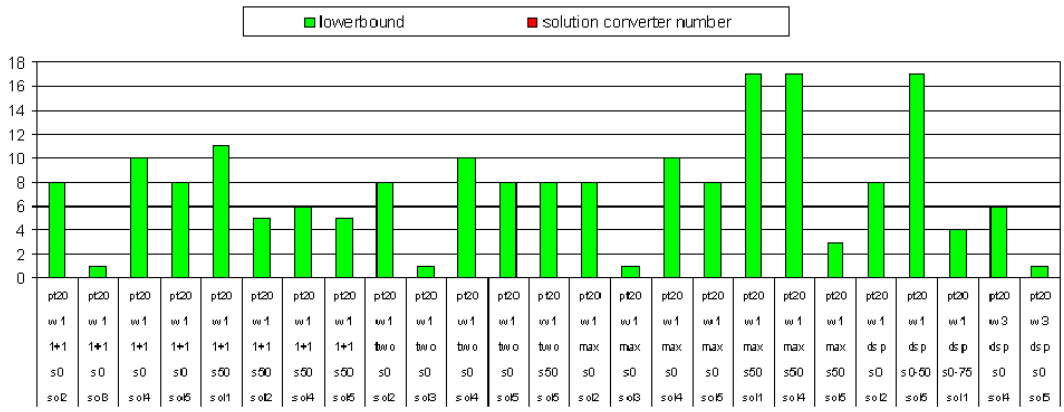


Figure 5.26: *Converter numbers for all instances whose optimality is directly proven by the LP optimum lower bound.*

are evaluated in Section 4.4.2.1. In this computational study, we exclusively use the strategies which have turned out to be most promising, an initialization with the basic column and application of restricted pricing as long as possible.
Interestingly, all LP relaxations of the path packing formulation for the 323 instances have an integral optimum value. This, however, does not reflect a general property (cf. page 238). In 197 cases, the LP value is zero and thus does not contribute to close the optimality gap, whereas the remaining 126 LPs return a valuable lower bound. In fact, a proof of optimality for the solution at hand is directly achieved in 25 cases this way. For this instance set, Figure 5.26 visualizes the computed lower bounds equal to the solution converter numbers.
Inspecting the particular instances, we observe that all directly proven optimal assignments occur in the same topology pt20. Even more, 119 of all non-zero lower bounds are obtained for this network scenario, while only 6 for pt28 and a single non-zero lower bound for pt50 are found. As a possible explanation for this striking concentration, pt20 excels by a typically higher link utilization and larger portion of fully occupied links compared to pt28 and pt50. For some cases already depicted in Figure 5.20, a comparison of all sol1 solutions in this study reveals that the average portion of completely utilized links is 48.95% for pt20, 42.03% for pt28, and just 27.93% for pt50, whereas the average number of spare channels per link increases from 4.29 for pt20 to 5.12 for pt28 and even up to 10.42 for pt50. So, wavelength multiplicities tighter to the wavelength requirements seem to facilitate identification of unavoidable converters. A further contributing factor could be the fact that pt20 holds the least total traffic amount among the networks, and note that most instances in Figure 5.26 have none or low-level survivability requirements, keeping the total lightpath number low as well. Establishment of more connections needs increased link capacities offering larger wavelength multiplicities which blow up the solution space of feasible assignments and make unsolvable conflicts harder to detect.
A next observation is that the unprotected s0 designs have the same lower bounds for '1+1', 'max', and 'two', but different ones for 'dsp'. For the first three schemes, this follows simply from the fact that the models are identically parameterized in case of no protection, and thus the same solution is found, though the results have been distinguished due to slightly different dimensioning and routing lower bounds.

In contrast, 'dsp' is based on a different routing model with further freedom degrees, and the distinct lower bounds show that obviously other designs are found this way though the previous ones are feasible, too. As soon as protection is requested, all schemes generate individual models and solutions for which varying lower bounds occur.
Moreover, Figure 5.26 reveals also that the obtained lower bounds are not restricted to very small values only. In context of coloring-like problems, non-trivial lower bounds are typically hard to gain at all (see, for instance, bounds on interference in frequency assignment reported in Eisenblätter [42] and Koster [90]). In case of Mcwap, however, optimality for solutions with up to 17 converters has been proven, and lower bounds up to a value of 51 unavoidable converters have been found. Figure 5.27 gives a histogram on the lower bound values for all 323 instances, reusing the previously applied categorization.
As expected in relation to the converter numbers displayed in Figure 5.25, the category sizes decrease with increasing lower bound values in a similar way. Among the non-zero results, nearly half of the instances have a lower bound of more than ten, with an average value of 14.29 over all 126 cases. This already indicates that the determined lower bounds do not only serve to identify instances without converter-free assignments, but indeed give valuable information to guide further improvement attempts, except for already proven optimal solutions, of course.
For the remaining 101 instances, Figure 5.28 illustrates the progress in terms of solution quality achieved by the lower bound values. Similar to the design cost evaluations in the previous section, the diagram bars reflect the converter numbers in the initial solution assignments, split into the lower bound value at the bottom and the resulting optimality gap at the top of each bar.
Since proven optimal solutions are excluded from this overview, each instance has a strictly positive optimality gap. The gap sizes indeed vary much, ranging in absolute values between one and 74 converters with an average of 16.38. Apparently, a concentration of larger differences occurs for the 19 'w3' instances with extended hardware model, for which the average absolute gap size raises to a value of 26.89. This increase can only by explained by a worse quality of the lower bounds or of the solutions (or both) for this scenario. In view of the path packing formulation, however, non-uniform wavelength spectra seem unlikely to worsen the resulting bound quality, and in fact the average lower bound is 15.53 and thus quite comparable to the average of 15.96 over all considered instances. Hence, we conjecture that the increased gaps are caused by less good solutions found for these instances, which

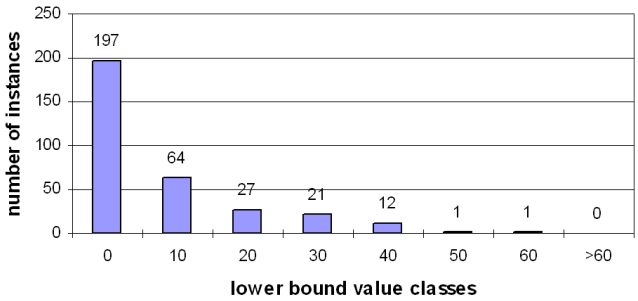


Figure 5.27: *Histogram of the lower bounds obtained as optimum value for the linear relaxation of the path packing formulation.*

corresponds to the previous observation of typically much higher converter numbers in the initial assignments for instances with inhomogeneous WDM systems.
A similar, but less obvious observation concerns the gap distribution over different survivability levels. Such an evaluation discloses that the average absolute gap of 16.38 converters splits into a value of 20.01 for the 67 instances with the two higher and of only 9.21 for the 34 instances with the two lower protection levels. Hence, the gaps for designs providing no or half protection are overall considerably smaller than for those with the two higher portions of protected traffic. But in this comparison, the lower bound qualities differ as well. While a slightly decreased average lower bound of absolutely 14.13 converters occurs for the solutions designed for predominantly survivable traffic, all instances protecting at most half of the connections find lower bounds with an absolute average of 19.56 converters which is quite strong compared to the overall average of 15.96. From this, we infer according to previous indications that lower survivability levels and thus lower numbers of lightpaths established in a less diversified way allow to expect both better lower bound and

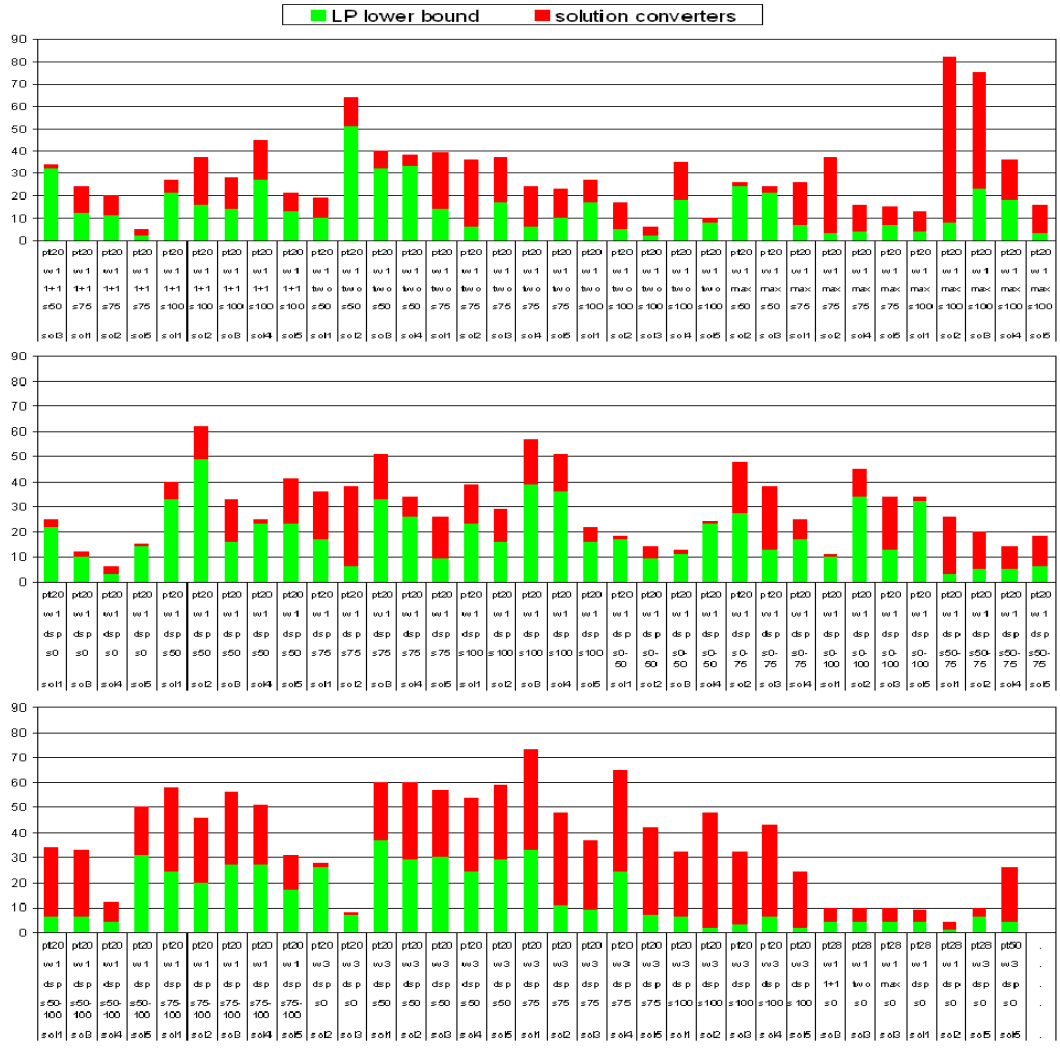


Figure 5.28: *Lower bounds and solution converter numbers for all instances with non-zero LP optimum.*

prot.	# y-variables			# priced x-variables		
level	pt20	pt28	pt50	pt20	pt28	pt50
s0	885	4,452	7,636	151	189	430
s50	2,449	12,927	19,291	236	346	713
s75	3,198	13,324	23,851	383	336	665
s100	2,947	13,442	32,611	424	380	901

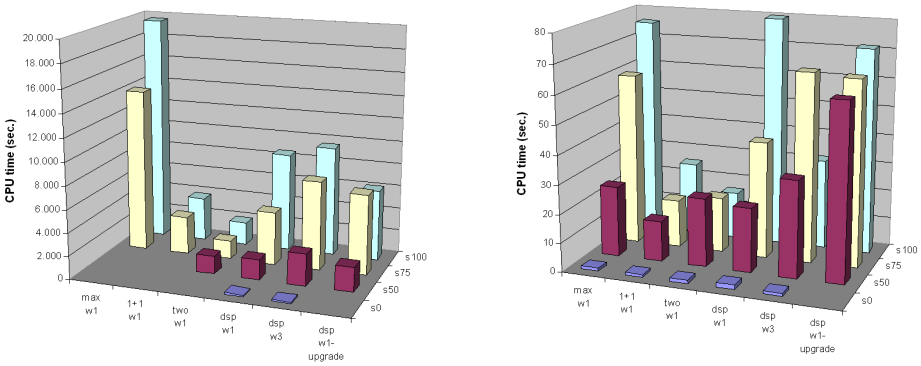
Table 5.7: *Examples for the number of y-variables and the number of priced x-variables for the* sol1 Mcwap *instances of the* DSP *scenario (except for s0 in pt50 using* sol4 *since* sol1 *has a converter-free assignment).*

solution qualities for Mcwap.
Altogether, the plots in Figure 5.28 exhibit also that the lower bound values are not seldom close to the solution value, even for higher converter numbers and throughout the test set. Moreover, though larger differences seem to dominate, the lower bound portion is on average 50.45% of the solution converter number over all depicted 101 instances. So, whenever non-zero lower bounds are found, we can conclude that on average a considerable optimality gap closing is achieved.

Column generation performance. After the quality evaluation of the lower bounds, we next turn to the performance for their computation by solving linear programs using the proposed column generation algorithm. For such measures, however, capabilities of the applied linear programming solver play a dominant role, and thus we restrict to some most significant facts. Thereby, we count with mean iteration numbers in the w3 scenarios having up to three pricing problems due to the resulting spectrum subsets (in fact, a single instance does not employ any `WDM20` system and thus has only two wavelength subsets to distinguish), whereas the computation times are, in contrast, always summed up.
As expected, each linear program with zero optimum is solved by exclusive use of restricted pricing. This does not surprise since the pricing in the restricted variant generates exactly those columns (or lightpath packings) that are needed for composition of a converter-free solution. The corresponding LP solutions are clearly integral in all y-variables, but typically contain many x-variables with non-integral values and therefore do unfortunately not encode feasible (integer) wavelength assignments. A less expected observation is that all these computations terminate without a single exact solving of the pricing problem, instead using the objective value as final stop criterion. As a consequence, new columns are priced out quite efficiently, and up to few hundred iterations (see Table 5.7 for typical numbers) always suffice to reach the zero optimum value, consuming on average a total time portion of only 12.15%. The lion's share of time is indeed spent for resolving the path packing linear program after each iteration which typically becomes more and more challenging with growing number of columns and when approaching the optimum value. In addition, the LP solvability depends heavily on the network size, best illustrated by the average total running time per instance of 9.02 sec. for pt17, 14.68 sec. for pt20, 593.83 sec. for pt28, and even 5,994.35 sec. for pt50. This results from the exploding variety of different routing paths (and thus partial paths, too) in meshed networks of increasing node and link number, blowing up the LP size as well. Table 5.7 lists some exemplary numbers of y-variables confirming this growth. However, the average LP solving times per iteration are usually (much) lower for the

instances with non-zero optimum, and so we conclude that reoptimizing the LPs is in particular difficult near zero as final objective value. As this occurs mostly for the large networks, it is not surprising that a portion of more than 96% of all total running times together is spent for the 197 zero-bound determinations, and over 85% for pt50 instances only. This gives a good figure for ranking the overall performance of the column generation algorithm, having most to struggle with the largest network scenario which in fact is of challenging size, whereas the other practical instances are solved quite efficiently, in particular in case of detecting a non-zero lower bound. Whenever a strictly positive LP optimum is found, at least two exact pricing steps are carried out in each computation, one for finishing the restricted pricing mode and one for the final confirmation of optimality. Theoretically, restricted pricing can suffice for these instances, too, but non-zero optimal solutions are rarely composed of such restricted path packings only, and we observe that at least one improving non-restricted path packing is always generated. Nevertheless, restricted pricing succeeds on average in over 60% of the iterations which fastens the generation of new columns. Moreover, more than two obligatory exact pricing steps are not often carried out, with up to totally ten such steps in 16 and more than ten steps in only nine of the 126 instances. Since solving the integer linear program of the pricing problem exactly takes clearly more time than a heuristical step, the portion of the average running time for pricings increases to 50.57% over these instances and thus nearly equals the LP resolving times. This balance indeed shows that the additional pricing effort does typically not dominate the performance. In fact, while most exact steps terminate rather quick, we find just three exceptional cases with quite hard integer pricing programs, one for the pt20 reference instance with w1 and DSP for full protection s100, and two for pt50 upgrade planning s75-100 using w1 and DSP as well. In these computations, the majority of pricings finishes regularly fast, but a couple of times the integer linear program needs up to around 15 minutes to be solved exactly, raising the portion of the pricing time to over 98%. In order to make the lower bound determination more efficient for such rare cases as well, further research on multi-set packings for speeding up the exact solution process is needed. At last, we analyze the LP optimization effort dependent on the particular design scenario. For better comparability, we select in either case, among the instances with zero and those with non-zero optimum, the most dominant network in view of the sums of total running time, being pt50 and pt20, respectively. In case of pt20, we also exclude the three highlighted instances with extraordinary exact pricing effort to obtain a scale which makes meaningful comparisons for the more regular cases possible. So, for in total 77 pt50 and 116 pt20 instances, Figure 5.29 illustrates the average CPU time in seconds per instance separately for different planning settings and protection levels (or targeted levels in case of upgrade planning).

Though the time scales differ by magnitudes, a comparison reveals similar relations in both plots. For s0, the computation times are negligible, and the effort typically grows with increasing protection level. This is explained by the fact that requesting higher survivability imposes an establishment of more (different) lightpaths in total, and the corresponding LP size increase prolongs the optimization duration. In view of the planning settings, we observe that 'max' instances are typically among the most time consuming ones, while those applying 'two' or '1+1' are solved fastest on average, and the 'dsp' are in between. Here, a similar argument applies, based on the lightpath hop-lengths which are largest in 'max' instances and smallest in scenarios



(a) *pt50 with zero bounds.* (b) *pt20 with non-zero bounds.*

Figure 5.29: *Average CPU times per instance for different scenarios and protection levels, restricted to the most dominating network among the instances (a) with zero bounds (pt50), and (b) non-zero bounds (pt20 without three exceptional cases).*

'two' and '1+1', see also Figure 5.11. The longer lightpaths are, the more partial paths are contained, generating larger LPs as well. Further, the interdependencies increase for longer lightpaths. Finally, the average running times for the 'dsp' scenarios used as reference, with extended hardware, and for upgrade planning show less regular performance relations. The instances with extended hardware hold in parallel up to three pricing problems to be solved in each iteration. Often, less iterations than for the corresponding reference instances are required, but a larger total number of columns is generated, which explains the higher running times. Surprisingly, non-zero bound instances with full protection perform very well, though exact pricings have to be carried out, too. In these instances, the corresponding column set subdivision possibly offers better opportunities for finding compatible column combinations. By upgrade planning, lightpaths can follow unexpected (long) routes selected for exploiting some free capacities. Such structural properties can influence the LP solvability, too, and are probably responsible for the observed deviations.
As concluding summary, the solution times for the path packing LP relaxation turn out to be highly sensitive against both the optimum value, whether being zero or strictly positive, and the LP size, driven by the problem size and structure. This holds in particular for resolving the LP after each iteration. In rare cases, exact pricing by solving of integer linear programs can become time-consuming, too, but typically the column generation procedure delivers quite fast, and many successful restricted pricing steps contribute also to limit the pricing effort. Altogether, determination of the path packing formulation lower bound remains challenging for the largest network, but succeeds most often in reasonable time for all other instances.

Combinatorial lower bound. The approach by Koster [91] presented in Section 4.4.2.2 provides a combinatorial alternative to obtain lower bounds for Mcwap. Focusing on any node as center of the surrounding star subnetwork, a transformation of the corresponding Mcwap subinstance to a special edge coloring problem allows to interpret an estimation on the number of uncolorable edges as lower bound for the number of unavoidable converters in the central node. As a major advantage, these estimations can be carried out very quickly, returning the lower bound B in far less than a second for any of the test set instances. Due to this fast response,

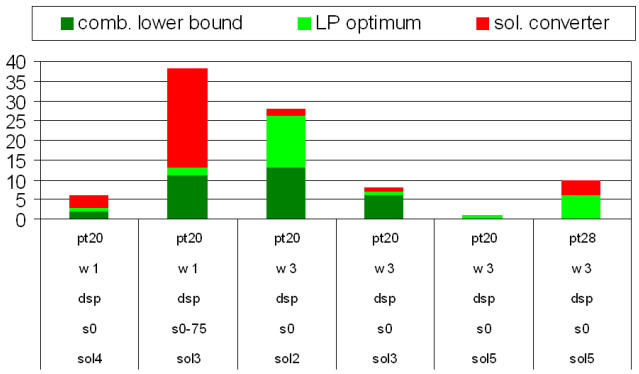


Figure 5.30: *Converter number from the solution, the LP lower bound, and the combinatorial lower bound for all instances where the lower bounds differ.*

the method qualifies in particular for integrated application within any method that can benefit from the delivered information.
Regarding the strength of the combinatorial lower bound, preliminary evaluations in Koster [91] on a small set of practical instances show no differences to the LP-based lower bounds. Theoretically, the dominance of the path packing formulation approach is diagnosed by Theorem 4.20 on page 174, and in the present study, first practical instances show this property, too.
Clearly, any Mcwap instance with a zero lower bound as LP optimum will also have a zero combinatorial lower bound, so we can restrict the comparison to the set of designs with positive LP optimum, which are listed in Figures 5.26 and 5.28. Among these 126 instances, a still surprisingly high number of 120 instances yield the same positive bound values by both approaches. The differences in the remaining six cases are illustrated in Figure 5.30, represented in a stacked way between the combinatorial bound value and the optimality gap filling up to the solution converter number.
This comparing plot visualizes that various relations in terms of small and large differences between both bounds and the solution value occur, though only a handful of instances is considered. For example, very similar bounds occur in the second and fourth bar, but the gap to the solution is once small, once large. The third bar shows an LP bound very close to the solution while the combinatorial bound is much worse. Further, both bounds do also not correlate in being positive or not. The two most right bars reflect a zero combinatorial bound where the LP optimum is non-zero, even reveals a solution's optimality in one of these cases. Such irregular relations avoid clear statements on the occurring differences and potential reasons.
Nevertheless, two conclusions can be drawn for a qualitative bound comparison. First, the combinatorial lower bound turns out to be equivalent in the wide majority of cases, therefore proves in fact to suit well for a quick bound estimation and could be applied as (fast responding) subroutine of other methods. Second, the occurrence of bound differences not only for artificial, but also for practical instances demonstrates that computation of the LP-based lower bound is to prefer whenever a best possible quality ranking is required.

Terminating lightpaths. The observed influence of terminating lightpaths on the combinatorial lower bound from Section 4.4.2.2 provides another argument for

the fact that most non-zero bounds are found for pt20 instances. In this network scenario, the least total traffic is routed, and as a geographically small topology, regenerators (also terminating lightpaths) are sparsely employed, too. In fact, the least average regenerator number per instance occurs for pt20 in the actual test set. Both other networks pt28 and pt50 require more regenerators on average and, in particular, have higher total demands. The nearly comparable quality of both lower bounds indicates that the same effects have a comparable influence on the LP-based lower bound, too. Experiments with a larger spectrum confirm the observed relation. We recomputed solutions for the reference instances and those using survivability schemes for pt50 with a doubled spectrum size, i.e., using a WDM system that provides 80 wavelengths at twice the cost of a 40 wavelength WDM system. As a result, most converter lower bounds now found for these designs are non-zero, with some exceptions only for the highest protection level with most lightpaths established. So, in combination with the earlier detected indication that need for conversions is driven by higher mean hop lengths of the lightpaths, a good explanation for the observed distribution of converter-free solutions and zero lower bounds over the study instances is obtained.

Summarizing, we can conclude that the described methods yield good lower bounds for Mcwap which offer in many cases a substantial contribution to rank the quality of the assignments. For 25 solutions, even a direct proof of optimality is found thereby. However, this leaves still 298 designs whose initial assignment either is optimal but lacks of a confirmation or, more probably, is non-optimal and thus still improvable.

5.3.4 Improving solutions

Within the design computations, initial assignments have been generated by short runs of an iterative heuristic. This way, optimal solutions have already been found for 336 of all 634 instances. For the remaining cases, we also make use of the advanced methods proposed in Section 4.4. We begin with longer runs of an enhanced heuristic including wavelength extraction. After that, we report on a special way to apply the exact branch-and-price method to seek for better solutions for instances of practical size, before closing this computational study on Mcwap by a concluding summary.

Enhanced heuristic outline. As enhanced heuristic, we apply the iterative method with extraction of wavelengths, described in Section 4.4.1.4. All iteration steps reorder the lightpath sequence again by use of the most promising strategy APRR (all converted lightpaths are pushed in front of the sequence in reverse order). Wavelength extractions are thereby initiated and carried out in different ways depending on the initial number of converters and whether having a positive (combinatorial) lower bound or not (which is in fact recomputed initially to avoid setting of individual runtime-parameters).
The lower bound value influences the selection of candidate wavelengths for extraction. In case of a zero bound, only unconverted wavelengths are extracted. Otherwise, we know in advance that some conversions are unavoidable and therefore allow also for extraction of wavelengths which are converted just once, provided the remaining spectrum after some extractions is small enough such that not too

many converters are fixed thereby. The extractions become this way more flexible to search for optimal assignments with converters.
According to these rules, the set of extractable wavelengths is determined at each iteration. The size of this set is then used as criterion for deciding whether an extraction is initiated. This decision is taken depending on the converter number in the initial assignment, which serves as indicator for the difficulty to find extractable wavelengths at all and thus to enable extraction steps. On the one hand, we clearly prefer to reduce the problem as often as possible, but on the other hand do not want to get stuck in small regions of the search space by allowing consecutive extractions too easy. Therefore, we make the following distinction. If the initial assignment contains more than 20 converters (half of the spectrum size), extractions of wavelengths are probably not often possible, and thus we facilitate such steps by initiation whenever at least two extractable wavelengths are found. Moreover, we allow for an investigation of the generated subproblem within a time limit of 0.25 seconds in competition with an iteration limit of at most 200 iterations. These limits account just for each individual subproblem and do not include time or iterations spent in further extractions which are initiated during the subproblem's investigation. Using such competing limits aims to balance the examination effort for subproblems of different sizes, since smaller ones process iterations much faster than the larger ones. With at most 20 converters in the initial assignment, more unconverted wavelengths can be expected to occur regularly, and we would like to avoid to extract too easily. So, an extraction is only initiated if at least an eighth of the current subproblem spectrum can be removed. For example, at least five wavelengths have to be extracted at once from the initial spectrum of 40 wavelengths, whereas a subproblem with only 23 wavelengths left does not initiate extractions removing less than three wavelengths. Furthermore, quicker subproblem investigations are enforced by allowing for at most 0.2 seconds or 125 iterations to spent on a subproblem, which is further reduced to at most 0.15 seconds or 100 iterations in case the initial wavelength assignment contains not more than 10 converters.
In any of the described application modes, no extraction is allowed during the first 200 iterations in order to avoid too rash plunging into less promising subproblems at the beginning. Whenever a new best solution is found in an extracted subproblem, the successful (sub-)sequence ordering is overtaken by the invoking instance. Finally, we set a CPU time limit of 1800 seconds for each run of the enhanced heuristic in the computational study evaluated next.

Heuristically improved assignments. The enhanced heuristic in fact succeeds to find solutions with less converters for most of the instances. Thereby, the reduction in the number of required converters often shrinks the optimality gap considerably, with an average of 35.84% over all instances. In more detail, Figure 5.31 presents the histogram of the percentages by which the optimality gap is closed by the new solutions found.
The plot shows that the initial assignment remains the best known solution for 82 instances, whereas in all other 216 cases the enhanced heuristic delivers improved solutions. Restricted to the latter set, an average saving of 8.7 converters per instance is achieved. This corresponds to an average gap closing of 49.45%, which splits nearly equally to 48.68% for the 143 instances with zero lower bound and 50.96% for the other 73 instances with non-zero lower bound. Hence, we can con-

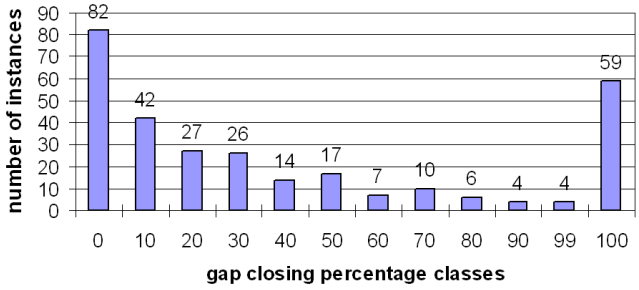


Figure 5.31: *Histogram of gap closing percentages by the improved solutions found with the enhanced heuristic.*

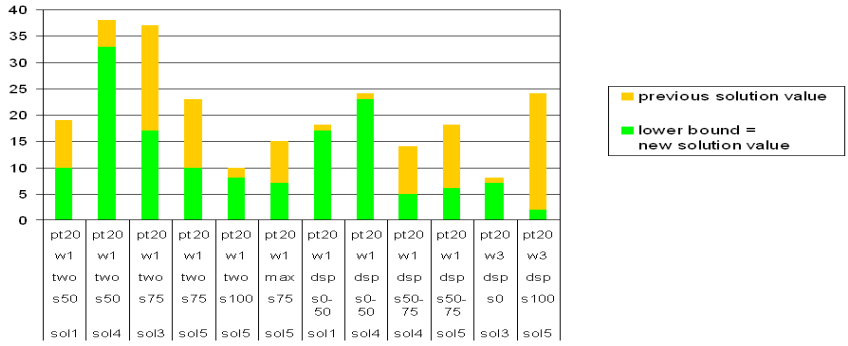


Figure 5.32: *Solution converter numbers for newly proven optimal instances with non-zero lower bound.*

clude that the lower bound value does not influence the improvement potential of the enhanced heuristic. Regarding the individual histogram classes, the lower but non-zero percentages can only refer to instances with sufficiently large gaps and thus high converter numbers, which often have zero lower bounds as observed in the preceeding evaluation. In these instances, unconverted wavelengths occur rarely, and hence extractions cannot be carried out very often. Nevertheless, these extractions yield subproblems which allow for detection of further converter savings. Moreover, we observe that larger gap closing percentages are not seldom realized. In particular, further 59 instances have been solved to optimality. In 47 of these cases, a converter-free solution is found by the enhanced heuristic, whereas the non-zero LP lower bound proves the remaining 12 solutions to be optimal. For these 12 instances, Figure 5.32 visualizes the detected optimal converter numbers in relation to those in the initial assignments.

Now, optimality for solutions with even up to 33 converters is proven by the lower bounds. A special case worth to mention is the solution sol3 for unprotected pt20 with extended hardware w3 setting which already occurred in Figure 5.30 since the corresponding lower bounds differ. So, optimality for this successfully improved solution is only confirmed by the path packing LP optimum. However, comparing all results in Figure 5.32 reveals that the additional conversion reductions by the enhanced heuristic are of varying size, ranging from a single converter up to 22 converters saved. Thereby, larger reduction amounts are not restricted to instances with small lower bounds only, confirming that potential improvements are not correlated with the values of the lower bound. A similar observation holds also for the

other successfully improved instances with non-zero bounds. For these instances, Figure 5.33 subsumes the converter values of both solutions, the initial assignment and the improved solution determined by the enhanced heuristic, together with the associated lower bound in the usual representation by overlaying bars.
For this set, the enhanced heuristic effectuates an average gap closing of 41.31%, where the absolute converter number reductions span a range between one and 21 converters similar to that for the optimized instances. Since the optimal solutions have been left out, each of the displayed instances keeps a strictly positive optimality gap, though some of them have become quite small. Still larger remaining optimality gaps occur predominantly for high protection levels for the scheme 'max' as well as w3 instances with extended hardware, which have already been identified as those for which low converter numbers are harder to obtain.
As in comparison to lower bound values, the achieved reductions do also not stand in an obvious correlation to the initial gap sizes. A mix of various ratios occurs in this regard, and it cannot be seen that, for instance, large advancements are only possible for those instances with a large gap for the initial assignment. Such a correlation can however be observed for the (targeted) protection levels. A closer look exhibits that the instances with higher survivability requirements typically allow for more improvement. In fact, the average number of saved converters is 9.2 over all full protected s100 solutions in Figure 5.33, decreasing to 7.0 for the s75 cases, 3.82 for s50, and only 1.5 for s0 without protection. A strict decrease, though with other slopes, occurs also for the set of zero-bounded instances and, hence, over all improved solutions in common, too. This correlation suggests that instances with more diversified routings and a higher number of lightpaths hold better chances for detecting better assignments when putting a focus on particular subsets of conflict-

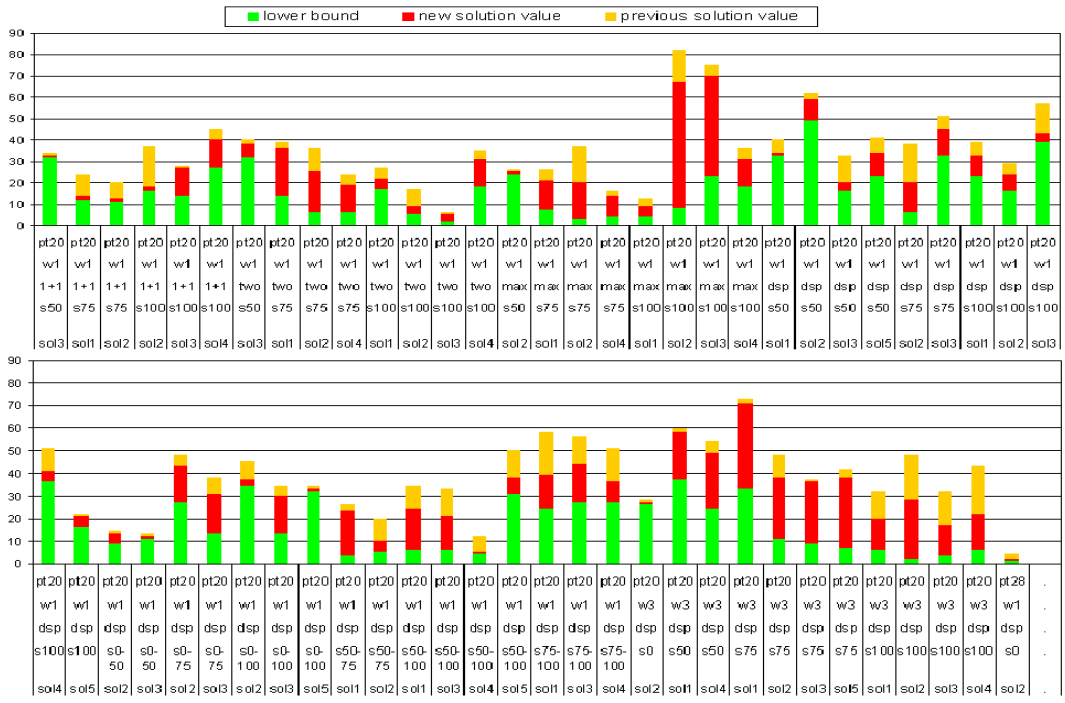


Figure 5.33: *Bounds and solution converter numbers for those instances where the improvement does not completely close the optimality gap to a non-zero lower bound.*

ing lightpaths and wavelengths.
Altogether, we can conclude that the heuristic enhanced by wavelength extractions moves the number of required conversions in many cases a good piece towards the lower bounds. In particular, the improvements bring several instances quite close to optimality. Especially such cases are welcome candidates for the more sophisticated exact approach.

Exact method application. For promising but still unsolved instances, we finally apply the exact branch-and-price method, though in a special way in order to increase the probability of finding improvements. As already mentioned, preliminary computational experiments with the exact method were successful for small instances only, in particular containing just few wavelengths. So, a plain application to the study instances with spectra of 40 wavelengths is not advisable, typically generating huge branching trees.
Instead, we carry out the investigation as follows. At first, we restrict ourselves to promising instances close to optimality, i.e., those instances with a remaining optimality gap of at most three. A small gap increases the chance to be able to early prune branching subtrees. In total, 30 instances apply to this criterion, of which 21 have a non-zero lower bound. Next, we initialize the branch-and price master integer linear program with the so far best solution found, which is transformed to a set of 40 initial columns. Furthermore, we insert the feasibility column which is not contained in the spectrum bound constraint and helps to avoid initial LP infeasibility in search tree nodes with added branching constraints. Now, we restrict the problem in a way inspired by the recent *local branching* approach of Fischetti and Lodi [49]. The goal of this generic approach is to try to improve the primal bound as fast as possible, i.e., seek for good integer solutions fastly. For this, Fischetti and Lodi propose to use asymmetrical branching rules which do not subdivide a problem into subproblems of comparable size (as most commonly in use), but into a small subproblem—in fact, one that can quickly be solved exactly—with as high as possible probability to contain a better integer solution, and the complete remaining problem as second branch. If the primary investigated small subproblems in fact bring up feasible solutions of high quality, further processing can prune wider areas of the still not visited search tree. We make use of this idea and start by considering just one such subproblem defined by adding a single constraint, namely the restriction to reuse at least 38 of the 40 solution columns. Obviously, the restricted problem contains (at least) one integer solution, and the freedom to exchange up to two path packings (without specification which ones) leaves some room for reaching alternative solutions, too. As particular advantage, the added constraint need not be taken into account within the column generation, since all further priced columns get a zero coefficient in the inserted constraint.
When approaching such a restricted problem, the main target is to find better integer solutions as fast as possible. For stimulating this, we implement a special ordering of branching rules. In experiments, some variants of branching on sums of subpath multiplicities as defined in Branching rule 3 on page 181 with uniform weights (of value one) for all subpaths have turned out to be most helpful in generating solutions with less fractional x-variables. The most promising of such rules is the already mentioned full sum branching with $S' = S$. Therefore, this branching rule is applied first. If no possibility for such a branching is given, we next con-

sider branching on y-variables according to Branching rule 1. If this fails, too, a next attempt is undertaken with two further variants of the sum branching rule. In the first variant, the sum is built over all subpaths traversing a node n, i.e., $S' = \{ s \in S \mid n \in N(s) \}$, whereas the second variant takes all undominated subpaths, i.e., $S' = \{ s \in S \mid L(s) \not\subset L(s') \; \forall \, s' \in S \setminus \{s\} \}$. If still lacking of a branching after that, we continue with branchings on a single path multiplicity as defined in Branching rule 2, branchings on sums of two paths, and, for final completeness, branchings on multiple subpath multiplicities in parallel specified by Branching rule 4. However, the latter rule has never been required in any computational experiments, and in the actual study, the two previous ones are never applied either.

Branch-and-price algorithm results. For this study, we set a time limit of again four hours of CPU time for each computation. Within the provided running time, the branch-and-price algorithm performs differently on the selected 30 instances of varying size, which are as usual listed in Appendix B with the corresponding results. In total, the branch-and-price method successfully solves seven restricted problems to proven optimality, while the remaining 23 runs are stopped by timeout.

The observed durations to determine the LP lower bound already indicate that limited progress is to expect, especially for the larger instances. In addition, restricting to a subproblem seems to make optimizing the linear programs harder. The results indeed show ten cases in which solving the root LP already exceeds the time limit, unsurprisingly both pt50 instances among them. In the 20 computations where the root LP optimum is successfully determined in time, we find 14 integral optimum values all matching the original LP lower bound, whereas the other six ones have become non-integral. For two of the latter cases, such an increased lower bound (for the restricted problem) allows to terminate directly by decreasing the absolute value of the root gap to less than one. Here, the subproblem restriction reveals as too tight for including solutions with less converters. So, for totally 18 instances the branching procedure is finally initiated. The higher LP resolving effort however prevents in most cases from finishing many branching tree nodes and thus interrupts the search too soon by timeout, but in five cases the algorithm terminates regular and returns in fact better assignments as optimal subproblem solutions. For these instances, Figure 5.34 gives an overview on all determined lower bound and solution values. The stacked values have now been slightly shifted in order to increase separability in case the used colors do not contrast sufficiently with each other.

The figure reveals that up to two converters are saved by the newly generated solutions. So, a final absolute gap of a single converter remains in three cases, whereas a global optimal solution has been detected this way for two more instances. Both proofs of optimality are again provided by the determined non-zero lower bounds, once indeed only by the LP-induced bound which thereby further confirms its strength. The overall success rate of the branch-and-price algorithm, however, is still improvable and advises further investigation, in particular on accelerating LP (re-)solving. Nevertheless, this experimental study shows that this method, in combination with good lower bounds, is a promising approach for Mcwap, also on instances of practical size. Thereby, the presented way of application, by use of local branching or similar ideas, can be a good inspiration for continuing research on this topic.

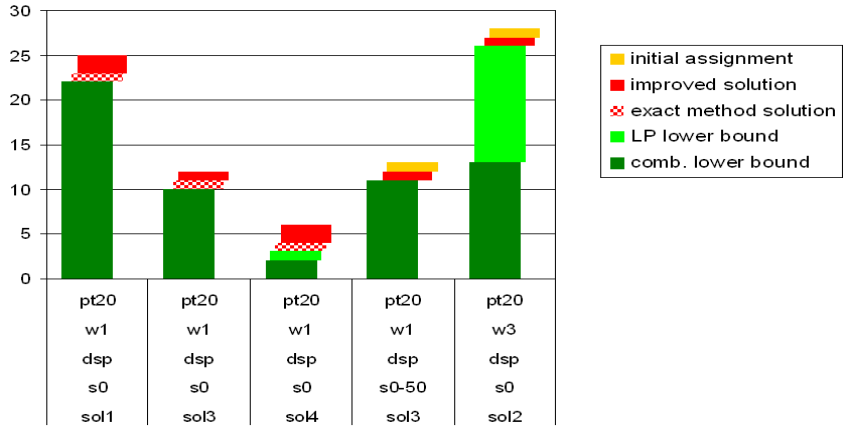


Figure 5.34: *Value overview for instances improved further by the exact branch-and-price algorithm.*

Summary of achieved Mcwap **results.** Closing the computational study on Mcwap, we give a brief overview on the overall achieved results. In the end, an optimal solution has been determined for totally 397 of the 634 instances in the test set. In most cases, the developed heuristics are able to generate a converter-free assignment, whereas optimality of the best known solution with converters is confirmed for 39 instances or roughly 10% of cases by the LP lower bound. We remark that optimality of a solution with a number of converters larger than the lower bound could not be proven so far. In the test set, a non-zero lower bound has been found for a total of 126 instances and contributes also in the 87 still unsolved cases substantially to a better figure of the quality of a solution at hand. This way, further 28 instances are identified with an absolute gap of not more than three converters left and thus to be quite close to (just their proof of?) optimality.

Conclusions

In this thesis, mathematical optimization methods for the cost-efficient design of survivable optical networks have been studied. The development was guided by the intention to catch the practical situation best possible for twofold benefit: to allow for helpful support of planners in their daily doing as a ready-to-use planning tool which has been delivered to Telekom Austria, and to serve as a flexible base for various studies on and evaluations of alternative devices, architectures, or concepts, as carried out in cooperation with T-Systems International. In either case, analyzing and comparing computed designs would be somehow random without the knowledge of their individual quality. This purpose especially entailed to seek for approaches which do not only produce good solutions, but offer further information to assess their quality, too.

Models and concepts. We have provided a flexible framework that enables an accurate representation of different network architectures and have formulated mathematical models encoding the corresponding problem to find cost-efficient designs. Three characteristic layouts of optical networks have been discussed in detail: opaque networks, transparent networks with single-hop routings, and transparent networks with multi-hop routing. A couple of extensions and adaptations have been exemplified, too, and carrying over models as well as algorithms to directed networks, demands, and routings is straightforward. In addition to this, there exist further variations with practical background which are less easy to implement. As an example, costs dedicated to individual routing paths are not integrated so far and would enable, for instance, to take costs for selective mounting of ports at switches or wavelength interfaces at multiplexers more precisely into account. Though framework and models are simply extendible for this, the solution methods need major changes and further research to maintain the achieved tractability.
Moreover, we contributed the new survivability concept Demand-wise Shared Protection (DSP), a path protection approach tailored to optical networks. As compromise between capacity-intensive dedicated path protection and operationally complicating shared path protection schemes, DSP constitutes a special balance between low backup resource requirements and small effort for network management and recovery. Applicability to scenarios with multiple failures has been proven as well. Further enhancements to reduce capacity consumption might be achievable when sharing of backup capacities by connections from different demands with a common endnode is explored. For instance, a backup connection from A to B via C could simultaneously serve to protect a part of traffic for a demand from C to B, if some disjointness relations hold. Exploitation of such capabilities with adequate formal-

ization and integration into the presented schemes requires further investigations.

Solution methodology. We have developed an appropriate solution methodology to meet the claimed requirements regarding closeness to reality, flexibility, a quality guarantee of solutions. In order to make the comprehensive task computationally tractable, we proposed to decompose the models in such a way that two individually hard subproblems are decoupled, while quality assessment by a still feasible lower bound on the total cost is maintained and only limited solution quality is sacrificed. Rather than routing of wavelengths, we explicitly separated the wavelength assignment from connection routing. We thus obtained on the one hand a dimensioning and routing part which has beneficial similarity to non-optical network design, and on the other hand a coloring-like problem for assigning the wavelengths to routed lightpaths using as few converters as possible.
In view of the dimensioning and routing part, we carried out suitable pre- and postprocessing transformations which enable us to access existing know-how and sophisticated methods for solving this well-known core task. A special note for application as planning support is to be remarked. Much faster provision of alternative dimensioning and routing solutions can be achieved by shrinking the sets of installable capacities to restricted selections, an idea we implemented in our tool by means of parameterized heuristic levels. With a correcting postprocessing on the unrestricted sets, the quality of solutions obtained with substantially shorter running time typically suffers limited loss. As a drawback, the lower bound reported when using restricted capacity sets is not valid anymore for the original problem, but a single run on the unrestricted setting can supply the desired information. Moreover, we have discussed different solver integrations and suggest for particular scenarios of interest to carry out further comparative evaluations to explore which way of application suits best.
For solving the subsequent wavelength assignment problem, the new and characteristic feature of optical network design, we followed combined approaches from both primal and dual side. We developed and enhanced heuristics for a fast construction of feasible assignments, accompanied by methods for determination of lower bounds on the number of required converters. In this regard, the path packing formulation has been shown to be a suitable model for getting strong lower bounds as well as a base for an exact branch-and-price method, but might still cause high effort for solving corresponding linear relaxations. Further progress on accelerating this process is therefore required. A similar remark holds for the exact branch-and-price method, where a promising idea based on local branching has been indicated. Such and other improving ideas might help to enlarge the range of successfully solvable instances. Additionally, a development of further heuristics with emphasis on cases with many unavoidable converters might provide valuable insights and enrich the algorithmic variety.

Computational study. Finally, we carried out comprehensive computational studies following two goals: to evaluate the performance of the solution methodology, and to exemplify the offered possibilities for planning support and design analyses. For this purpose, our industrial partners provided us with several practice-relevant instances from small to challenging size together with realistic settings for hardware and cost relations.
The obtained results indicate the potential of our approach, models, and algorithms

and reveal that practically applicable optical network designs of high quality can be determined within reasonable time. Moreover, DSP has proven to be a promising concept, with particular strength when requesting partial protection, for achieving survivability cost savings at similar operational effort compared to 1+1 path protection as most widely applied concept in practice. Among its variants, the most general DSP scheme provides the highest potential, but also adds further complexity to the problem. Methodological advancements to master the reduced tractability are needed, and we strongly believe that the concept's outcome performance would benefit from this further.

Outlook. As final outlook, some more general research directions for network design enhancements are to be mentioned. A main aspect for future work consists of the integration of traffic grooming capabilities with respect to different bitrate lightpaths or even to optical in parallel to other technologies. Such a concerted planning of multiple layers of networks, known as *multi-layer design*, forms an interesting, though very complex extension of the design problems.

Another generalization stipulated by practice consists of a comprehensive approach for coordinated planning of infrastructure over longer periods of time, known as *multi-period planning*. Keeping our methodology open for upgrade planning purposes is a first step in this direction. Nevertheless, taking network migration along multiple subsequent reorganizations explicitly into account remains a challenging task.

Moreover, network providers often complain about an experience they make repetitively: the uncertainty of demands. Neither highest care in taking decisions nor even provably optimal designs rolled out in the networks rule out the danger that traffic predictions are fooled, and the established infrastructure turns out to be suboptimal for the demands arising in reality. Therefore, providers regularly urge for development of methods that take prediction inaccuracy explicitly into account. Network designs with high probability of adequacy in most relevant future scenarios have top priority on the wishlists.

Appendix A

Notation

The following overview gives a brief summary on notation used throughout the thesis. Some profound introductions into relevant areas are provided by Lorentz [106] for linear algebra, Papadimitriou [136] and Garey and Johnson [52] for complexity theory, Diestel [38] for graph theory, Jensen and Toft [79] for graph coloring problems, Ahuja et al. [2] for networks and flows, Chvátal [32] for linear programming, and Nemhauser and Wolsey [127] for integer and combinatorial optimization. Finally, we remark the handbooks by Grötschel et al. [58] and Schrijver [150] as comprehensive compilations of knowledge and literature in combinatorial optimization.

Sets and algebra

A *set* is a collection of unique items of any type. In contrast, a *multi-set*, also known as *family*, is an arbitrary collection of items of any type, i.e., allows for repetitive occurrence of items. We call the number of items equal to s in a multi-set S the *multiplicity of s in S*, denoted by $m_S(s)$ as *multiplicity function* defined on any possible item (while only items contained in the multi-set S have non-zero multiplicity). If the referred multi-set S is clear, we also write abbreviately $m^s := m_S(s)$. The *cardinality* of a set or multi-set S is the total number of contained (not necessarily different) items and denoted by $|S|$. The empty set (or multi-set) is denoted by $\emptyset$.
Some standard notation is used for sets. For two sets A and B, we write $A \times B$ for their Cartesian product, $A \cup B$ and $A \cap B$ for their union and intersection, respectively, and $A \setminus B$ for the set difference. Moreover, $A \subset B$ and $A \subsetneq B$ stand for set inclusion respective proper set inclusion. The same notation carries over to multi-sets as follows. For two multi-sets C and D, the union $S = C \cup D$ is a multi-set where each item s has the multiplicity $m_S(s) = m_C(s) + m_D(s)$. In the intersection multi-set $S = C \cap D$, each item has multiplicity $m_S(s) = \min\{m_C(s), m_D(s)\}$. The multi-set difference $S = C \setminus D$ contains each item with multiplicity $m_S(s) = \max\{0, m_C(s) - m_D(s)\}$. Finally, multi-set C is included in multi-set D, i.e., $C \subset D$, if $m_C(s) \leq m_D(s)$ holds for all items $s \in C$, and properly included, i.e., $C \subsetneq D$, if $m_C(s) < m_D(s)$ holds for at least one such item.

We use the well-known notion of $\mathbb{R}$, $\mathbb{Q}$, and $\mathbb{Z}$ for the sets of all real, rational, and integer numbers, respectively. The restrictions to the corresponding subsets of non-negative numbers including zero are denoted by $\mathbb{R}_+$, $\mathbb{Q}_+$, and $\mathbb{Z}_+$. For strictly positive numbers, additional exclusion of zero is indicated for integers by use of the set of natural numbers $\mathbb{N} := \mathbb{Z}_+ \setminus \{0\}$ and otherwise stated explicitly.
Given any real number $a \in \mathbb{R}$, we refer to the minimum integer number larger than or equal to a by $\lceil a \rceil$, and to the maximum integer number smaller than or equal to a by $\lfloor a \rfloor$. Let $k, n \in \mathbb{Z}_+$, then $k!$ is equal to 1 if $k = 0$ and equal to $1 \cdot \ldots \cdot k$ otherwise, and $\binom{n}{k}$ is equal to $\frac{n!}{k!(n-k)!}$ if $0 \leq k \leq n$ and 0 otherwise. Let $f : \mathbb{Z}_+ \to \mathbb{Z}_+$ be an arbitrary function on non-negative integers. Then the set $\mathcal{O}(f) := \{ g : \mathbb{Z}_+ \to \mathbb{Z}_+ \mid \exists\, b, c, N \in \mathbb{Z}_+ : g(n) \leq b + c \cdot f(n) \; \forall\, n \geq N \}$ contains all such functions whose growth is asymptotically bounded by that of f.

Complexity

The *complexity classes* $\mathcal{P}$ and $\mathcal{NP}$ contain those decision problems which can be solved by a deterministic respectively a non-deterministic Turing machine in polynomial time (in the size of input). The complexity class $\mathcal{ZPP}$ contains all decision problems which can be solved by a probabilistic Turing machine in expected polynomial time (in the size of input).
A decision problem is called $\mathcal{NP}$-complete if it is in $\mathcal{NP}$ and any other problem in $\mathcal{NP}$ can be reduced to it in polynomial time and size. An optimization problem is called $\mathcal{NP}$-hard if its associated decision problem is $\mathcal{NP}$-complete.

Graphs and networks

An *undirected graph* $G = (V, E)$ (or simply *graph*) is a pair of two finite sets, a non-empty set $V \neq \emptyset$ of *vertices* and a set $E \subset \{ \{v, w\} \mid v, w \in V,\ v \neq w \}$ of unordered pairs of different vertices, each such pair called an *edge*. If E constitutes a multi-set of edges, we call $G = (V, E)$ an *undirected multi-graph* (or simply *multi-graph*). In the following, we introduce further notation in common for an arbitrary graph or multi-graph $G = (V, E)$ and distinguish between both cases only where necessary.
Given an edge $e = \{v, w\} \in E$, we also write abbreviately $e = vw$ and say that v and w are the *endvertices* of e, vertex v is *adjacent* to w (and vice versa), while edge e is *incident* to v and w (and vice versa). (Note that the specification of an edge by its endvertices is only unique for graphs, not for multi-graphs.) Two different edges with the same endvertices in a multi-graph are called *parallel*. For a vertex $v \in V$, we call each adjacent vertex $w \in V$ a *neighbor of v (in G)* and denote the set of all neighbors of v by $V(v) := \{ w \in V \mid \{v, w\} \in E \}$. Similarly, we denote the set of all edges incident to v by $E(v)$. The number of incident edges of a vertex v in G is called the *degree of v (in G)*, and the *degree of G*, denoted by $\Delta(G)$, is the maximum degree over all vertices in G.
We call $G' = (V', E')$ with $V' \subset V$ and $E' \subset E$ a *subgraph* (or *multi-subgraph*) of G. Special subgraphs are obtained by removal of edges, which are simply deleted from the edge set, and removal of nodes, which are removed from the node set to-

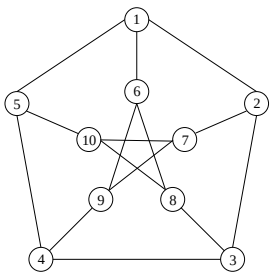
gether with a removal of all incident edges from the edge set. For $W \subset V$, the *subgraph induced by* W (or *multi-subgraph induced by* W) is the (multi-) subgraph $G[W] = (W, E[W])$ with $E[W] := \{\, e \in E \mid e = \{v, w\},\ v, w \in W \,\}$.
A mapping $\omega : E \to \mathbb{R}$ is called an *edge weighting (on* G*)*, and the values $\omega_e := \omega(e)$ associated with the edges are called *edge weights*. Similarly, a *vertex weighting (on* G*)* is a mapping $\omega : V \to \mathbb{R}$ and $\omega_v := \omega(v)$ are the *vertex weights*.
A *path (in* G*)* is an alternating sequence of vertices and edges in G of the form $p = (v_0, e_1, v_1, e_2, v_2, \ldots, e_h, v_h)$ where $v_0, \ldots, v_h \in V$ and $e_1, \ldots, e_h \in E$ such that $e_i = v_{i-1}v_i$ for all $i = 1, \ldots, h$. The vertices v_0 and v_h are the *endvertices of* p. If G has edge weights ω, the *length of* p *(w.r.t.* ω*)* is defined as $\omega(p) := \sum_{e \in L(p)} \omega(e)$. Moreover, we define the *path vertex set* $V[p] := \{v_0, \ldots, v_h\}$, the *path inner vertex set* $V(p) := \{v_1, \ldots, v_{h-1}\}$, and the *path edge set* $E(p) := \{e_1, \ldots, e_h\}$. A path p' is a *subpath of* p if there are $0 \le k < l \le h$ such that $p' = (v_k, e_{k+1}, v_{k+1}, \ldots, e_l, v_l)$. A path p is called *closed* or a *cycle* if $v_0 = v_h$. A path p is called *simple* if it does not contain a subpath $p' \neq p$ being a cycle.
We call two vertices $v, w \in V$ *connected*, if there exists a path with v, w as endvertices. We call G *connected* if any two different vertices $v, w \in V$ are connected. The maximal connected induced subgraphs of G are the *components of* G.
For two vertices $v, w \in V$, an *v,w-edge cut (in* G*)* is a (multi-) subset $\Gamma \subset E$ of edges whose removal disconnects v and w, i.e., the (multi-) subgraph $G' = (V, E \setminus \Gamma)$ does not contain a path with v, w as endvertices. Similarly, an *v,w-node cut (in* G*)* is a subset $\Gamma \subset V$ of vertices whose removal disconnects v and w. A *generalized v,w-cut (in* G*)* is a (multi-) subset $\Gamma \subset V \cup E$ of vertices and edges whose removal disconnects v and w.
A *vertex coloring (of* G*)* is a mapping $C : V \to \{1, \ldots, c\}$ from vertices to a set of colors, represented by the numbers $1, \ldots, c$, such that any two adjacent vertices have different colors, i.e., $C(v) \neq C(w)$ holds for any edge $e = vw \in E$. The *chromatic number of* G is the minimum number of colors for which a vertex coloring of G exists and denoted by $\chi(G)$. Similarly, an *edge coloring (of* G*)* is a mapping $C : E \to \{1, \ldots, c\}$ from edges to a set of colors, represented by the numbers $1, \ldots, c$, such that any two edges incident to a common vertex have different colors, i.e., $C(e) \neq C(f)$ holds whenever there exists a vertex v indicent to e and f. The *chromatic index of* G is the minimum number of colors for which an edge coloring exists and denoted by $\chi'(G)$.
A *hypergraph* $G = (V, E)$ is a pair of finite sets, with a non-empty set $V \neq \emptyset$ of vertices and a set $E \subset 2^V := \{\, V' \mid V' \subset V \,\}$ of unordered subsets of vertices called *hyperedges*.
A *digraph* (or *directed graph*) $D = (V, A)$ is a pair of finite sets, a non-empty set $V \neq \emptyset$ of vertices as in graphs and a set $A = \{\, (v, w) \mid v, w \in V,\ v \neq w \,\}$ of ordered pairs of different vertices called *arcs*. With A being a multi-set, $D = (V, A)$ is called *multi-digraph*. The arcs can be interpreted as directed edges, where an arc $a = (v, w)$ starts at its *tail* v and ends at its *head* w. A digraph is called *bidirected* if for each arc $a = (v, w) \in A$, the opposite arc $(w, v) \in A$ is also contained (with same multiplicity in case of multi-digraphs). A *directed path from* v_0 *to* v_h in a digraph $D = (V, A)$ is an alternating sequence of vertices and arcs $p = (v_0, a_1, v_1, a_2, v_2, \ldots, a_h, v_h)$ with $v_0, \ldots, v_h \in V$ and $a_1, \ldots a_h \in A$ such that $a_i = (v_{i-1}, v_i)$ for all $i = 1, \ldots, h$. A *(v,w)-arc cut (in* D*)* is a (multi-) subset $\Gamma \subset A$ after whose removal no directed path from v to w exists.

We remark that the well-known theorems of Menger [118] as well as of Ford and Fulkerson [50] for edge cuts can be carried over to generalized cuts by using the following transformation. For this, we consider the theorem variants on digraphs, where (multi-) graph G is transformed to a (multi-) digraph D by substituting each edge $e = vw$ by the two arcs (v, w) and (w, v) (with same multiplicities in case of multi-graphs). For the transformation, we replace each vertex v by a pair of vertices v_1, v_2, where each arc (v, w) with tail v is replaced by an arc (v_2, w) with v_2 as tail, and each arc (w, v) with v as head is replaced by an arc (w, v_1) with head v_1. Moreover, an additional arc (v_1, v_2) from v_1 to v_2 is inserted. Then each generalized cut in G containing a vertex v can be transformed to an arc cut in the constructed (multi-) digraph containing the arc (v_1, v_2), and vice versa. We implicitly use this correspondence when applying the theorems appropriately adapted to generalized cuts.

In connection with telecommunication infrastructures, we deal with *networks* (and *topologies*) which are used synonymously to multi-graphs. Moreover, we denote a network by $\mathcal{N} = (N, L)$ and use the namings *nodes* instead of vertices and *links* instead of edges. By use of these replacements, all notations introduced for graphs apply synonymously to networks, topologies, nodes, and links as well. For stating explicitly that a network does not refer to a multi-graph, but to a graph, we call the network *simple*.

Some special classes of graphs and networks are to be mentioned. As our definitions carry over literally to multi-graphs (with optional addition of 'multi' to the naming), we restrict the description to graphs. Moreover, we use the same names for reference to the corresponding networks (with optional addition of 'network').

A *chain* is a connected graph where each vertex has at most two neighbors. A *forest* is a graph which does not contain any simple cycle with at least three edges as subgraph. A *tree* is a connected forest. A *star* is a connected graph with at most one vertex that has two or more neighbors. A *spider* is a connected graph with at most one vertex that has three or more neighbors. A *ring* is a connected graph where each vertex has exactly two neighbors. A (multi-) graph is *series-parallel* if it arises from a forest by repetitive replacement of an edge by parallel edges or by a series of edges. A graph is *bipartite* if it does not contain a cycle with odd number of edges. A *clique* is a graph where each pair of vertices is connected by an edge. The *clique number* of a graph G, denoted by $\omega(G)$, is the maximum number of vertices in an induced clique subgraph of G. A graph G is *perfect* if $\chi(G') = \omega(G')$ holds for each induced subgraph G' of G. An *interval graph* is the intersection graph of a multi-set $\mathcal{I}$ of non-empty intervals on the real line, i.e., the graph has a vertex v_I for each interval $I \in \mathcal{I}$, and two vertices v_I, v_J are adjacent if and only if $I \cap J \neq \emptyset$. Similarly, a *circular-arc graph* is the intersection graph of a multi-set of non-empty intervals on a circle. Finally, the *Petersen graph* is shown below.



Appendix B

Numerical data and results

This appendix contains a compilation of the numerical data and results which have been subsumingly reported, evaluated, and interpreted in Chapter 5. The structuring follows the chapter's presentation. We first provide as input data the demand matrices together with all link lengths and list further topology information. Then we summarize the numerical results, beginning with the results from the first investigation on optical network designs in the same order of case studies, and followed by the results on wavelength assignments.

For reference to individual instances, we reuse the abbreviation scheme introduced in Chapter 5. We refer

- to networks by the topology identifiers pt14, pt17, pt20, pt28, and pt50,
- to survivability schemes by the symbols '1+1' for 1+1 path protection, 'two' for 2-DSP, 'max' for max-DSP, and 'dsp' for the general DSP scheme,
- to protection levels by the percentage indicators s0, s50, s75, s100 for greenfield planning and their originating-targeting level combinations s0-50, s0-75, s0-100, s50-75, s50-100, and s75-100 for upgrade planning scenarios,
- to hardware settings by 'w1' for the regular and by 'w3' for the extended transparent case as well as by 'opq' for the opaque case, and
- to alternative solutions by the numbering sol1, sol2, ... for each instance.

B.1 Traffic and topology data

The complete input data for each instance consists of the hardware specifications, the network topology, and the traffic demands. In Chapter 5, the hardware devices have already been described in full detail, and all module types can be installed arbitrarily often. Hence, the data input is to be completed by the particular demand values and link lengths, which we next present in a combined matrix for each network. The demand values refer to symmetric and thus undirected traffic requirements and are shown in the upper right triangle matrices. The lower left triangle matrices contain the link lengths in kilometers for all involved physical links.

pt14	Palo Alto	San Diego	Boulder	Washington	Atlanta	Urbana	Ann Arbor	Lincoln	Princeton	Ithaca	Pittsburgh	Houston	Salt Lake City	Seattle
Palo Alto		26	9	20	16	32	17	7	8	25	20	26	11	12
San Diego	834		13	29	23	47	24	10	11	37	28	38	15	16
Boulder				15	11	22	12	6	5	17	13	20	8	7
Washington					37	74	51	17	17	57	44	51	15	17
Atlanta						61	35	11	30	84	97	39	12	13
Urbana							63	22	27	91	72	86	23	26
Ann Arbor								13	16	55	43	42	13	14
Lincoln			870			864			5	17	13	19	6	6
Princeton				312			942			84	118	18	6	6
Ithaca				468			720				162	61	18	21
Pittsburgh					1008	846			540	438		47	14	16
Houston		2520	1746	2364	1350								20	21
Salt Lake City	1152		684				2820							8
Seattle	1338	2056				3408								

pt17	Berlin	Bremen	Dortmund	Duesseldorf	Essen	Frankfurt	Hamburg	Hannover	Karlsruhe	Koeln	Leipzig	Mannheim	Muenchen	Norden	Nuernberg	Stuttgart	Ulm
Berlin		5	7	8	6	24	10	11	3	8	18	4	8		7	10	6
Bremen			3	4	3	11	6	6	1	4	7	2	4		3	4	3
Dortmund				7	6	16	6	7	2	7	10	3	5		4	6	4
Duesseldorf					7	18	7	8	2	10	11	3	6		5	7	4
Essen			37	36		13	5	6	2	6	8	3	4		4	5	3
Frankfurt							21	24	6	19	34	12	19	116	16	24	14
Hamburg	306	114						11	2	7	14	4	7		6	9	5
Hannover	298	120	208			316	157		3	8	16	4	8		7	10	6
Karlsruhe										2	4	2	2		2	3	2
Koeln			88	41		182					11	4	6		5	8	5
Leipzig	174					353		258				6	12		10	14	9
Mannheim						85			64				4		3	5	3
Muenchen															6	9	6
Norden		144	278														
Nuernberg						224					275		179			7	5
Stuttgart									74						187		8
Ulm													143			86	

pt20	Bischofshofen	Dornbirn	Eisenstadt	Feldkirch	Graz1	Graz2	Innsbruck1	Innsbruck2	Klagenfurt1	Klagenfurt2	Liezen	Linz1	Linz2	Salzburg1	St.Johann	St.Poelten	Wien1	Wien2	W. Neustadt	Salzburg2
Bischofshofen		3		3	3		6		3		3	3	3	3	3	3	3	3	3	
Dornbirn				6	3		15		3		3	3	3	9	3	3	3	3	3	
Eisenstadt					9												9			
Feldkirch		25			3		3		3		3	3	3	3	3	3	3	3	3	
Graz1			190				3		21		3	6	3	3	3	3	27	9	6	
Graz2					1													6		
Innsbruck1				170					12		3	3	3	6	3	3	9	6	3	
Innsbruck2	200	290					3							6			6			
Klagenfurt1	190				150						3	3	3	3	3	3	12	6	3	
Klagenfurt2						290			5				6					6		
Liezen					160							3	3	3	3	3	3	3	3	
Linz1											120		27	24	3	6	30	9	3	
Linz2												10		6	3	6	3	9	3	
Salzburg1	50											150			6	3	36	3	3	
St.Johann	10						200			210	130					3	3	3	3	
St.Poelten												130	190				12	12	3	
Wien1			70								170					80		66	15	
Wien2																70	5		9	
W. Neustadt			50			190												60		
Salzburg2													190	5	150					

pt28	Amsterdam	Athens	Barcelona	Belgrade	Berlin	Bordeaux	Brussels	Budapest	Copenhagen	Dublin	Frankfurt	Glasgow	Hamburg	London	Lyon	Madrid	Milan	Munich	Oslo	Paris	Prague	Rome	Stockholm	Strasbourg	Vienna	Warsaw	Zagreb	Zurich
Amsterdam		1	3	1	5	2	3	1	2	1	5	8	5	10	3	3	4	5	2	3	2	4	2	3	2	5	1	2
Athens			1	1	2	1	1	1	1	1	2	2	2	2	1	1	2	2	1	1	1	2	1	1	1	2	1	1
Barcelona				1	3	3	2	1	1	1	3	5	3	6	3	4	4	3	1	2	2	4	2	2	2	4	1	2
Belgrade		1209			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1
Berlin						3	3	2	2	1	6	8	8	8	3	3	5	6	2	3	4	5	3	3	3	8	1	3
Bordeaux							2	1	1	1	3	4	3	5	3	3	3	3	1	2	1	3	1	2	1	3	1	2
Brussels	259							1	1	1	3	4	3	5	2	2	3	3	1	2	1	2	1	2	1	3	1	1
Budapest				474					1	1	2	2	2	2	1	1	2	2	1	1	1	2	1	1	2	3	1	1
Copenhagen					540					1	2	3	2	3	1	1	2	2	1	1	1	2	1	1	1	2	1	1
Dublin											1	2	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Frankfurt							474					8	6	9	3	3	6	7	2	4	3	5	3	6	2	6	1	3
Glasgow	1067									462			8	18	4	5	8	8	3	5	3	7	5	4	3	8	1	4
Hamburg	552				381						592			9	3	3	5	6	2	3	3	4	3	3	2	7	1	3
London	540									690					5	5	8	9	3	7	4	8	5	5	3	9	1	4
Lyon			796													2	5	3	1	3	2	3	2	3	1	3	1	2
Madrid			760			834											3	3	1	2	2	3	2	2	1	3	1	2
Milan																		7	2	4	3	7	2	5	3	6	1	4
Munich					757						456					522			2	3	4	6	3	5	3	7	1	4
Oslo								722												1	1	1	1	1	1	2	1	1
Paris						747	393						514	594							2	3	2	3	1	3	1	2
Prague					420			668														3	1	2	2	4	1	2
Rome		1500														720							2	3	2	6	1	3
Stockholm																			623					2	1	4	1	1
Strasbourg											271									600					2	4	1	3
Vienna																		534			376				3	1	1
Warsaw					775			819															1213				2	3
Zagreb				551																		783		400			1
Zurich															507	327								218				

pt50 (part 1)	Essen	Duesseldorf	Koeln	Dortmund	Aachen	Muenster	Koblenz	Siegen	Wesel	Berlin	Leipzig	Dresden	Erfurt	Chemnitz	Schwerin	Magdeburg	Greifswald	Hamburg	Bremen	Kiel	Oldenburg	Flensburg	Bremerhaven	Norden	Hannover	Bielefeld	Braunschweig	Osnabrueck	Kassel	Frankfurt	Darmstadt	Mannheim	Kaiserslautern	Giessen	Trier
Essen		34	9	9	2	2	2	2	2	2	2	2	2	2	2	2		2	2	2			2	2	2		2		2	2	2				
Duesseldorf	30		76	25	3	6	2	2	5	4	2	2	2		2	2	2	15	2	2	2	2	2	2	19	4	2	2	2	33	2	2		2	2
Koeln		34		13	10	2	5	4	2	11	3	2	2	2	2	2		5	2	2	2			2	2	12	2	2	2	2	28	2	2	2	2	5
Dortmund	31			2	6	2	5	2	2	5	2	2	2	2	2	2	2	3	2	2	2	2	2	2	15	8	2	2	2	6	2	2		2	
Aachen		61			2	2		2	2		2	2			2				2				2	2				2		3			2		2
Muenster			51					2	2	2					2			2	2				2	2	2	5		2	3	3					
Koblenz		78						2		2		2						2					2	2		2			2	2	2	6	2	2	2	2
Siegen			81				64			2								2					2						2		2		2		
Wesel	35			103						2								2					2	2		2			2						
Berlin											20	11	3	2	3	7	4	23	2	2	2		2		25	2	2	2	2	16	2	2	2	2	2	2
Leipzig										145		10	11	3	2	6	2	17							3	2	2			4	2	2			
Dresden										162	99		2	4	2	2	2	2	2	2				2	2	2	2		2	2	2	2			2
Erfurt											102	189		2	2	6		2	2	2					2	2	2		2	5	2	2		2	
Chemnitz												60	136		2	2		2							2	2	2			2		2		2	
Schwerin										176						2	2	8	2	3					2	2				2		2			
Magdeburg										123	102				163		2	2	2	2					2	2	3			2		2			
Greifswald										175					135			2	2	2					2	2				2					
Hamburg															97				13	14	2	3	3	2	71	3	2	2	2	11	2	2		2	
Bremen																				2	4	2	2	3	11	2	2	2	2	2	2	2		2	2
Kiel															119			85			2	3	2		2	2	2		2	2	2	2			
Oldenburg									197										39				2	2	2	2		2		2					
Flensburg																				65			2	2	2					5					
Bremerhaven																			52			147		2	2					2					
Norden									219												84			2	2			2		2					
Hannover																		115	55						25	5	3	6	49	2	2		2	2
Bielefeld						61		133																82		2	4	3	3	2	2		2		
Braunschweig																	77	149						98	138			2	2						
Osnabrueck						44															96			88					2						
Kassel			142										114													127			2	2	2		2		
Frankfurt							81																							25	35	2	9	2
Darmstadt																													27		4	2	2	3
Mannheim																														44		4	2	2
Kaiserslautern							103																							80				2
Giessen								56																				100	51					
Trier				120		94																												
Fulda																												86	85			71		
Saarbruecken																															56		65	
Stuttgart																																		
Karlsruhe																														53	66			
Ulm																																		
Konstanz																																		
Freiburg																																		
Muenchen																																		
Augsburg																																		
Kempten																																		
Passau																																		
Nuernberg																																		
Bayreuth											164			135																					
Wuerzburg													153																						
Regensburg																																		

continued on next page

continued from previous page

pt50 (part 2)	Fulda	Saarbruecken	Stuttgart	Karlsruhe	Ulm	Konstanz	Freiburg	Muenchen	Augsburg	Kempten	Passau	Nuernberg	Bayreuth	Wuerzburg	Regensburg
Essen		2	2	2				2	2			2		2	
Duesseldorf	2	2	5	2	2		2	4	2			2		2	2
Koeln	2	2	7	2	2	2	2	3	2	2		3	2	2	2
Dortmund	2	2	2	2	2		2	2	2			2		2	
Aachen	2	2				2	3					2			
Muenster	2					2						2			
Koblenz	2	2				2								2	
Siegen	2	2				2						2			
Wesel	2	2				2									
Berlin	2	2	3	2	2	2	2	4	2		2	3	2	2	
Leipzig			2	2	2			3	2			9		2	
Dresden			2	2	2	2	2	2	2			2	2	2	2
Erfurt	2		2	2				2	2			2	2	2	
Chemnitz			2	2				2	2			2	2	2	
Schwerin			2	2				2	2			2		2	
Magdeburg			2	2				2	2			2	2	2	
Greifswald				2											
Hamburg	2	2	3	2	2		2	6	2			2		2	2
Bremen			2	2				2	2			2		2	
Kiel			2	2				2	2			2	2	2	
Oldenburg			2					2							
Flensburg															
Bremerhaven															
Norden		2				2		2			2				
Hannover	2	2	2	2	2	2	2	2	2			4		2	
Bielefeld	2	2	2	2	2			2	2			2	2	2	
Braunschweig			2	2				2							
Osnabrueck			2					2							
Kassel	2		2	2				2				2			
Frankfurt	2	5	19	8	5	3	2	3	2	2	2	23	2	3	2
Darmstadt	2	2	2	2	2		2	2	2			2		2	
Mannheim	2	5	5	4	2	2	2	2	2	2		2	2	2	
Kaiserslautern		2	2	3				2							
Giessen	2		2	2				2				2		2	
Trier		2	2	2				2							
Fulda			2	2				2				2	2	2	
Saarbruecken			2	2			2	2	2			2			
Stuttgart				35	11	4	6	30	2	2	2	35	2	2	2
Karlsruhe		101	62		2	2	5	4	2	2		2	2	2	
Ulm			72			2		2	2	2		2		2	
Konstanz			123				3	5	2	2			2		
Freiburg				119		107		2							
Muenchen									20	2	5	32	2	3	2
Augsburg					66			55		2		2	2	2	
Kempten						85		103							
Passau								148				2	2	2	2
Nuernberg								149					5	10	9
Bayreuth												67		3	3
Wuerzburg	86		125						172			90			
Regensburg								103			108	91			

In Section 5.2, we discuss estimations on unavoidable regenerators, deduced from the length of a shortest path or, in case of connections protected by 1+1 path protection, of a shortest cycle connecting the endnodes of a commodity. For each considered network, Table B.1 lists the maxima over all node pairs of these values together with the optical reach, confirming that no need for regenerations is implied this way.

topology	max. length of a shortest path	max. length of a shortest cycle	optical reach
pt14 (USA)	6940 km	12100 km	8000 km
pt17 (Germany small)	951 km	1976 km	1200 km
pt20 (Austria)	883 km	2155 km	1200 km
pt28 (Europe)	5154 km	13875 km	8000 km
pt50 (Germany large)	970 km	2067 km	1200 km

Table B.1: *The maximum length for connection of any pair of different nodes by a shortest path (also known as network diameter) and by a shortest cycle in each considered network, together with the specified optical reach.*

B.2 Numerical results

In the following, we list the major numerical output data which has been discussed and illustrated in Chapter 5. Clearly, many further information has been generated by the computations, including all obtained optical network designs in full detail as well as computational progress logs. On request, any of these data can be accessed through the author.

B.2.1 Optical network design solutions

We begin with the results for the reference instances listed in the first table. The first two columns specify the instance, indicating the scenario group-wise in the first column (`scenario`) and the requested protection level individually in the second column (`prot. level`). The following two columns show the solution costs split into the cost sum over all nodes (`nodes`) and over all links (`links`) which sum up to the total design cost. The fifth column lists the corresponding cost lower bound value (`lower bound`) which yields the quality guarantee, i.e., no feasible design satisfying all requirements can cost less than this value. The next two columns give the total number of regenerators (`reg.`) and wavelength converters (`conv.`) employed in each computed design. Column eight (`# backup conn.`) holds the total number of established backup connections to achieve the requested protection, i.e., to ensure that each single link or node failure is survived by *at least* as many connections as specified for each demand (and summed up in Table 5.2 on page 191). The last column provides the average hop number per connection (`avg. hops / conn.`).

sce-	prot.	total cost for		lower	total # of		# backup	avg. hops
nario	level	nodes	links	bound	reg.	conv.	conn.	/ conn.
pt14 w1 dsp	s0	21,155.8	32,075.20	51,761.894	1	0	0	2.03
	s50	26,972.2	44,405.40	69,483.005	1	0	30	2.79
	s75	37,817.0	63,521.70	98,728.137	11	0	1,050	2.86
	s100	49,295.6	83,166.40	129,085.545	24	0	2,211	2.86
pt17 w1 dsp	s0	9,668.0	2,963.80	11,742.553	0	0	0	2.38
	s50	12,464.4	3,732.75	15,073.298	15	0	42	3.18
	s75	17,414.0	5,486.45	21,706.382	5	0	575	3.09
	s100	21,703.2	6,923.75	26,989.179	8	0	946	3.15
pt20 w1 dsp	s0	7,412.0	1,707.90	7,964.941	0	25	0	2.38
	s50	9,674.0	2,356.65	9,747.184	0	40	57	3.03
	s75	13,406.4	3,345.60	14,089.507	4	36	384	3.13
	s100	14,966.4	3,739.35	15,981.308	0	39	595	2.97
pt28 w1 dsp	s0	13,517.8	11,734.10	22,687.352	0	9	0	3.34
	s50	20,020.6	19,261.50	35,120.533	1	34	230	4.52
	s75	27,616.6	27,069.95	49,611.817	1	12	783	4.46
	s100	29,704.4	29,090.05	54,273.870	1	53	974	4.40
pt50 w1 dsp	s0	29,020.4	6,726.20	29,086.034	0	0	0	3.00
	s50	34,560.0	8,467.00	33,868.362	2	24	52	3.79
	s75	60,075.6	15,580.10	54,691.036	12	23	1,742	4.21
	s100	62,622.8	15,576.75	60,295.710	20	39	1,880	4.19

In comparison to these references, the first study comprises the evaluation of designs obtained with different survivability concepts, and the numerical results for all alternative schemes are completed by the next table in exactly the same form.

sce-nario	prot. level	total cost for nodes	total cost for links	lower bound	total # of reg.	total # of conv.	# backup conn.	avg. hops / conn.
pt14 w1 max	s0	21,226.8	32,680.50	51,762.662	0	0	0	2.00
	s50	26,362.4	44,500.50	69,633.998	6	0	16	2.79
	s75	36,179.4	69,754.80	103,560.603	179	0	634	3.20
	s100	46,878.6	90,724.30	135,302.176	235	0	1,668	3.19
pt17 w1 max	s0	9,906.4	2,898.90	11,749.571	0	0	0	2.41
	s50	12,464.4	3,785.45	15,207.497	15	0	38	3.22
	s75	17,800.4	5,880.95	22,832.892	42	0	490	3.43
	s100	22,037.6	7,076.25	27,883.544	49	7	873	3.38
pt20 w1 max	s0	7,407.2	1,670.95	7,973.305	0	0	0	2.38
	s50	9,628.0	2,462.85	9,887.559	0	17	34	3.13
	s75	12,872.8	3,481.90	14,487.641	0	26	283	3.51
	s100	14,577.2	3,939.15	16,402.006	2	13	440	3.48
pt28 w1 max	s0	13,416.8	12,085.25	22,723.844	0	0	0	3.32
	s50	19,933.4	19,071.75	35,800.127	1	94	208	4.63
	s75	27,600.4	27,381.45	51,160.547	4	68	691	4.75
	s100	29,463.0	29,838.35	55,738.056	1	52	875	4.63
pt50 w1 max	s0	29,628.8	6,603.15	29,082.200	0	0	0	3.03
	s50	35,432.8	8,483.60	34,024.734	2	0	23	3.77
	s75	55,232.4	14,341.50	58,206.970	20	33	1,040	4.70
	s100	59,650.8	15,592.55	64,430.437	43	129	1,463	4.66
pt14 w1 two	s0	21,226.8	32,680.50	51,762.617	0	0	0	2.00
	s50	26,656.8	44,358.10	69,900.856	0	0	48	2.80
	s75	39,226.8	66,252.20	104,455.944	0	0	1,424	2.79
	s100	51,024.0	86,810.50	136,625.912	0	0	2,710	2.77
pt17 w1 two	s0	9,906.4	2,898.90	11,749.703	0	0	0	2.41
	s50	12,450.4	3,768.75	15,191.697	8	0	49	3.14
	s75	17,410.0	5,613.55	22,126.990	3	0	607	3.09
	s100	21,824.4	6,951.15	27,511.381	9	0	1,021	3.05
pt20 w1 two	s0	7,407.2	1,670.95	7,973.878	0	0	0	2.38
	s50	9,438.8	2,392.45	10,076.163	0	19	88	3.07
	s75	12,944.0	3,323.15	14,693.224	0	39	492	2.93
	s100	15,158.0	3,753.10	16,668.651	3	27	702	2.83
pt28 w1 two	s0	13,416.8	12,085.25	22,722.500	0	0	0	3.32
	s50	20,053.8	19,237.25	35,491.363	2	29	250	4.41
	s75	27,662.6	26,993.75	50,042.087	1	52	806	4.41
	s100	29,705.4	28,987.00	54,695.978	1	78	1,008	4.35
pt50 w1 two	s0	29,628.8	6,603.15	29,082.363	0	0	0	3.03
	s50	35,373.2	8,577.10	34,235.641	0	1	87	3.70
	s75	55,825.2	14,160.05	58,539.587	0	0	1,829	3.74
	s100	61,435.6	15,496.55	64,719.593	0	8	2,365	3.67
pt14 w1 1+1	s0	21,226.8	32,680.50	51,762.848	0	0	0	2.00
	s50	36,846.8	60,466.10	94,889.736	0	0	1,379	2.52
	s75	44,006.2	74,885.80	116,489.476	1	0	2,067	2.67
	s100	51,262.4	86,788.60	136,591.413	0	0	2,710	2.77
pt17 w1 1+1	s0	9,906.4	2,914.95	11,741.661	0	0	0	2.43
	s50	16,035.2	5,091.50	19,723.638	0	0	535	2.85
	s75	19,206.4	6,152.20	24,118.478	1	0	814	2.96
	s100	21,812.4	6,951.15	27,442.906	3	0	1,021	3.06
pt20 w1 1+1	s0	7,407.2	1,670.95	7,974.378	0	0	0	2.38
	s50	12,337.2	3,056.85	12,810.464	0	11	395	2.71
	s75	14,286.8	3,561.35	15,495.029	2	24	597	2.82
	s100	15,180.8	3,792.35	16,652.533	0	27	702	2.84
pt28 w1 1+1	s0	13,416.8	12,085.25	22,727.886	0	0	0	3.32
	s50	23,359.2	23,236.45	42,404.855	3	9	629	4.10
	s75	28,320.4	27,446.05	51,594.082	1	63	907	4.32
	s100	29,560.4	29,276.70	54,654.239	1	5	1,008	4.23
pt50 w1 1+1	s0	29,628.8	6,603.15	29,082.629	0	0	0	3.03
	s50	46,424.4	11,164.15	46,040.330	0	0	1,226	3.42
	s75	58,812.0	14,872.20	58,336.572	0	28	2,097	3.77
	s100	62,353.6	15,740.80	61,698.055	0	21	2,365	3.78

The two tables on this page contain in similar outline the study results on upgrade planning and on the extended hardware setting, where the new columns show the average numbers of spare switch ports per node (`ports`) and spare channels per link (`ch.s`) resp. the number of links with fully occupied capacity (`# full links`).

sce-nario	prot. levels	total cost for nodes	total cost for links	lower bound	total # of reg.	total # of conv.	# backup conn.	avg. spare ports	avg. spare ch.s
pt14 w1 dsp	s0-50	5,919.2	12,898.10	17,638.307	0	0	28	20.14	6.00
	s0-75	17,008.4	31,675.10	45,705.792	12	0	1,081	29.29	6.86
	s0-100	28,025.4	50,977.30	76,110.644	15	0	2,224	17.43	8.00
	s50-75	11,215.8	19,443.40	27,576.980	35	0	1,051	21.86	3.57
	s50-100	22,643.6	37,946.90	57,980.038	16	0	2,253	34.00	2.95
	s75-100	11,804.6	19,330.90	28,064.927	23	0	2,207	32.29	4.76
pt17 w1 dsp	s0-50	3,578.0	1,022.60	3,493.916	5	0	47	24.12	8.96
	s0-75	7,894.0	2,522.65	9,391.857	5	0	568	24.24	2.81
	s0-100	12,321.2	3,923.55	14,660.444	3	0	946	24.82	2.00
	s50-75	5,639.6	1,863.30	5,966.914	2	0	561	21.88	3.54
	s50-100	9,508.8	3,154.60	11,232.394	17	0	952	24.82	1.58
	s75-100	4,511.2	1,376.55	4,344.719	3	0	953	24.82	1.46
pt20 w1 dsp	s0-50	2,866.0	670.30	1,281.533	3	18	50	28.50	6.61
	s0-75	6,239.6	1,658.70	5,327.395	0	4	405	25.90	6.00
	s0-100	8,156.8	1,959.75	7,221.165	0	11	571	30.60	5.48
	s50-75	3,821.2	971.75	2,500.579	4	26	400	23.50	3.91
	s50-100	5,707.6	1,305.80	4,297.185	2	34	558	29.40	3.30
	s75-100	2,364.4	373.05	554.509	2	58	554	32.60	3.70
pt28 w1 dsp	s0-50	7,142.0	7,611.95	10,812.439	1	75	232	29.93	4.59
	s0-75	14,731.8	15,440.40	24,962.793	3	64	774	33.50	4.07
	s0-100	16,725.6	17,589.10	29,463.116	2	50	969	36.86	4.66
	s50-75	8,364.8	7,655.75	11,256.042	4	48	764	34.07	7.10
	s50-100	10,877.0	9,885.45	15,881.097	1	80	965	39.14	3.39
	s75-100	2,679.8	2,057.10	1,832.960	1	40	966	36.86	5.54
pt50 w1 dsp	s0-50	7,394.8	2,006.00	1,673.717	0	4	48	25.84	12.76
	s0-75	25,110.0	6,788.90	21,080.045	7	54	1,395	25.08	8.90
	s0-100	31,026.4	8,177.95	26,877.984	11	63	1,999	29.72	7.95
	s50-75	21,425.2	5,094.70	14,597.520	2	54	1,428	33.40	8.92
	s50-100	25,506.4	6,294.90	20,472.328	6	60	1,928	28.76	9.01
	s75-100	6,005.6	1,016.05	559.741	0	0	2,074	33.08	30.03

sce-nario	prot. level	total cost for nodes	total cost for links	lower bound	total # of reg.	total # of conv.	# backup conn.	avg.sp. ch.s	# full links
pt14 w3 dsp	s0	17,789.4	32,783.10	48,197.735	3	0	0	1.86	9
	s50	23,449.0	44,977.30	65,116.689	3	0	29	2.76	8
	s75	32,268.0	63,776.40	92,717.726	22	0	1,071	3.19	9
	s100	41,324.8	82,838.50	121,253.788	14	0	2,207	3.67	8
pt17 w3 dsp	s0	8,584.8	2,939.90	10,361.191	0	0	0	1.31	14
	s50	11,210.4	3,770.40	12,815.121	0	0	41	1.27	12
	s75	14,956.0	5,623.95	18,824.346	0	0	544	3.65	15
	s100	18,582.0	6,953.90	23,395.994	9	0	978	1.38	15
pt20 w3 dsp	s0	7,003.2	1,679.20	7,097.076	0	2	0	1.30	15
	s50	8,596.4	2,361.10	8,862.384	0	60	48	2.52	14
	s75	11,766.0	3,414.65	12,369.920	3	73	375	1.61	19
	s100	12,750.8	3,800.60	14,129.536	5	32	547	4.06	14
pt28 w3 dsp	s0	12,028.0	12,117.40	20,854.055	0	2	0	2.73	15
	s50	18,481.2	19,288.85	32,616.670	1	97	241	2.12	24
	s75	22,964.2	27,757.30	45,630.636	1	72	761	3.51	14
	s100	25,763.2	29,477.05	49,973.684	1	105	970	1.88	21
pt50 w3 dsp	s0	25,360.4	6,493.30	25,872.429	0	27	0	3.91	30
	s50	30,816.0	8,248.90	30,195.681	0	144	52	3.44	36
	s75	49,794.0	14,740.20	34,098.571	40	182	1,450	4.32	29
	s100	52,512.4	15,328.55	51,959.355	4	208	1,927	2.55	39

Since the values for spare capacities and full links in the previous table are also compared to those for the reference instances, the latter ones are complemented in the next table.

scenario	prot. level	avg. spare ports	avg. spare ch.s	# full links
pt14 w1 dsp	s0	24.43	0.62	14
	s50	33.29	2.95	14
	s75	24.71	5.38	6
	s100	29.43	5.05	7
pt17 w1 dsp	s0	15.00	5.04	11
	s50	26.94	0.65	17
	s75	24.24	2.65	15
	s100	22.47	3.12	17
pt20 w1 dsp	s0	20.70	2.67	24
	s50	28.50	5.58	16
	s75	24.70	4.94	14
	s100	24.20	4.42	11
pt28 w1 dsp	s0	30.00	3.71	20
	s50	26.21	6.80	8
	s75	34.93	7.12	7
	s100	36.86	4.95	14
pt50 w1 dsp	s0	26.18	9.91	25
	s50	24.56	10.53	24
	s75	27.96	14.84	16
	s100	35.32	10.61	27

The last study in Section 5.2.5 on opaque optical networks restricts to cost comparisons, for which the associated table below shows the opaque scenario results in the corresponding subset of already explained columns.

scenario	prot. level	total cost for nodes	total cost for links	lower bound
pt14 opq dsp	s0	21,226.8	48,617.80	68,001.876
	s50	26,537.6	67,583.50	92,277.640
	s75	37,278.4	96,755.70	130,741.764
	s100	48,837.2	125,840.50	171,032.676
pt17 opq dsp	s0	9,787.2	8,080.40	16,866.616
	s50	12,434.4	10,532.75	22,030.080
	s75	17,404.0	15,486.45	31,706.458
	s100	21,687.2	19,359.50	39,392.418
pt20 opq dsp	s0	7,407.2	5,110.95	11,481.505
	s50	9,668.0	7,298.60	14,318.392
	s75	13,416.8	10,314.60	20,692.089
	s100	14,933.6	11,753.55	23,404.826
pt28 opq dsp	s0	13,638.8	22,241.00	32,978.419
	s50	19,960.8	36,731.35	51,780.068
	s75	27,310.4	51,679.00	73,161.695
	s100	29,304.0	55,554.60	80,034.897
pt50 opq dsp	s0	29,480.8	22,311.60	42,967.879
	s50	35,087.6	28,869.35	50,548.949
	s75	55,200.4	47,067.30	82,885.410
	s100	60,346.8	51,206.20	90,170.927

B.2.2 Wavelength assignment data

The second computational study focuses on minimum converter wavelength assignments and is based on an appropriately extended set of 634 instances. The initial assignments for 311 of these instances have been converter-free and thus already optimal, hence we neglect these cases in the following. For each of the remaining 323 instances, the next table subsumes the values evaluated in various ways in Section 5.3. The particular MCWAP instance is specified in the first three columns, with a group-wise scenario indication (`scenario`), the associated protection level (`prot. level`), and the solution number (`sol nr.`) of the particular design. The fourth column (`total cost`) holds the total cost of the initial design of each instance. The next two columns list the total number of converters in the initial design (`init.`) and the assignment returned by the enhanced heuristic (`impr.`). In comparison to this, the following two columns show the lower bounds obtained by solving the linear relaxation of the path packing formulation (`LP`) and by the combinatorial method (`comb.`). Performance measures for the LP solving are shown in the two last columns, which contain the CPU time seconds for the total computation (`total`) and for all pricing steps in sum (`pricing`).

sce-	prot.	sol	total	# conv.		lower bounds		LP seconds	
nario	level	nr.	cost	init.	impr.	LP	comb.	total	pricing
pt14 w1 max	s100	1	29,113.85	7	0	0	0	8.6	2.5
		2	29,121.70	5	0	0	0	8.4	2.4
		3	29,152.10	1	0	0	0	9.3	2.3
		4	29,153.25	8	0	0	0	9.9	2.6
pt20 w1 1+1	s0	2	9,185.35	8	8	8	8	1.0	0.6
		3	9,189.10	1	1	1	1	0.6	0.3
		4	9,226.60	10	10	10	10	1.0	0.6
		5	9,262.10	8	8	8	8	1.1	0.7
	s50	1	15,394.05	11	11	11	11	15.3	6.7
		2	15,140.40	5	5	5	5	12.4	4.7
		3	15,458.35	34	33	32	32	17.9	8.1
		4	15,357.85	6	6	6	6	11.0	5.0
		5	15,172.40	5	5	5	5	13.7	5.0
	s75	1	17,848.15	24	14	12	12	21.8	9.2
		2	17,840.40	20	13	11	11	17.6	7.7
		4	17,696.15	6	1	0	0	10.2	4.4
		5	17,890.95	5	5	2	2	10.2	3.4
	s100	1	18,973.15	27	27	21	21	24.5	10.0
		2	18,779.25	37	18	16	16	21.6	9.4
		3	18,771.75	28	27	14	14	23.9	10.9
		4	19,060.15	45	40	27	27	32.2	11.7
		5	18,964.70	21	21	13	13	24.9	12.2
pt20 w1 two	s0	2	9,185.35	8	8	8	8	1.2	0.6
		3	9,189.10	1	1	1	1	0.7	0.4
		4	9,226.60	10	10	10	10	1.2	0.7
		5	9,262.10	8	8	8	8	1.1	0.7
	s50	1	11,831.25	19	10	10	10	23.5	13.0
		2	11,921.50	64	64	51	51	18.0	8.5
		3	12,011.85	40	38	32	32	55.3	45.9
		4	12,079.70	38	33	33	33	11.8	5.2
		5	12,043.35	8	8	8	8	10.5	5.4
	s75	1	16,267.15	39	36	14	14	23.2	10.0

sce-nario	prot. level	sol nr.	total cost	# conv. init.	# conv. impr.	lower bounds LP	lower bounds comb.	LP seconds total	LP seconds pricing
		2	16,412.80	36	25	6	6	13.8	4.8
		3	16,540.50	37	17	17	17	17.1	7.8
		4	16,571.40	24	19	6	6	22.7	10.8
		5	16,594.05	23	10	10	10	19.8	8.9
	s100	1	18,911.10	27	22	17	17	19.0	9.0
		2	18,949.45	17	9	5	5	14.7	5.9
		3	18,870.45	6	5	2	2	8.7	3.3
		4	19,024.45	35	31	18	18	17.2	10.3
		5	18,926.95	10	8	8	8	22.1	10.0
pt20 w1 max	s0	2	9,185.35	8	8	8	8	1.3	0.7
		3	9,189.10	1	1	1	1	0.7	0.3
		4	9,226.60	10	10	10	10	1.2	0.7
		5	9,262.10	8	8	8	8	1.2	0.6
	s50	1	12,090.85	17	17	17	17	25.3	12.9
		2	11,868.00	26	25	24	24	28.7	15.8
		3	12,185.95	24	24	21	21	28.5	12.1
		4	12,173.65	17	17	17	17	17.4	6.6
		5	12,187.15	3	3	3	3	22.1	9.0
	s75	1	16,354.70	26	21	7	7	87.9	35.1
		2	16,431.20	37	20	3	3	45.9	15.5
		3	16,386.35	14	1	0	0	27.7	11.5
		4	16,435.70	16	14	4	4	48.7	16.9
		5	16,399.35	15	7	7	7	56.3	24.3
	s100	1	18,516.35	13	9	4	4	37.8	14.6
		2	18,719.05	82	67	8	8	84.2	37.2
		3	18,763.80	75	70	23	23	102.6	41.3
		4	18,715.35	36	31	18	18	104.2	61.7
		5	18,501.60	16	16	3	3	49.7	16.7
pt20 w1 dsp	s0	1	9,119.90	25	25	22	22	2.0	1.3
		2	9,094.15	8	8	8	8	1.3	0.8
		3	9,120.45	12	12	10	10	2.1	1.3
		4	9,124.85	6	6	3	2	1.6	0.9
		5	9,200.15	15	15	14	14	1.3	0.9
	s50	1	12,030.65	40	34	33	33	14.7	7.0
		2	12,150.65	62	59	49	49	25.6	13.3
		3	12,120.15	33	20	16	16	24.6	11.6
		4	12,116.45	25	25	23	23	19.6	7.4
		5	12,194.10	41	34	23	23	26.6	11.1
	s75	1	16,752.00	36	36	17	17	50.1	26.5
		2	16,822.50	38	20	6	6	27.2	10.5
		3	16,989.40	51	45	33	33	35.0	16.2
		4	17,015.40	34	34	26	26	51.2	29.0
		5	17,014.10	26	26	9	9	37.7	16.2
	s100	1	18,705.75	39	33	23	23	316.9	297.6
		2	18,795.30	29	24	16	16	2,956.5	2,927.4
		3	18,967.35	57	43	39	39	27.1	11.4
		4	18,968.00	51	41	36	36	28.2	12.2
		5	18,915.15	22	21	16	16	26.6	12.2
pt20 w1 dsp	s0-50	1	3,536.30	18	17	17	17	14.6	6.4
		2	3,808.40	14	13	9	9	21.7	9.0
		3	3,826.45	13	12	11	11	20.1	7.5
		4	3,924.85	24	23	23	23	24.9	12.1
		5	3,927.60	17	17	17	17	223.6	213.1
	s0-75	1	7,898.30	4	4	4	4	24.9	10.1
		2	8,047.75	48	43	27	27	52.2	28.8
		3	8,063.00	38	31	13	11	59.2	31.7
		4	8,121.00	25	25	17	17	249.1	230.3

sce-	prot.	sol	total	# conv.		lower bounds		LP seconds	
nario	level	nr.	cost	init.	impr.	LP	comb.	total	pricing
		5	8,187.75	8	0	0	0	12.8	5.3
	s0-100	1	10,116.55	11	11	10	10	35.4	10.8
		2	10,270.45	45	37	34	34	44.8	19.9
		3	10,343.70	34	30	13	13	35.8	13.8
		4	10,292.85	14	0	0	0	18.6	8.1
		5	10,341.65	34	33	32	32	31.5	15.0
	s50-75	1	4,792.95	26	23	3	3	27.5	8.7
		2	4,953.90	20	10	5	5	29.4	8.7
		3	5,056.85	12	1	0	0	18.1	6.7
		4	5,072.15	14	5	5	5	36.9	13.2
		5	5,137.60	18	6	6	6	37.4	12.1
	s50-100	1	7,013.40	34	24	6	6	34.2	14.1
		2	7,174.75	8	0	0	0	14.8	7.3
		3	7,366.25	33	21	6	6	43.4	15.3
		4	7,328.95	12	5	4	4	38.1	16.6
		5	7,434.40	50	38	31	31	355.2	332.3
	s75-100	1	2,737.45	58	39	24	24	68.4	33.9
		2	3,032.15	46	46	20	20	8,332.6	8,308.5
		3	3,115.35	56	44	27	27	191.4	163.3
		4	3,115.35	51	36	27	27	50.4	33.2
		5	3,091.85	31	31	17	17	2,377.3	2,348.9
pt20 w3 dsp	s0	1	8,682.40	2	2	0	0	0.7	0.4
		2	8,739.65	28	27	26	13	1.3	0.9
		3	8,778.25	8	7	7	6	1.1	0.6
		4	8,783.85	6	6	6	6	1.0	0.6
		5	8,782.10	1	1	1	0	0.9	0.5
	s50	1	10,957.50	60	58	37	37	30.9	19.2
		2	10,948.15	60	60	29	29	26.0	13.7
		3	10,974.25	57	57	30	30	36.6	24.3
		4	10,969.60	54	49	24	24	24.7	14.7
		5	10,999.15	59	59	29	29	48.9	32.9
	s75	1	15,180.65	73	71	33	33	79.4	51.3
		2	15,165.00	48	38	11	11	50.2	28.0
		3	15,116.30	37	36	9	9	61.2	40.4
		4	15,176.85	65	65	24	24	87.3	61.0
		5	15,130.50	42	38	7	7	49.4	29.7
	s100	1	16,551.40	32	20	6	6	47.4	25.2
		2	16,587.40	48	28	2	2	25.9	14.9
		3	16,552.60	32	17	3	3	22.0	11.7
		4	16,583.35	43	22	6	6	44.3	22.0
		5	16,562.35	24	2	2	2	17.0	9.8
pt28 w1 1+1	s0	3	25,535.95	10	10	4	4	34.8	9.2
	s50	1	46,595.65	9	0	0	0	407.8	21.9
		2	46,629.20	9	0	0	0	409.2	30.0
		3	46,639.25	9	0	0	0	423.4	24.1
		4	46,726.90	17	17	0	0	408.7	24.6
		5	46,700.10	5	0	0	0	267.5	21.0
	s75	1	55,766.45	63	62	0	0	677.7	34.8
		2	56,045.20	57	57	0	0	795.2	37.4
		3	56,074.50	21	21	0	0	645.8	34.0
		4	56,158.00	42	42	0	0	553.5	34.6
		5	56,321.60	26	14	0	0	333.8	26.3
	s100	1	58,837.10	5	0	0	0	333.6	24.2
		2	59,194.50	91	90	0	0	850.0	49.1
		3	59,093.20	37	22	0	0	452.2	34.2
		4	59,050.75	15	15	0	0	382.4	30.5
		5	59,117.30	33	33	0	0	582.4	36.0

sce-	prot.	sol	total	# conv.		lower bounds		LP seconds	
nario	level	nr.	cost	init.	impr.	LP	comb.	total	pricing
pt28 w1 two	s0	3	25,535.95	10	10	4	4	39.6	11.0
	s50	1	39,291.05	29	21	0	0	290.8	30.8
		2	39,242.05	13	0	0	0	288.7	30.4
		3	39,378.10	11	8	0	0	269.7	26.9
		4	39,418.05	2	2	0	0	190.4	21.7
		5	39,485.05	24	22	0	0	334.1	33.2
	s75	1	54,656.35	52	49	0	0	456.4	41.2
		2	54,534.40	10	0	0	0	305.6	26.3
		3	54,749.45	26	26	0	0	366.6	31.4
		4	55,055.45	116	116	0	0	795.2	46.5
		5	54,719.80	3	0	0	0	299.7	34.1
	s100	1	58,692.40	78	74	0	0	656.8	38.1
		2	58,860.05	15	12	0	0	357.0	29.2
		3	59,127.95	22	22	0	0	535.2	33.9
		4	59,179.35	4	0	0	0	266.3	24.7
		5	59,216.40	4	0	0	0	309.6	27.2
pt28 w1 max	s0	3	25,535.95	10	10	4	4	40.5	11.3
	s50	1	39,005.15	94	82	0	0	824.3	52.5
		2	38,838.15	90	85	0	0	889.5	62.3
		3	39,210.00	92	77	0	0	847.1	53.7
		4	39,136.70	54	47	0	0	505.0	41.0
		5	39,173.25	53	53	0	0	595.0	51.8
	s75	1	54,981.85	68	68	0	0	1,140.2	60.5
		2	54,996.95	76	76	0	0	1,037.9	61.3
		3	55,036.75	52	52	0	0	835.2	55.7
		4	55,063.95	61	57	0	0	735.9	59.0
		5	55,206.25	98	93	0	0	1,023.2	65.2
	s100	1	59,301.35	52	48	0	0	1,285.9	59.9
		2	59,638.85	43	35	0	0	938.3	54.8
		3	59,715.95	59	50	0	0	812.8	52.5
		4	59,748.20	66	50	0	0	879.1	54.4
		5	59,711.10	51	46	0	0	793.9	48.6
pt28 w1 dsp	s0	1	25,251.90	9	9	4	4	22.3	7.1
		2	25,287.50	4	2	1	1	23.4	7.3
		3	25,395.85	2	1	0	0	17.9	6.3
	s50	1	39,282.10	34	34	0	0	360.3	37.6
		2	39,295.70	24	24	0	0	319.1	35.0
		3	39,401.55	43	43	0	0	344.6	30.8
		4	39,681.50	113	113	0	0	673.3	55.1
		5	39,780.60	130	130	0	0	1,106.9	62.4
	s75	1	54,686.55	12	12	0	0	334.2	29.9
		2	55,051.70	61	61	0	0	399.2	34.3
		3	54,626.35	46	46	0	0	557.0	37.7
		4	55,206.05	16	16	0	0	528.8	37.8
		5	55,265.65	3	0	0	0	392.4	39.4
	s100	1	58,794.45	53	53	0	0	440.3	36.7
		2	58,895.15	12	0	0	0	403.2	33.6
		3	59,021.05	49	49	0	0	576.2	37.4
		4	59,303.90	80	80	0	0	576.9	43.9
		5	59,170.30	27	27	0	0	482.4	33.2
pt28 w1 dsp	s0-50	1	14,753.95	75	69	0	0	906.6	47.0
		2	14,832.95	46	36	0	0	460.3	40.5
		3	14,853.65	46	33	0	0	512.5	37.8
		4	14,985.80	32	29	0	0	473.7	41.2
		5	14,908.70	5	0	0	0	404.0	32.5
	s0-75	1	30,172.20	64	64	0	0	561.1	33.8
		2	30,382.15	78	78	0	0	497.7	37.8

sce-	prot.	sol	total	# conv.		lower bounds		LP seconds	
nario	level	nr.	cost	init.	impr.	LP	comb.	total	pricing
		3	30,427.35	88	88	0	0	825.1	42.9
		4	30,563.85	17	13	0	0	495.4	37.6
		5	30,419.90	2	0	0	0	394.2	33.5
	s0-100	1	34,314.70	50	50	0	0	620.5	38.2
		2	34,526.30	42	42	0	0	658.3	37.4
		3	34,629.10	68	66	0	0	715.6	39.0
		4	34,599.60	34	30	0	0	435.0	30.5
		5	34,725.90	49	46	0	0	737.4	39.4
	s50-75	1	16,020.55	48	40	0	0	440.8	35.4
		2	15,957.05	21	21	0	0	466.9	39.2
		3	16,312.65	28	28	0	0	436.8	34.7
		4	16,429.85	20	20	0	0	459.2	35.1
		5	16,473.55	27	19	0	0	539.7	34.1
	s50-100	1	20,762.45	80	78	0	0	835.2	44.6
		2	20,531.90	91	88	0	0	671.2	45.9
		3	21,543.65	106	96	0	0	1,249.0	52.4
		4	21,459.75	78	78	0	0	815.5	48.5
		5	21,390.35	31	30	0	0	421.4	35.8
	s75-100	1	4,736.90	40	31	0	0	623.4	37.6
		2	4,707.75	27	24	0	0	524.6	36.5
		3	4,734.90	12	0	0	0	420.6	31.5
		4	4,978.85	43	43	0	0	608.1	39.6
		5	4,995.00	46	31	0	0	641.7	38.2
pt28 w3 dsp	s0	1	24,145.40	2	0	0	0	16.2	6.5
		2	24,280.95	17	1	0	0	16.7	6.9
		3	24,395.60	12	10	0	0	15.8	6.9
		5	24,461.65	10	10	6	0	14.5	5.5
	s50	1	37,770.05	97	91	0	0	658.9	62.0
		2	37,911.60	108	102	0	0	423.2	64.5
		3	37,866.85	67	67	0	0	386.5	66.5
		4	37,840.50	60	51	0	0	364.0	60.8
		5	38,059.45	85	85	0	0	348.6	62.3
	s75	1	50,721.50	72	65	0	0	485.9	66.5
		2	51,018.60	77	66	0	0	425.3	62.6
		3	50,861.95	45	45	0	0	358.7	54.7
		4	50,990.05	73	68	0	0	427.7	60.9
		5	50,911.70	87	78	0	0	469.5	68.5
	s100	1	55,240.25	105	81	0	0	573.9	65.5
		2	55,166.85	63	63	0	0	548.1	66.8
		3	55,350.70	102	89	0	0	526.6	60.4
		4	55,383.90	88	81	0	0	488.8	65.6
		5	55,428.85	92	66	0	0	555.7	65.3
pt50 w1 1+1	s75	1	73,684.20	28	0	0	0	3,549.7	402.8
		2	74,601.45	6	0	0	0	2,458.8	295.0
		3	74,608.60	4	0	0	0	3,007.2	329.3
		4	75,556.70	23	0	0	0	3,972.6	373.2
	s100	1	78,094.40	21	5	0	0	2,052.6	260.4
		2	79,006.55	9	0	0	0	4,191.8	369.2
		3	78,775.10	13	0	0	0	4,002.5	337.9
		4	80,108.20	11	0	0	0	4,875.2	458.9
pt50 w1 two	s50	1	43,950.30	1	1	0	0	1,545.8	176.3
		3	43,555.95	1	0	0	0	1,283.3	150.0
		4	43,368.90	5	0	0	0	1,406.9	189.3
		5	43,601.10	4	4	0	0	1,866.2	216.5
	s75	2	69,453.60	13	10	0	0	1,812.7	246.0
		3	70,523.30	17	0	0	0	1,473.1	215.0
		4	70,627.65	26	9	0	0	1,720.5	196.4

sce-	prot.	sol	total	# conv.		lower bounds		LP seconds	
nario	level	nr.	cost	init.	impr.	LP	comb.	total	pricing
		5	70,314.35	5	0	0	0	1,326.6	199.9
	s100	1	76,932.15	8	0	0	0	1,611.0	191.8
		2	77,216.65	12	0	0	0	2,273.4	259.9
		3	76,248.75	14	0	0	0	1,675.5	209.3
		4	77,217.35	8	0	0	0	1,549.8	227.0
		5	77,665.15	16	0	0	0	2,679.6	267.1
pt50 w1 max	s75	1	69,573.90	33	33	0	0	11,071.0	639.6
		2	70,297.45	39	24	0	0	21,870.4	987.7
		3	69,902.00	37	33	0	0	12,251.2	628.4
		4	69,320.75	64	46	0	0	13,129.5	648.8
		5	70,214.95	45	38	0	0	12,099.0	615.4
	s100	1	75,243.35	129	107	0	0	25,581.8	1,060.7
		2	76,658.85	74	34	0	0	12,590.4	800.4
		3	76,250.75	79	63	0	0	11,738.6	624.8
		4	75,465.25	95	60	0	0	26,697.8	1,069.8
		5	76,405.20	105	69	0	0	21,510.7	975.5
pt50 w1 dsp	s0	4	35,958.05	13	3	0	0	172.0	30.6
	s50	1	43,027.00	24	11	0	0	2,151.4	288.2
		2	43,683.75	2	0	0	0	1,272.2	173.9
		3	43,710.00	4	0	0	0	1,625.0	223.2
		4	43,547.65	14	14	0	0	1,952.5	208.1
	s75	1	75,655.70	23	0	0	0	3,375.6	439.5
		2	77,644.90	43	29	0	0	7,835.2	907.2
		3	76,585.30	13	0	0	0	2,457.8	291.3
	s100	1	78,199.55	39	24	0	0	8,564.3	641.4
pt50 w1 dsp	s0-50	1	9,400.80	4	0	0	0	2,141.4	222.2
	s0-75	1	31,898.90	54	53	0	0	7,098.0	544.8
		2	33,777.70	49	39	0	0	6,515.3	524.2
		3	34,216.15	47	47	0	0	8,872.4	609.3
	s0-100	1	39,204.35	63	35	0	0	7,562.5	582.9
		2	39,459.40	23	16	0	0	4,525.9	438.2
		3	39,587.40	18	0	0	0	3,659.1	298.2
		4	39,604.25	17	6	0	0	6,883.0	429.2
	s50-75	1	26,519.90	54	52	0	0	7,799.2	667.4
		2	26,022.45	12	0	0	0	5,867.7	487.5
		3	26,800.50	25	23	0	0	4,422.5	441.8
		4	27,517.50	61	58	0	0	9,159.3	632.4
		5	27,752.70	25	22	0	0	6,037.3	586.2
	s50-100	1	31,801.30	60	27	0	0	6,326.3	534.9
		2	33,060.20	28	28	0	0	6,202.9	476.9
		3	32,373.55	20	7	0	0	5,698.2	487.6
		4	32,769.40	43	41	0	0	7,667.5	571.0
		5	32,789.25	32	16	0	0	6,794.7	477.6
pt50 w3 dsp	s0	1	31,853.70	27	23	0	0	191.9	64.5
		2	31,865.20	24	16	0	0	143.0	52.7
		3	31,988.15	18	7	0	0	128.3	54.2
		4	31,774.50	1	0	0	0	69.2	29.4
		5	32,168.95	26	26	4	4	552.1	131.8
	s50	1	39,064.90	144	143	0	0	2,801.3	489.2
		2	38,951.60	108	108	0	0	2,451.5	441.0
		3	38,673.75	122	122	0	0	2,934.3	502.3
		4	39,625.20	125	125	0	0	2,548.7	452.7
		5	39,227.80	161	160	0	0	2,931.1	512.0
	s75	1	64,534.20	182	174	0	0	14,447.8	1,449.8
		2	66,169.05	156	156	0	0	5,454.2	863.1
		3	66,979.15	145	134	0	0	5,869.3	1,016.7
		4	67,063.25	176	176	0	0	7,695.6	1,160.0

sce-	prot.	sol	total	# conv.		lower bounds		LP seconds	
nario	level	nr.	cost	init.	impr.	LP	comb.	total	pricing
		5	66,261.35	173	169	0	0	4,804.5	870.5
	s100	1	67,840.95	208	205	0	0	11,508.3	1,419.6
		2	70,199.45	162	162	0	0	10,531.8	1,141.9
		3	72,008.25	188	181	0	0	8,211.0	1,350.9
		4	72,244.00	193	177	0	0	11,369.7	1,493.6
		5	71,860.30	176	165	0	0	5,959.5	939.2

The last table documents the computations with the branch-and-price method which has been carried out for 30 instances with a gap of less than three between the LP lower bound and the best solution known. These instances are specified in the first five columns (`instance`). Next, two columns show the converter number in the so far best known assignment used for initialization (`init`) and in the assignment returned by the branch-and-price method (`final`), highlighting improvements in bold face. For simpler comparison, the next column repeats the previously reported LP lower bound (`LP bound`), and here bold values indicate new optimal solutions. The last three columns document the performance of the branch-and-price method, showing the number of visited search tree nodes (`nodes`) and those still unfinished (`left`) as well as the total CPU time in seconds (`time (sec.)`).

					# converters		LP	B&P performance		
		instance			init	final	bound	nodes	left	time (sec.)
pt20	w1	1+1	s50	sol3	33	33	32	19	16	14,411.8
pt20	w1	1+1	s75	sol1	14	14	12	1	1	14,422.5
pt20	w1	1+1	s75	sol2	13	13	11	1	1	14,408.3
pt20	w1	1+1	s75	sol4	1	1	0	1	0	150.9
pt20	w1	1+1	s75	sol5	5	5	2	1	1	14,413.0
pt20	w1	1+1	s100	sol2	18	18	16	1	1	14,412.5
pt20	w1	two	s100	sol3	5	5	2	4	4	14,405.6
pt20	w1	max	s50	sol2	25	25	24	8	7	14,411.4
pt20	w1	max	s50	sol3	24	24	21	1	1	14,415.3
pt20	w1	max	s75	sol3	1	1	0	5	4	14,408.2
pt20	w1	dsp	s0	sol1	25	**23**	22	497	0	4,102.6
pt20	w1	dsp	s0	sol3	12	**11**	10	40	0	420.6
pt20	w1	dsp	s0	sol4	6	**4**	3	139	0	670.4
pt20	w1	dsp	s0	sol5	15	15	14	1757	889	14,403.3
pt20	w1	dsp	s50	sol1	34	34	33	14	11	14,403.5
pt20	w1	dsp	s50	sol4	25	25	23	1	1	14,403.9
pt20	w1	dsp	s0-50	sol3	12	**11**	**11**	2	0	2,753.3
pt20	w1	dsp	s0-100	sol1	11	11	10	25	20	14,421.9
pt20	w1	dsp	s0-100	sol2	37	37	34	1	1	14,413.2
pt20	w1	dsp	s0-100	sol5	33	33	32	17	14	14,432.5
pt20	w1	dsp	s50-75	sol3	1	1	0	13	11	14,409.0
pt20	w1	dsp	s50-100	sol4	5	5	4	1	1	14,405.1
pt20	w3	dsp	s0	sol1	2	2	0	1	0	3.4
pt20	w3	dsp	s0	sol2	27	**26**	**26**	357	0	11,206.9
pt28	w1	two	s50	sol4	2	2	0	4	4	14,411.9
pt28	w1	dsp	s0	sol2	2	2	1	3	2	14,421.5
pt28	w1	dsp	s0	sol3	1	1	0	5	4	14,408.6
pt28	w3	dsp	s0	sol2	1	1	0	5	4	14,413.1
pt50	w1	two	s50	sol1	1	1	0	1	1	14,524.3
pt50	w1	dsp	s0	sol4	3	3	0	1	1	14,417.9

Bibliography

[1] K.I. Aardal, C.P.M. van Hoesel, A.M.C.A. Koster, C. Mannino, and A. Sassano. Models and solution techniques for the frequency assignment problem. *4OR*, 1(4):261–317, 2003.

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[3] D. Alevras, M. Grötschel, and R. Wessäly. A network dimensioning tool. ZIB report ZR-96-49, Zuse Institute Berlin (ZIB), 1996.

[4] D. Alevras, M. Grötschel, and R. Wessäly. Capacity and survivability models for telecommunications networks. In *Plenaries and Tutorials: Proc. of Joint 15th EURO/34th INFORMS*, Barcelona, Spain, 1997.

[5] M. Andrews and L. Zhang. Wavelength assignment in optical networks with fixed fiber capacity. In *Proc. of ICALP*, number 3142 in Lecture Notes in Computer Science, pages 134–145. Springer-Verlag, 2004.

[6] M. Andrews and L. Zhang. Complexity of wavelength assignment in optical network optimization. In *Proc. of IEEE INFOCOM*, Barcelona, Spain, 2006.

[7] K. Appel and W. Haken. Every planar map is four colorable. Part I. Discharging. *Illinois J. Math.*, 21:429–490, 1977.

[8] K. Appel, W. Haken, and J. Koch. Every planar map is four colorable. Part II. Reducibility. *Illinois J. Math.*, 21:491–567, 1977.

[9] atesio GmbH. DISCNET. http://www.atesio.de/solutions.

[10] V. Auletta, I. Caragiannis, L. Gargano, C. Kaklamanis, and P. Persiano. Sparse and limited wavelength conversion in all-optical tree networks. *Theoretical Computer Science*, 266(1–2):887–934, 2001.

[11] V. Auletta, I. Caragiannis, C. Kaklamanis, and P. Persiano. Efficient wavelength routing in trees with low-degree converters. In *Multichannel Optical Networks: Theory and Practice*, volume 46 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 1–14. American Mathematical Society, 1998.

[12] B. Beauquier, J.-C. Bermond, L. Gargano, P. Hell, S. Pérennes, and U. Vaccaro. Graph problems arising from wavelength-routing in all-optical networks. In *Proc. of 2nd WOCS*, 1997.

[13] P. Belotti. *Multicommodity network design with survivability constraints: Some models and algorithms.* PhD thesis, Politecnico di Milano, 2002.

[14] P. Belotti and T. Stidsen. Optimal placement of wavelength converting nodes. In *Proc. of DRCN*, Budapest, Hungary, 2001.

[15] J. Benders. *Partitioning in mathematical programming.* PhD thesis, University of Utrecht, 1960.

[16] J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[17] A. Betker, C. Gerlach, R. Hülsermann, M. Jäger, M. Barry, S. Bodamer, J. Späth, C. Gauger, and M. Köhn. Reference transport network scenarios. Technical report, BMBF MultiTeraNet, 2003. http://www.ikr.uni-stuttgart.de/IKRSimLib/Referenz_Netze_v14_full.pdf.

[18] R. Bhandari. *Survivable Networks, Algorithms for Diverse Routing.* Kluwer Academic Publishers, 1999.

[19] D. Bienstock, S. Chopra, O. Günlük, and C.-Y. Tsai. Minimum cost capacity installation for multicommodity flows. *Mathematical Programming*, 81:177–199, 1998.

[20] D. Bienstock and G. Muratore. Strong inequalities for capacitated survivable network design problems. *Mathematical Programming*, 89:127–147, 2001.

[21] G.A. Birkan, J.L. Kennington, E.V. Olinick, A. Ortynski, and G. Spiride. Making a case for using integer programming to design DWDM networks. *Optical Networks Magazine*, 4(6):107–120, 2003.

[22] G.A. Birkan, J.L. Kennington, E.V. Olinick, A. Ortynski, and G. Spiride. Robust solutions for the WDM routing and provisioning problem: Models and algorithms. *Optical Networks Magazine*, 4(2):74–84, 2003.

[23] G.A. Birkan, J.L. Kennington, E.V. Olinick, A. Ortynski, and G. Spiride. Practical integrated design strategies for opaque and all-optical DWDM networks: Optimization models and solution procedures. *Telecommunication Systems*, 31:61–83, 2006.

[24] A. Bley, A. Kröller, A.M.C.A. Koster, R. Wessäly, and A. Zymolka. Kosten- und Qualitätsoptimierung in Kommunikationsnetzen. *TeleKommunikation Aktuell*, 57(07+08):1–62, 2003.

[25] L. Brunetta, F. Malucelli, P. Värbrand, and D. Yuan. Joint optical network design, routing and wavelength assignment by integer programming. *Telecommunication Systems*, 26(1):53–67, 2004.

[26] H. Buchta, E. Patzak, J. Saniter, and C. Gauger. Maximal and effective throughput of optical switching nodes for optical burst switching. In *4. ITG-Fachtagung Photonische Netze*, pages 1–10, Fachhochschule der Deutschen Telekom, Leipzig, Germany, 2003.

[27] I. Caragiannis, A. Ferreira, C. Kaklamanis, S. Pérennes, and H. Rivano. Fractional path coloring with applications to WDM networks. In *Proc. of 28th ICALP*, Lecture Notes in Computer Science 2076, pages 732–743. Springer-Verlag, 2001.

[28] I. Caragiannis and C. Kaklamanis. Approximate path coloring with applications to wavelength assignment in WDM optical networks. In *Proc. of 21th STACS*, Lecture Notes in Computer Science 2996, pages 258–269. Springer-Verlag, 2004.

[29] I. Caragiannis, C. Kaklamanis, and P. Persiano. Wavelength routing in all-optical tree networks: A survey. *Computers and Informatics*, 20(2):95–120, 2001.

[30] D. Cavendish, A. Kolarov, and B. Sengupta. Minimizing the number of wavelength conversions in WDM networks with hybrid optical cross-connects. In *Proc. of INFORMS Telecommunications Conference*, pages 203–224, Boca Raton, USA, 2004.

[31] X. Chu, B. Li, and I. Chlamtac. Wavelength converter placement under different RWA algorithms in wavelength-routed optical networks. *IEEE Transactions on Communications*, 51(4):607–617, 2003.

[32] V. Chvátal. *Linear Programming*. Freeman and Company, 1983.

[33] D. Colle. *Design and Evolution of Data-centric Optical Networks*. PhD thesis, Universiteit Ghent, 2002.

[34] D. Coudert and H. Rivano. Lightpath assignment for multifiber WDM networks with wavelength translators. In *Proc. of GLOBECOM*, volume 3, pages 2686–2690, 2002.

[35] G. Dahl and M. Stoer. A polyhedral approach to multicommodity survivable network design. *Numerische Mathematik*, 68(1):149–167, 1994.

[36] G. Dahl and M. Stoer. A cutting plane algorithm for multicommodity survivable network design problems. *INFORMS Journal on Computing*, 10(1):1–11, 1998.

[37] J. Desrosiers and M.E. Lübbecke. A primer in column generation. In *Column Generation*, pages 1–32. Kluwer Academic Publishers, 2005.

[38] R. Diestel. *Graph Theory (Second Edition)*. Springer-Verlag, 2000.

[39] T. Van Do, R. Chakka, and Z. Pandi. Novel analysis method for optical packet switching nodes. In *Proc. of ONDM*, volume II, pages 755–767, Budapest, Hungary, 2003.

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

[41] A. Ehrhardt. Next generation optical networks: an operator's point of view. In *Proc. of 8th ICTON*, volume 1, pages 93–97, Nottingham, United Kingdom, 2006.

[42] A. Eisenblätter. *Frequency Assignment in GSM Networks—Models, Heuristics, and Lower Bounds*. PhD thesis, Technische Universität Berlin, 2001.

[43] T. Erlebach and K. Jansen. Call scheduling in trees, rings and meshes. In *Proc. of HICSS*, pages 221–222, 1997.

[44] T. Erlebach and K. Jansen. The complexity of path coloring and call scheduling. *Theoretical Computer Science*, 255(1–2):33–50, 2001.

[45] T. Erlebach and S. Stefanakos. Wavelength conversion in all-optical networks with shortest-path routing. *Algorithmica*, 43:43–61, 2005.

[46] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4):691–703, 1976.

[47] S. Ferguson. Optical switching technologies and switching in optical future networks. In *Proc. of 8th ICTON*, volume 1, pages 106–107, Nottingham, United Kingdom, 2006.

[48] A. Ferreira, S. Pérennes, H. Rivano, A.W. Richa, and N. Stier Moses. Models, complexity and algorithms for the design of multi-fiber WDM networks. *Telecommunication Systems*, 24(2–4):123–138, 2003.

[49] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1–3):23–47, 2003.

[50] L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[51] T. Freeman. 40G moves back onto the agenda. *Fibre Systems Europe*, pages 23–24, 2004.

[52] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-completeness*. Freeman and Company, 1979.

[53] M.R. Garey, D.S. Johnson, G. Miller, and C. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal on Discrete Mathematics*, 1(2):216–227, 1980.

[54] O. Gerstel and R. Ramaswami. Optical layer survivability: A services perspective. *IEEE Communications Magazine*, pages 104–113, 2000.

[55] O. Gerstel and R. Ramaswami. Optical layer survivability: An implementation perspective. *IEEE Journal on Selected Areas in Communications*, 18:1885–1899, 2000.

[56] A. Groebbens, D. Colle, C. Develder, S. de Maessschalck, M. Pickavet, and P. Demeester. Use of backup trees to improve resource efficiency of MPLS protection mechanisms. In *Proc. of DRCN*, pages 152–159, Budapest, Hungary, 2001.

[57] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[58] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Number 2 in Algorithms and Combinatorics. Springer-Verlag, 1988.

[59] M. Grötschel, C.L. Monma, and M. Stoer. Design of survivable networks. In *Handbooks in Operations Research and Management Science*, pages 617–672. North-Holland, 1995.

[60] W.D. Grover. The selfhealing network: A fast distributed restoration technique for networks using digital cross-connect machines. In *Proc. of IEEE Globecom*, pages 1090–1095, Tokyo, Japan, 1987.

[61] W.D. Grover. *Mesh-Based Survivable Networks*. Prentice Hall, 2004.

[62] C. Gruber, A.M.C.A. Koster, S. Orlowski, R. Wessäly, and A. Zymolka. A computational study for demand-wise shared protection. In *Proc. of DRCN*, pages 421–428, Isle of Ischia, Naples, Italy, 2005.

[63] O. Günlük. A branch-and-cut algorithm for capacitated network design. *Mathematical Programming*, 86:17–39, 1999.

[64] S.L. Hakimi and O. Kariv. On a generalization of edge coloring in graphs. *Journal of Graph Theory*, 10:139–154, 1986.

[65] G.Y. Handler and I. Zang. A dual algorithm for the constrained shortest path problem. *Networks*, 10:293–310, 1980.

[66] E.J. Harder. *Routing and wavelength assignment in all-optical WDM wavelength-routed networks*. PhD thesis, George Washington University, 1998.

[67] D.S. Hochbaum. *Approximation Algorithms for $\mathcal{NP}$-hard Problems*. PWS Publishing Co., 1997.

[68] I. Holyer. The $\mathcal{NP}$-completeness of edge-coloring. *SIAM Journal on Computing*, 10:718–720, 1981.

[69] X. Hu and T. Shuai. Wavelength assignment for satisfying maximal number of requests in all-optical networks. In *Proc. of 1st AAIM*, volume 3521 of *Lecture Notes in Computer Science*, pages 320–329. Springer-Verlag, 2005.

[70] R. Hülsermann, M. Jäger, A.M.C.A Koster, S. Orlowski, R. Wessäly, and A. Zymolka. Availability and cost based evaluation of demand-wise shared protection. In *Tagungsband der 7. ITG-Fachtagung Photonische Netze*, pages 161–168, Fachhochschule der Deutschen Telekom, Leipzig, Germany, 2006.

[71] ILOG. CPLEX version 9.130, 2005. http://www.ilog.com/products/cplex.

[72] M. Iri. On an extension of the maximum-flow minimum-cut theorem to multicommodity flows. *Journal of the Operations Research Society of Japan*, 13(3):129–135, 1971.

[73] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48:761–777, 2001.

[74] B. Jaumard, C. Meyer, and B. Thiongane. Comparison of ILP formulations for the RWA problem. Les cahiers de GERAD G–2004–66, GERAD, 2004.

[75] B. Jaumard, C. Meyer, and B. Thiongane. ILP formulations for the RWA problem — symmetric systems. Les cahiers de GERAD G–2004–95, GERAD, 2004.

[76] B. Jaumard, C. Meyer, and B. Thiongane. On column generation formulations for the RWA problem. Les cahiers de GERAD G–2004–94, GERAD, 2004.

[77] B. Jaumard, C. Meyer, and B. Thiongane. On column generation formulations for the RWA problem. In *Proc. of INOC*, volume 1, pages B.52–B.59, Lisbon, Portugal, 2005.

[78] B. Jaumard, C. Meyer, and X. Yu. Of how much wavelength conversion allows to reduce the blocking rate?, 2005.

[79] T.R. Jensen and B. Toft. *Graph Coloring Problems*. Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., 1995.

[80] B.G. Josza and D. Orincsay. Shared backup path optimization in telecommunication networks. In *Proc. of DRCN*, pages 251–257, Budapest, Hungary, 2001.

[81] S. Junghans and Ch. M. Gauger. Architectures for resource reservation modules for optical burst switching core nodes. In *Tagungsband der 4. ITG-Fachtagung Photonische Netze*, pages 109–117, Fachhochschule der Deutschen Telekom, Leipzig, 2003.

[82] C. Kaklamanis. Recent advances in wavelength routing. In *Proc. of SOFSEM*, volume 2234 of *Lecture Notes in Computer Science*, pages 58–72. Springer-Verlag, 2001.

[83] E. Karasan, O.E. Karasan, and G. Erdogan. Optimum placement of wavelength interchanging nodes in optical networks with sparse conversion. In *Proc. of 9th International Conference on Networks & Optical Communications*, Eindhoven, Netherlands, 2004.

[84] R.M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[85] R.M. Karp. On the complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

[86] J.L. Kennington, E.V. Olinick, A. Ortynski, and G. Spiride. Wavelength routing and assignment in a survivable WDM mesh network. *Operations Research*, 51(1):67–79, 2003.

[87] H. Kerivin, B. Liau, and T.-T.-L. Pham. Survivable capacitated networks - Comparison of shared protection mechanisms. In *Proc. of Networks*, pages 379–388, Munich, Germany, 2002.

[88] J. Kleinberg and A. Kumar. Wavelength conversion in optical networks. In *Proc. of 10th SODA*, pages 566–575, Baltimore, USA, 1999.

[89] M. Kneser. Aufgabe 360. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 58:27, 1955.

[90] A.M.C.A. Koster. *Frequency Assignment—Models and Algorithms*. PhD thesis, Universiteit Maastricht, 1999.

[91] A.M.C.A. Koster. Wavelength assignment in multi-fiber WDM networks by generalized edge coloring. In *Proc. of INOC*, pages 60–66, Lisbon, Portugal, 2005.

[92] A.M.C.A. Koster and A. Zymolka. Stable multi-sets. *Mathematical Methods of Operations Research*, 56(1):45–65, 2002.

[93] A.M.C.A. Koster and A. Zymolka. Minimum converter wavelength assignment in all-optical networks. In *Proc. of ONDM*, pages 517–535, Ghent, Belgium, 2004.

[94] A.M.C.A. Koster and A. Zymolka. Linear programming lower bounds for minimum converter wavelength assignment in optical networks. In *Proc. of INOC*, pages 44–51, Lisbon, Portugal, 2005.

[95] A.M.C.A. Koster and A. Zymolka. On cycles and the stable multi-set polytope. *Discrete Optimization*, 2(3):241–255, 2005.

[96] A.M.C.A. Koster and A. Zymolka. Provably good solutions for wavelength assignment in optical networks. In *Proc. of ONDM*, pages 335–345, Milan, Italy, 2005.

[97] A.M.C.A. Koster, A. Zymolka, M. Jäger, and R. Hülsermann. Demand-wise shared protection for meshed optical networks. *Journal of Network and Systems Management*, 13(1):35–55, 2005.

[98] A.M.C.A. Koster, A. Zymolka, M. Jäger, R. Hülsermann, and C. Gerlach. Demand-wise shared protection for meshed optical networks. In *Proc. of DRCN*, pages 85–92, Banff, Canada, 2003.

[99] A. Kröller. Network optimization: Integration of hardware configuration and capacity dimensioning. Master's thesis, Technische Universität Berlin, 2003.

[100] M. Lackovic and C. Bungerzeanu. An approach to planning and performance evaluation of networks based on optical packet switching. In *Proc. of ONDM*, volume II, pages 785–797, Budapest, Hungary, 2003.

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

[102] G. Li and R. Simha. On the wavelength assignment problem in multifiber WDM star and ring networks. In *Proc. of INFOCOM*, pages 1771–1780, 2000.

[103] G. Li and R. Simha. The partition coloring problem and its application to wavelength routing and assignment. In *Optical Networks Workshop*, Richardson, USA, 2000.

[104] G. Li and R. Simha. On the wavelength assignment problem in multifiber WDM star and ring networks. *IEEE/ACM Transactions on Networking*, 9(1):60–68, 2001.

[105] Light Reading, Inc. (Byte and Switch). Alcatel holds world record for a day. http://www.byteandswitch.com/document.asp?doc_id=4380&print=true, March 2001. Press Release.

[106] F. Lorentz. *Lineare Algebra*. Spektrum Akademischer Verlag, 2003.

[107] L. Lovász. Kneser's conjecture, chromatic numbers, and homotopy. *Journal of Combinatorial Theory, Series A*, 25:319–324, 1978.

[108] H. Marchand and L.A. Wolsey. Aggregation and mixed integer rounding to solve MIPs. Technical Report 9839, CORE, 1998.

[109] L. Margara and J. Simon. Wavelength assignment problem on all-optical networks with k fibres per link. In *Proc. of ICALP*, volume 1853 of *Lecture Notes in Computer Science*, pages 768–779. Springer-Verlag, 2000.

[110] L. Margara and J. Simon. Decidable properties of graphs of all-optical networks. In *Proc. of ICALP*, volume 2076 of *Lecture Notes in Computer Science*, pages 518–529. Springer-Verlag, 2001.

[111] J. Matoušek and G.M. Ziegler. Topological lower bounds for the chromatic number: A hierarchy. *Jahresbericht der DMV*, 106:71–90, 2004.

[112] C. Mauz. Allocation of spare capacity for shared protection of optical paths in transport networks. In *Proc. of DRCN*, pages 22–27, Budapest, Hungary, 2001.

[113] A. Meddeb, A. Girard, and B. Sanso. Design model for minimum cost flat hybrid optoelectronic networks. In *Proc. of 5th INFORMS Telecommunication Systems*, pages 31–32, 2000.

[114] K. Mehlhorn, S. Näher, M. Seel, and C. Uhrig. LEDA version 4.1, 2000. http://www.mpi-sb.mpg.de/LEDA/MANUAL/MANUAL.html.

[115] A. Mehrotra and M.A. Trick. A column generation approach for graph coloring. *INFORMS Journal on Computing*, 8(4):344–354, 1996.

[116] B. Mélián, M. Laguna, and J.A. Moreno-Pérez. Capacity expansion of fiber optic networks with WDM systems: Problem formulation and comparative analysis. *Computers and Operations Research*, 31(3):461–472, 2004.

[117] B. Mélián, M. Laguna, and J.A. Moreno-Pérez. Minimizing the cost of placing and sizing wavelength division multiplexing and optical cross-connect equipment in a telecommunications network. *Networks*, 45(4):199–209, 2005.

[118] K. Menger. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927.

[119] M. Minoux. Optimum synthesis of a network with non-simultaneous multicommodity flow requirements. In *Studies on Graphs and Discrete Programming*, pages 269–277. North-Holland Publishing Company, 1981.

[120] M. Minoux. Discrete cost multicommodity network optimization problems and exact solution methods. *Annals of Operations Research*, 106:19–46, 2001.

[121] M.S.O. Molloy and B. Reed. *Graph Coloring and the Probabilistic Method*, volume 23 of *Algorithms and Combinatorics*. Springer-Verlag, 2002.

[122] A. Morea and J. Poirrier. A critical analysis of the possible cost savings of translucent networks. In *Proc. of DRCN*, pages 311–317, Isle of Ischia, Naples, Italy, 2005.

[123] B. Mukherjee. *Optical Communication Networks*. McGraw-Hill, 1997.

[124] MultiTeraNet. Bmbf projecthttp://www.dlr.de/pt_it/kt/foerderbereiche/photonische_kommunikationsnetze/multiteranet.

[125] S.-I. Nakano, T. Nishizeki, and N. Saito. On the f-coloring of multigraphs. *IEEE Transactions on Circuits and Systems*, 35(3):345–353, 1988.

[126] NEC. NEC sets new 10.9 terabit/second, ultra-dense WDM transmission capacity world record. http://www.nec.co.jp/press/en/0103/2201.html, March 2001. Press Release.

[127] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., 1988.

[128] H.-P. Nolting. All-optical 3R-regeneration for photonic networks (COST 266). In *Proc. of ONDM*, volume I, pages 129–131, Budapest, Hungary, 2003.

[129] Ch. Nomikos. Path coloring problems in graphs are non-approximable. In *Proc. of 7th Hellenic Conference on Informatics*, Ioannina, Greece, 1999.

[130] Ch. Nomikos, A. Pagourtzis, K. Potika, and S. Zachos. Fiber cost reduction and wavelength minimization in multifiber WDM networks. In *NETWORKING 2004*, volume 3042 of *Lecture Notes in Computer Science*, pages 150–161, 2004.

[131] Ch. Nomikos, A. Pagourtzis, and S. Zachos. Routing and path multicoloring. *Information Processing Letters*, 80:249–256, 2001.

[132] T.F. Noronha and C.C. Ribeiro. Routing and wavelength assignment by partition colouring. *European Journal of Operations Research*, 171:797–810, 2006.

[133] S. Orlowski and R. Wessäly. An integer programming model for multi-layer network design. ZIB-Report ZR-04-49, Zuse Institute Berlin (ZIB), 2004.

[134] A.E. Ozdaglar and D.P. Bertsekas. Optimal solution of integer multicommodity flow problems with application in optical networks. In *Proc. of Symposium on Global Optimization*, Santorini, Greece, 2003.

[135] A. Pal and U. Patel. Routing and wavelength assignment in wavelength division multiplexing networks. In *Proc. of 6th IWDC*, volume 3326 of *Lecture Notes in Computer Science*, pages 391–396. Springer-Verlag, 2004.

[136] C.M. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, Inc., 1994.

[137] P. Pardalos, T. Mavridou, and J. Xue. The graph coloring problem: A bibliographic survey. In *Handbook of Combinatorial Optimization*, volume 2, pages 331–395. Kluwer Academic Publishers, 1998.

[138] J.H. Park and A. Takada. Node configuration, QoS and network performance evaluation of an ultra-high speed optical packet switching ring network. In *Proc. of ONDM*, volume I, pages 663–678, Budapest, Hungary, 2003.

[139] S. De Patre, G. Maier, M. Martinelli, and A. Pattavina. Design of static WDM mesh networks with dedicated path-protection. In *Next Generation Optical Network Design and Modelling*, pages 281–294. Kluwer Academic Publishers, 2003.

[140] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufman Publishers, 2004.

[141] D. Poensgen. *Facets of Online Optimization – Online Dial-a-Ride Problems and Dynamic Configuration of All-Optical Networks*. PhD thesis, Technische Universität Berlin, 2003.

[142] C. Qiao and M. Yoo. Choices, features and issues in optical burst switching. *Optical Networks Magazine*, 1(2):36–44, 2000.

[143] P. Raghavan and E. Upfal. Efficient routing in all-optical networks. In *Proc. of 26th ACM Symp. on Theory of Computing*, pages 134–143, 1994.

[144] B. Ramamurthy, H. Feng, D. Datta, J.P. Heritage, and B. Mukherjee. Transparent vs. opaque vs. translucent wavelength-routed optical networks. In *Proc. of OFC*, pages 59–61, San Diego, USA, 1999.

[145] B. Ramamurthy and B. Mukherjee. Wavelength conversion in WDM networking. *IEEE Journal on Selected Areas in Communications*, 16(7):1061–1073, 1998.

[146] R. Ramaswami and G.H. Sasaki. Multiwavelength optical networks with limited wavelength conversion. In *Proc. of INFOCOM*, pages 489–498, 1997.

[147] R. Ramaswami and K.N. Sivarajan. *Optical Networks: A Practical Perspective*. Morgan Kaufmann Publishers, Inc., 1998.

[148] E.R. Scheinerman and D.H. Ullman. *Fractional Graph Theory.* Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., 1997.

[149] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

[150] A. Schrijver. *Combinatorial Optimization – Polyhedra and Efficiency.* Springer-Verlag, 2003.

[151] N. Sengezer and E. Karasan. A tabu search algorithm for sparse placement of wavelength converters in optical networks. In *ISCIS 2004*, number 3280 in Lecture Notes in Computer Science, pages 247–256. Springer-Verlag, 2004.

[152] SNDlib. A library of test instances for survivable fixed telecommunication network design. http://sndlib.zib.de.

[153] D. Stamatelakis. Theory and algorithms for preconfiguration of spare capacity in mesh restorable networks. Master's thesis, University of Alberta, 1997.

[154] T.E. Stern and K. Bala. *Multiwavelength Optical Networks: A Layered Approach.* Addison Wesley Longman, Inc., 1999.

[155] S. Subramaniam, M. Azizoglu, and A.K. Somani. All-optical networks with sparse wavelength conversion. *IEEE/ACM Transactions on Networking*, 4(4):544–557, 1996.

[156] J.W. Suurballe and R.E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14:325–336, 1984.

[157] T-Systems International GmbH. http://www.t-systems.de.

[158] Telekom Austria AG. http://www.telekom.at.

[159] ITU (International Telecommunication Union). Types and characteristics of SDH network protection architectures. ITU-T Recommendation G.841, 1995.

[160] ITU (International Telecommunication Union). Architecture of optical transport networks. ITU-T Recommendation G.872, 1999.

[161] S. van Hoesel. Optimization in telecommunication networks. *Statistica Neerlandica*, 59(2):180–205, 2004.

[162] W. Vanderbauwhede and D.A. Harle. Novel design for an asynchronous optical packet switch. In *Proc. of ONDM*, volume II, pages 737–754, Budapest, Hungary, 2003.

[163] V.G. Vizing. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz.*, 3:25–30, 1964. (Russian).

[164] H.L. Vu, Z. Rosenberg, and M. Zukerman. Performance evaluation of optical burst switching networks with limited wavelength conversion. In *Proc. of ONDM*, volume II, pages 1155–1169, Budapest, Hungary, 2003.

[165] J. Walraevens, S. Wittevrongel, and H. Bruneel. An analytic technique to evaluate the performance of optical packet switches. In *Proc. of ONDM*, volume II, pages 1171–1185, Budapest, Hungary, 2003.

[166] R. Wessäly. *DImensioning Survivable Capacitated NETworks*. PhD thesis, Technische Universität Berlin, 2000.

[167] R. Wessäly. SNDlib: Official launch of the data library for survivable network design. In *Proc. of DRCN*, page 203, Isle of Ischia, Naples, Italy, 2005.

[168] R. Wessäly, S. Orlowski, A.M.C.A. Koster, A. Zymolka, and C. Gruber. Demand-wise shared protection revisited: A new model for survivable network design. In *Proc. of INOC*, pages 100–105, Lisbon, Portugal, 2005.

[169] P. Winkler and L. Zhang. Wavelength assignment and generalized interval graph coloring. In *Proc. of 14th SODA*, pages 830–831, 2003.

[170] L.A. Wolsey. Valid inequalities for 0/1 knapsacks and MIPs with Generalized Upper Bound constraints. *Discrete Applied Mathematics*, 29:251–261, 1990.

[171] T.-H. Wu. *Fiber Network Service Survivability*. Telecommunications Library. Artech House, 1992.

[172] X. Yang, M. Dang, M. Zhang, and L. Li. An adaptive-threshold based multiclass optical burst assembly technique to support QoS in optical burst switching network. In *Proc. of ONDM*, volume II, pages 1089–1101, Budapest, Hungary, 2003.

[173] M. Yannakakis. Node- and edge-deletion $\mathcal{NP}$-complete problems. In *Proc. of 10th ACM Symp. on Theory of Computing*, pages 253–264, 1978.

[174] S. Yao, B. Mukherjee, and S. Dixit. Asynchronous optical packet-switched networks: A preliminary study of contention-resolution schemes. In *Optical Network Workshop*, Dallas, USA, 2000.

[175] D. Yuan. An annotated bibliography in communication network design and routing. In *Optimization Models and Methods for Communication Network Design and Routing*. Linköping Studies in Science and Technology, Dissertations, No. 682, 2001.

[176] D. Yuan. *Optimization Models and Methods for Communication Network Design and Routing*. PhD thesis, Linköpings Universiteit, 2001. Linköping Studies in Science and Technology, Dissertations, No. 682.

[177] H. Zang, J.P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical Networks Magazine*, 1(1):47–60, 2000.

[178] A. Zymolka, A.M.C.A. Koster, and R. Wessäly. Transparent optical network design with sparse wavelength conversion. In *Proc. of ONDM*, pages 61–80, Budapest, Hungary, 2003.

Index