Design of Multi-layer Telecommunication Networks: Fairness, Resilience, and Load Balancing

Kubilinskas, Eligijus

2008

Design of Multi-layer Telecommunication Networks

Fairness, Resilience, and Load Balancing

Eligijus Kubilinskas

LUND UNIVERSITY

Department of Electrical and Information Technology
Faculty of Engineering, LTH

ISSN 1654-790X No. 1

Printed in Sweden
Tryckeriet i E-huset
Lund 2008

To my girlfriend Rimgailė, my family and friends

This thesis is submitted to the Research Education Board of the Faculty of Engineering at Lund University, in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Engineering.

Contact information:

Eligijus Kubilinskas
Department of Electrical and Information Technology
Lund University
PO Box 118
SE-221 00 LUND
Sweden

Phone: +46 46 222 77 32
Fax: +46 46 14 58 23
e-mail: eligijus.kubilinskas@eit.lth.se

Abstract

Migration to Next Generation Internet architectures poses new challenges for network operators in planning core networks and calls for efficient network planning and optimization tools. Optimization models underlying such tools are developed in this thesis. We study a number of single and two-layer core network design problems defined as mathematical programmes, focusing on fair bandwidth allocation among demands, recovery mechanisms, and load balancing on network links.

Assuming elastic traffic, fair allocation of network bandwidth among the users is not trivial since different users may have different preferences and requirements for minimum bandwidth. We study single and two layer network dimensioning tasks where elastic and non-elastic demands are combined, and investigate different fairness principles, with special attention devoted to proportional fairness. The models are developed for designing the networks for the normal state of network operation, as well as for failure states.

For the two-layer problems it is not at all clear in which layer the recovery should be performed, and what recovery mechanisms to use. Therefore, recovery aspects in different layers are studied and models are provided for different recovery mechanisms. Furthermore, a generic resolution framework and heuristic algorithms for the selected dimensioning and allocation problems in two-layer networks are developed.

Balancing of load on network links decreases probability of rejection of future requests due to shortage of resources in some parts of the network. In the thesis different load balancing options are discussed, and an integrated routing, recovery, and load balancing strategy is developed. It combines failure dependent backup path protection, shortest path routing, and load balancing according to proportional fairness principle.

The thesis presents both theoretical findings, models, and resolution algorithms for the studied problems. Efficiency of the algorithms is illustrated by numerical examples. The thesis also gives a systematic view and classification of different aspects related to network architecture, recovery, fairness, and flow/congestion control.

Acknowledgments

I am very grateful to my advisor Professor Michał Pióro for introducing me into the dynamic and fascinating topic of network design, for guidance, valuable advices, long and thought-provoking discussions and inspiration. Special thanks to my colleague Pål Nilsson for critical listening to my ideas, brainstorming sessions, proof-readings and help in many other ways, and for making it so fun to work together. My gratitudes to all the former colleagues at the Department of Communication Systems. Many thanks to Professor Ulf Körner for having me as a Ph.D. student and Ingrid Nilsson for always helping me out with all the administrative things. I want to thank Jens A. Andersson for consultations about networking technology and for highlighting the practical networking aspects. Many thanks to Associate Professor Christian Nyberg for help with mathematical problems and organizational issues. I am also grateful to Associate Professor Ulf Ahlfors, Jianhua Cao, Piotr Chołda, Mateusz Dzida, Börje Jenssen and Michał Zagożdżon who helped me in different aspects of this thesis writing.

My gratitudes to Sandro Bosio, Associate Professor Antonio Capone, Giuliana Carello, Stefano Gualandi, Professor Francesco Maffioli, Professor Federico Malucelli, and others at Department of Information and Communication Technology, Technical University of Milan for their help in many ways during my visit there.

My special thanks to Professor Vidas Lauruška and others at Faculty of Technology, Šiauliai University, Lithuania, for helping me to make first steps as a researcher and for not forgetting me. Also, I want to thank all other people not mentioned here who contributed to this thesis in any way.

Last, but not least, I want to thank my girlfriend Rimgailė and my family for their love and for always being near, helping, encouraging and supporting me in the most difficult moments of this thesis writing. I am also grateful to all friends in Lithuania and the new friends I have met here in Lund and around the world for all the happy moments, adventures, and understanding.

The work in this thesis has been done within the following projects: 1) "Design of Modern Telecommunication Networks—Multi-commodity Flow Optimization Approach", granted by the Swedish Research Council (grant number 621-2003-2767) and 2) “Design and Engineering of the Next Generation Internet—towards convergent multi-service networks”, carried within Network of Excellence Euro-NGI, which is a part of the 6th Framework European Project. Their support is thankfully acknowledged.

Eligijus Kubilinskas
Lund, February 2008

Table of Contents

List of Acronyms

10 GbE	10 Gigabit Ethernet
AC	Admission Control
AIMD	Additive Increase Multiplicative Decrease
AQM	Active Queue Management
AS	Autonomous System
ASN	Autonomous System Number
ASON	Automatic Switched Optical Network
ASP	Available Shortest Path
ASTN	Automatic Switched Transport Network
ATM	Asynchronous Transfer Mode
B&B	Branch-and-Bound
B&C	Branch-and-Cut
BGP	Border Gateway Protocol
CAPEX	Capital Expenditure
CATV	Cable Television
CM	Cost Minimization
CR-LDP	Constrained Label Distribution Protocol
CSFQ	Core-Stateless Fair Queueing
CSPF	Constrained Shortest Path First
CXP	Convex Programming Problem
DiffServ	Differentiated Services
DPI	Deep Packet Inspection
DRR	Deficit Round Robin

DSL	Digital Subscriber Line
DSP	Demand-wise Shared Protection
DWA	Dynamic Wavelength Assignment
DWDM	Dense Wavelength Division Multiplexing
EGP	Exterior Gateway Protocol
E-NNI	External Network-Network Interface
FDBP	Failure-dependent Backup Path
FEC	Forward Error Correction
FIFO	First In First Out
FQ	Fair Queueing
FQS	Fair Queueing Scheduling
FR	Frame Relay
FRED	Flow Random Early Drop
FTTH	Fiber To The Home
GFP	Generalized Framing Procedure
GL	Generalized Label
GMPLS	Generalized Multi-protocol Label Switching
HP	High Priority
HS	Hot Stand-by
HTTP	Hyper Text Transfer Protocol
IGP	Interior Gateway Protocol
I-NNI	Internal Network-Network Interface
ION	Intelligent Optical Network
IP	Internet Protocol
IP	Integer Programming Problem
IQPR	Improved Quasi Path Restoration
ISIS	Intersystem to Intersystem
ISP	Internet Service Provider
ITN	Intelligent Transport Network
L1	Layer 1
L2	Layer 2
L2SC	Layer Two Switch Capable
LAN	Local Area Network
LMP	Link Management Protocol
LP	Linear Programming Problem
L-P	Link-Path
LR	Lagrangian Relaxation
LSC, λSC	Lambda Switch Capable

LSP	Label Switched Path
LSR	Label Switch Router
MAC	Medium Access
MC	Multi-criteria
MIP	Mixed Integer Programming Problem
MMF	Max-Min Fairness
MPLS	Multi-Protocol Label Switching
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
NDP	Network Design Problem
NGI	Next Generation Internet
OAM&P	Operation, Administration, Maintenance, and Provisioning
OM	Optimization Model
OPEX	Operational Expenditure
OQPR	Original Quasi-Path Restoration
OSI	Open Systems Interconnect
OSPF	Open Shortest Path First
OTN	Optical Transport Network
OXC	Optical Cross Connect
P2P	Peer-to-Peer
p-cycle	Pre-configured Protection Cycle
PD	Path Diversity
PF	Proportional Fairness
PG	Path Generation
PoS	Packet over SONET
PSC	Packet Switch Capable
QoS	Quality of Service
QPR	Quasi-Path Restoration
RB	Residual Bandwidth
RSVP	Resource Reservation Protocol
SAN	Storage Area Network
SBP	Single Backup Path
S-D	Source-Destination
SDH	Synchronous Digital Hierarchy
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network
SP	Shortest Path
SRLG	Shared Risk Link Group

SS7	Signaling System 7
TCP	Transmission Control Protocol
TE	Traffic Engineering
TM	Throughput Maximization
TMN	Telecommunication Management Network
UDP	User Datagram Protocol
UNI	User-Network Interface
VPN	Virtual Private Network
WAN	Wide Area Network
WFQ	Weighted Fair Queueing

Notation

Sets

$\mathcal{D}$	set of indices of demands
$\bar{\mathscr{P}}_d$	set of all possible paths in the upper layer for realizing demand d
$\widehat{\mathscr{P}_d}$	set of allowable paths in the upper layer for realizing demand d; $\widehat{\mathscr{P}_d} \subseteq \bar{\mathscr{P}}_d$
$\widehat{\mathcal{P}}_d$	set of indices of paths in $\widehat{\mathscr{P}_d}$
$\mathscr{P}_d$	set of candidate paths in the upper layer for realizing demand d; $\mathscr{P}_d \subseteq \widehat{\mathscr{P}_d}$
$\mathcal{P}_d$	set of indices of paths in $\mathscr{P}_d$
$\mathcal{E}$	set of indices of upper layer links
$\bar{\mathscr{K}}_e$	set of all possible paths in the lower layer realizing capacity of upper layer link e
$\widehat{\mathscr{K}_e}$	set of allowable paths in the lower layer realizing capacity of upper layer link e; $\widehat{\mathscr{K}_e} \subseteq \bar{\mathscr{K}}_e$
$\widehat{\mathcal{K}}_e$	set of indices of paths in $\widehat{\mathscr{K}_e}$
$\mathscr{K}_e$	set of candidate paths in the lower layer realizing capacity of upper layer link e; $\mathscr{K}_e \subseteq \widehat{\mathscr{K}_e}$
$\mathcal{K}_e$	set of indices of paths in $\mathscr{K}_e$
$\mathcal{G}$	set of indices of lower layer links
$\mathcal{S}$	set of indices of failure states

Indices

$d = 1, 2, \ldots, D$	or $d \in \mathcal{D}$	demands (pairs of nodes)
$p = 1, 2, \ldots, m(d)$	or $p \in \mathcal{P}_d$	candidate paths for flows realizing demand d
$e = 1, 2, \ldots, E$	or $e \in \mathcal{E}$	upper layer links
$k = 1, 2, \ldots, n(e)$	or $k \in \mathcal{K}_e$	candidate paths realizing capacity of link e
$g = 1, 2, \ldots, G$	or $g \in \mathcal{G}$	lower layer links
$s = 1, 2, \ldots, S$	or $s \in \mathcal{S}$	failure states
$i = 1, 2, \ldots, I$	or $i \in \mathcal{I}$	approximation pieces
$l = 0, 1$	or $l \in \{0, 1\}$	index 0 denotes nominal and 1 denotes protection path

Constants

D	number of demands
$m(d)$	number of upper layer paths realizing demand d
E	number of upper layer links
$n(e)$	number of lower layer paths realizing capacity of upper layer link e
G	number of lower layer links
S	number of states
δ_{edp}	$= 1$ if link e belongs to the path p realizing demand d, 0 otherwise
w_d	revenue coefficient for demand d
h_d	lower bound for total flow realizing demand d
H_d	upper bound for total flow realizing demand d
w_{ds}	revenue from demand d in state s
h_{ds}	lower bound for total flow realizing demand d in state s
H_{ds}	upper bound for total flow realizing demand d in state s
n_d	flow splitting factor for demand d
c_e	given capacity of link e
ξ_e	unit cost of link e
α_{es}	availability coefficient of link e in state s , $0 \leq \alpha_{es} \leq 1$
θ_{dps}	availability coefficient for path p of demand d in state s; $\theta_{dps} = \min\{\alpha_{es} : \delta_{edp} = 1\}$
φ_{gek}	$= 1$ if link g belongs to the path k realizing capacity of link e, 0 otherwise
α_{gs}	availability coefficient of link g in state s , $0 \leq \alpha_{gs} \leq 1$
θ_{eks}	availability coefficient for path k realizing capacity of link e in state s; $\theta_{eks} = \min\{\alpha_{gs} : \varphi_{gek} = 1\}$
n_e	flow splitting factor for link e
L	module size of the demand for the upper layer
M	module size of the demand for the lower layer

N	module size for the lower layer link capacities (=number of wavelengths per fiber)
a_i	approximation coefficient
b_i	approximation coefficient
B	assumed budget

Variables

x_{dp}	flow allocated to path p of demand d
x_{dps}	flow allocated to path p of demand d in state s
X_d	total flow allocated to demand d
X_{ds}	total flow allocated to demand d in state s
f_d	total approximated flow of demand d
f_{ds}	total approximated flow of demand d in state s
v_{dps}	$= 1$ if path p is used to realize all the flow of demand d in state s; 0 otherwise
v_{dpl}	$= 1$ if path p is used to carry all the flow of demand d in nominal state (for $l = 0$) or failure states (for $l = 1$); 0 otherwise
y_e	capacity of link e
$\bar{y}_e$	load of link e
y_{es}	capacity of link e in state s
$\bar{y}_{es}$	load of link e in state s
z_{ek}	flow allocated to path k realizing capacity of link e
z_{eks}	flow allocated to path k realizing capacity of link e in state s
u_g	capacity of link g
u_{gs}	capacity of link g in state s
t	auxiliary variable
R	revenue
R_s	revenue in state s
T	total network throughput
C	total network cost

CHAPTER 1

Introduction

1

1.1 Background

The Internet experiences a constant tremendous growth in the amount of network users, new services and applications that use the network, the resulting traffic, and the installed bandwidth. In effect, the Internet Protocol (IP) packet traffic constitutes the majority of traffic carried by backbone transport networks and by far exceeds the traditional voice traffic. Rapidly growing peer-to-peer (P2P) overlay networks and services create enormous loads on the internet service providers' networks without generating satisfactory revenue, still forcing the network providers to constantly expand the network capacity to keep customers' requirements and expectations satisfied. Besides, many services are migrating to IP. It has became vital for network providers to offer the so called triple-play services, i.e., data, voice and video. The data traffic streams generated by the packet-based IP data, Ethernet frames, cell-based Asynchronous Transfer Mode (ATM) data and Frame Relay (FR) data, circuit-based voice and fax data, as well as delay sensitive streaming video and voice over IP data are multiplexed and carried over the same multi-service backbone networks. The need of carrying multiple services in one network has lead to a complex multi-layer architectures of network resources (and protocols). In a multi-layer network each resource layer basically corresponds to a different data transfer technology. Such architectures are not only inefficient in terms of resource utilization (because of packet overheads required by each layer), but also imply high maintenance and operation complexity (and costs) for network operators. Constantly growing demand for bandwidth and necessity to account for traffic peaks have been usually addressed by over-dimensioning of network resources what is economically questionable in today's market.

In order to stay competitive, operators are forced to increase efficiency and flexibility of their networks. This can be done in several ways. First of all, the trend is to simplify the existing network architecture, converging towards a two layer, IP/MPLS over DWDM, architecture, where IP takes role of the service integration traffic layer, multi-protocol label switching (MPLS) provides traffic engineering possibilities and dense wavelength division multiplexing (DWDM) is a transport layer of high capacity. The two layer architecture is foreseen to form the basis of next generation Internet (NGI). What is important, such type of new architectures also open new possibilities for deploying resource-efficient recovery mechanisms. Finally, we observe that constantly growing amount of traffic generated by bandwidth-greedy services (such as those based on P2P) makes it necessary to assign available network resources among the users of best-effort network services in a fair way in order to increase users' satisfaction with network services.

New network architectures, growing and changing traffic, new service paradigms, requirements for fair resource sharing, and resilience pose a challenge for network planning and managing process. This raises a need for efficient network design tools based on optimization algorithms that allow to take all these aspects into account both for the greenfield network design and for the routine periodical network redesign actions in operating networks. Development of network design/optimization models underlying such network design tools are at the core of this thesis. In the thesis we state many relevant single and multi-layer network design problems as optimization models and develop efficient algorithms (both exact and heuristic) for their resolution. Different recovery options, fairness principles, and load balancing approaches are considered in this thesis. The developed models and algorithms can be used in network planing and decision support systems.

1.2 Thesis outline

The thesis is organized as follows. We start the presentation in Chapter 2 by giving a brief introduction to backbone telecommunication networks from technological perspective. We discuss network architectures and explain why and how different technologies are combined in backbone networks leading to the architectures composed of several layers of resources. This brings us to presentation of reasons and market drivers for the evolution of the multi-layer network architectures towards next generation Internet. We present different standardization activities and proposed models for NGI. Network recovery is undeniably important part of network functionality, and recovery mechanisms are closely related to network architectures. Therefore we first discuss common causes for network failures, and then review and classify various recovery mechanisms and their interoperability strategies for multi-layer networks.

In Chapter 3 we introduce a modeling framework. Network design problems studied in the thesis are presented in a form of multi-commodity flow optimization models. In many cases the models are multi-criteria optimization problems. Therefore we start this chapter by introducing a general optimization model and discuss different entities of it, paying special attention to multi-criteria goal functions. With the help of the general model we roughly classify optimization problems and give an overview of main resolutions methods/algorithms. We then turn our attention to modeling of network entities and functions and introduce the reader to both single and two layer network modeling using the introduced mathematical framework.

In Chapter 4 we give an introduction to the theory of fair resource sharing in data networks. Fairness is in one of the major aspects prevailing in most of the optimization models considered in the thesis. Thus the chapter covers different aspects of fairness. We start by describing elastic traffic—the type of traffic which constitutes majority of the traffic in the Internet. We further reason why do available network resources have to be shared among the competing users in a fair way by discussing growth and changing nature of traffic, as well as capacity expansion in networks and market forces. A question of fair resource sharing brings us to the topic of utility and welfare functions which we briefly describe thus creating a framework for classifying different fairness principles. We then cover two major types of fairness, namely max-min and proportional fairness, encountered in the optimization models presented in the thesis. The discussion leads us to definition of lexicographical maximization, as a means of achieving max-min fairness. We then compare bandwidth efficiency of the fairness principles. Finally we give a detailed overview of flow/congestion control mechanisms from the fairness perspective and provide classification of mechanisms for fair resource sharing in telecommunication networks.

In Chapter 5 we study fair bandwidth sharing in single layer networks. The network design problems studied in this chapter are combined allocation and dimensioning problems with an emphasis on proportionally fair flow allocation among demands. We first present a model assuming only normal network state and provide a detailed derivation of the path generation algorithm for the problem resolution. The discussion is intended also to help understanding path generation algorithms for other problems discussed in the thesis. Then we extend the model to take possible failures into account. The latter optimization model is of multi-criteria optimization type and combines two types of fairness—among demands and among failure states. We assume proportional fair bandwidth sharing among demands and investigate various options for sharing revenue among failure states. We also discuss different extensions to the problems. We then present two algorithms for resolving the problem with max-min fair sharing among failure states, as well as develop a path generation algorithm. The numerical examples illustrate the efficiency of algorithms.

In Chapter 6 we extend the optimization models presented in Chapter 5 to two-layer networks. Four network optimization models are presented in this chapter: one for the normal network state and three for resilient network design. The three problems differ by where the recovery is performed: in the lower layer only, in the upper layer only, and in both layers in a coordinated way. All the problems concern capacity dimensioning and flow allocation in both network layers, while fair allocation of bandwidth among demands and failure states is the main objective. We assume proportionally fair allocation of bandwidth among demands and max-min fair allocation of revenue among failure states. In the chapter we mainly study the bandwidth efficiency of performing recovery in different layers. We develop algorithms for resolving the considered problems by extending the algorithms presented in Chapter 5.

In Chapter 7 we develop path generation algorithms for all the problems developed in Chapter 6 and provide some numerical examples illustrating the efficiency of the algorithms.

In Chapter 8 we develop a framework and the heuristic algorithm for iterative resolution of two-layer problems. The framework allows for efficient resolution of diverse two-layer (can be extended to more layers) network design problems, employing different recovery mechanisms and objectives. We illustrate this as follows. First we present models for several most often used recovery mechanisms in traffic and transport layers. We then give three examples of two-layer problems, employing different recovery mechanisms and objectives, and develop the heuristic algorithm for resolving the problems. This way also the general resolution framework is explained by example.

In chapter 9 we present an integrated recovery, routing, and load balancing strategy for an upper (IP/MPLS) layer of a two-layer network. It applies for the case when demands have associated priority levels, and we want to optimize network resource allocation in order to serve more incoming requests and to maintain as much as possible LSPs when a failure occurs. The proposed recovery strategy is failure-dependent backup path protection with shortest path routing of protection paths. The load balancing part is based on the proportionally fair allocation of residual bandwidths. All the three aspects are combined into a multi-criteria objective. The efficiency of the proposed strategy is compared against single backup path routing and various on-line recovery schemes.

In Chapter 10 we summarize contributions of the thesis, provide general conclusions of the work and identify directions for future work.

1.3 List of publications

The thesis is mainly but not exclusively based on the following papers:

1. **Resilient Dimensioning of Proportionally Fair Networks**
   Michał Pióro, Eligijus Kubilinskas and Pål Nilsson
   European Transactions on Telecommunications, Vol. 16, No. 3, John Wiley & Sons, UK, May-June 2005.
   (First appeared in Proc. of *2nd Polish-German Teletraffic Symposium, PGTS 2002*, Gdansk, Poland, September 2002.)

2. **Design Models for Robust Multi-Layer Next Generation Internet Core Networks Carrying Elastic Traffic**
   Eligijus Kubilinskas, Pål Nilsson and Michał Pióro
   Journal of Network and Systems Management, Vol. 13, No. 1, Kluwer, USA, March 2005.
   (First appeared in Proc. of *Fourth International Workshop on Design of Reliable Communication Networks, DRCN 2003*, Banff, Canada, October 2003.)

3. **Notes On Application of Path Generation to Multi-Layer Network Design Problems with PF Flow Allocation**
   Eligijus Kubilinskas
   Technical report CODEN: LUTEDX(TETS-7200)/1-24/(2004)&local 7, Dept. of Communication Systems, Lund University, Lund, Sweden, 2004.

4. **On Application of Path Generation to Multi-Layer Network Design Problems with Proportionally Fair Flow Allocation**
   Eligijus Kubilinskas
   In Proc. of *Seventeenth Nordic Teletraffic Seminar, NTS 17*, Fornebu, Norway, August 2004.

5. **An IP/MPLS over WDM Network Design Problem**
   Eligijus Kubilinskas and Michał Pióro
   In Proc. of *International Network Optimization Conference, INOC'05*, Lisbon, Portugal, March 2005.

6. **Two Design Problems for the IP/MPLS over WDM Networks**
   Eligijus Kubilinskas and Michał Pióro
   In Proc. of *5th International Workshop on Design of Reliable Communication Networks, DRCN2005*, Island of Ischia (Naples), Italy, October 2005.

7. **Network Protection by Single and Failure-Dependent Backup Paths - Modeling and Efficiency Comparison**
   Eligijus Kubilinskas, Faisal Aslam, Mateusz Dzida and Michał Pióro
   System Science, Vol. 32, No. 3, Poland, 2006.
   (First appeared in Proc. of *4th Polish-German Teletraffic Symposium, PGTS 2006*, Wrocław, Poland, September 2006.)

8. **Recovery, Routing and Load Balancing Strategy for an IP/MPLS Network**
   Eligijus Kubilinskas, Faisal Aslam, Mateusz Dzida and Michał Pióro
   Lecture Notes in Computer Science, 2007(4516), Springer,
   Proc. of *Managing Traffic Performance in Converged Networks, 20th International Teletraffic Congress, ITC20 2007*, Ottawa, Canada, June 2007.

9. **An Iterative Framework for Resolving Two-Layer Telecommunication Network Design Problems**
   Eligijus Kubilinskas and Michał Pióro
   Technical report CODEN:LUTEDX(TETS-7222)/1-12/(2008)&local 2,
   Dept. of Electrical and Information Technology, Lund University, Lund, Sweden, February 2008.

10. **Load Balancing and Network Protection by Failure-dependent Backup Paths**
    Eligijus Kubilinskas, Faisal Aslam, Mateusz Dzida and Michał Pióro
    Technical report CODEN:LUTEDX(TETS-7223)/1-22/(2008)&local 3,
    Dept. of Electrical and Information Technology, Lund University, Lund, Sweden, February 2008.

CHAPTER 2

Network architectures and resilience

2

2.1 Local and wide area telecommunication networks

Packet switched networks can be roughly divided into two broad categories: local area networks (LANs) and wide area networks (WANs). Local area networks span short distances and are intended for interconnection of nodes in a single room, office or campus. Wide area networks, on the other hand, span large distances and employ technology allowing for information exchange at high transfer rates. Metropolitan area networks, nation-wide networks or even international networks are examples of WANs , which are also sometimes called *long haul networks* due to the large covered distances. The most popular technology for LANs is Ethernet, while WAN network architecture is quite complicated and is discussed in details below. WANs interconnect smaller networks (and thus are called *backbone networks*) and connect with other WANs to assure global reachability.

2.2 Backbone telecommunication networks

The scope of this thesis is backbone networks. Formally, each such large network is defined as an autonomous system. Interconnection of all autonomous systems forms the Internet backbone. We will discuss autonomous systems and the Internet backbone in more detail.

2.2.1 An autonomous system

An autonomous system (AS) is a set of IP networks and routers (see Figure 2.1) under the administrative control of one or more parties that present to the Internet a common routing policy [1; 2]. A unique AS number (ASN) is allocated to each AS and is used in a Border Gateway Protocol (BGP) routing. Each AS is free to choose any internal routing architecture and protocols. For routing between

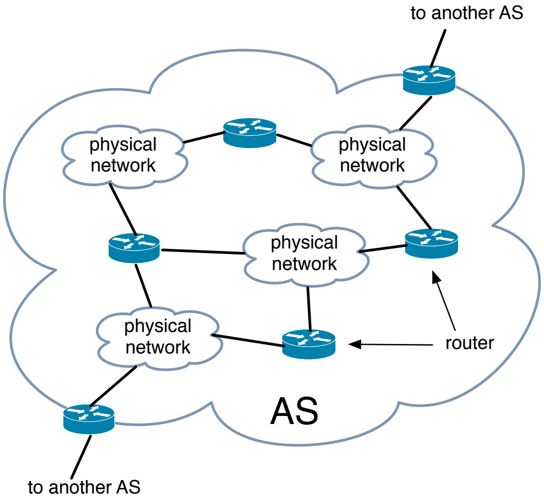


Figure 2.1: An autonomous system composed of several networks and routers.

ASs the BGP protocol is used. Depending on the connections and operation, there are three types of ASs defined: *multi-homed, stub* and *transit*. A multi-homed AS is an AS that connects to at least two ASs. The multiple connection gives an advantage that the AS remains connected to the Internet even if one of the connections fail. However, multi-homed ASs typically do not allow transit traffic (which is on the way from one AS to another AS) to pass through it. Stub AS is such that connects to only one other AS. Autonomous systems that allow connections through itself (transit) to other ASs are called transit ASs. Internet service providers (ISPs) are always transit ASs [2].

A network of an ISP usually consists of two functional parts: a core (backbone) network and an access network, as illustrated in Figure 2.2. Access networks are used to connect clients to provider's network and, compared to a core network, offer rather low transfer rate connections. However, transfer rates are constantly increasing in access networks as well, as new high-speed technologies, such as fiber to the home (FTTH), are introduced. Traffic collected from access networks is transfered by a high speed core network, which also connects provider's network to other ASs.

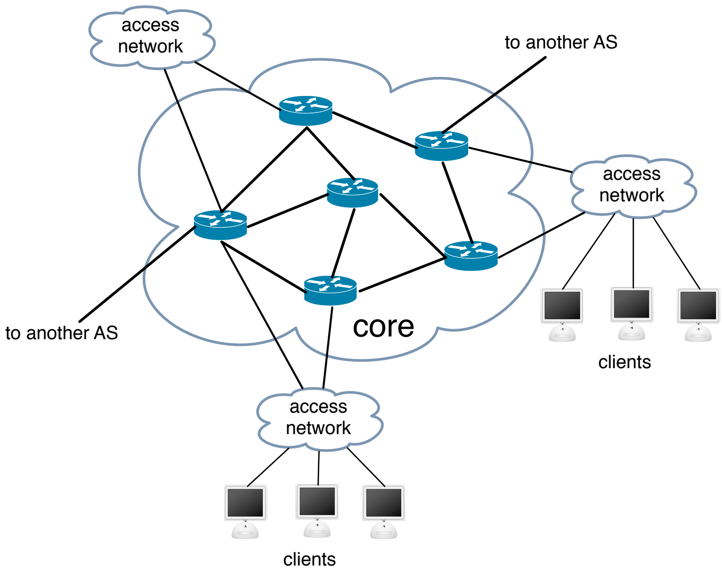


Figure 2.2: A typical network of an ISP consisting of core and access networks.

2.2.2 The Internet backbone

The Internet backbone consists of many interconnected ASs, that is, commercial, academic, governmental and other high throughput large-scale networks. Typically, networks of the largest ISPs, carrier network providers (that have smaller ISPs and corporate networks as their clients), as well as corporate and governmental networks interconnect either at Internet exchanges (IXP or IX), or by private peering links. At IXPs the networks are usually interconnected using open systems interconnection (OSI) layer 2 access technology. Nowadays, 1/10 Gbps Ethernet switches are used for the interconnection purposes, where each network gets connected via a single port. The largest of the IXP's both in terms connected peers and throughput is the Amsterdam Internet Exchange [2]. An alternative way for given two providers to interconnect their networks is by private peering links, where the dedicated capacity is not shared by other parties. Historically such an interconnection was done using synchronous optical network/synchronous digital hierarchy (SONET/SDH) circuits between the facilities of the providers. Today the interconnection usually occurs at the so called carrier hotels or carrier-neutral colocation facilities. Physically the interconnection is of point-to-point type using the suitable technology, or even a dark fiber. Most traffic exchange in the Internet occurs via private peering. Routing information between ASs is exchanged using BGP.

Interconnection of two networks is accompanied by an interconnection agreement, which specifies on which terms and conditions do the networks exchange traffic. From perspective of traffic exchange the relationship between two interconnected networks can be denoted as *transit, peer* or *customer*. Transit (or *pay*) relationship implies that one network provider (let us call it NET1) pays settlement to other provider (NET2) for Internet access. Peer (also called *swap*) relationship implies that the two networks (NET1 and NET2) exchange traffic freely. Customer (or *sell*) relationship means that a network provider (NET2) is getting payed to provide Internet access to another network (NET1). Thus, in order for one network to be able to reach another network on the Internet, it must either peer with the network directly, or sell Internet connection to that network, or pay some other network for transit to the desired network, where the transited network must also be in peer, sell or pay relationship with the next network on the way [2].

2.3 Architectures of backbone telecommunication networks

The resources of today's backbone telecommunication networks are structured in a multi-layered fashion which is referred to as the *multi-layer network architecture*. As opposed to the protocol architecture (e.g., specified by the OSI model), the architecture of resources relates to different technologies (or instances of the same technology) that are deployed in the network. The networks are usually composed of several resource layers, corresponding to different technologies that are stacked one upon another in order to achieve the desired overall network functionality. An example of a commonly used architecture is IP over ATM over SDH over DWDM, as shown in Figure 2.3(a). The reason for using multi-layer architectures is that no single technology is able to provide the functionality required by diverse services, reliability, high throughput, etc. Due to the popularity of the Internet and widespread of the Ethernet transport, as the main standard for LANs, IP has become the main platform for introducing and providing different services. Nowadays, IP traffic constitutes the majority of traffic carried in the networks. Unfortunately, native IP does not have any traffic engineering capabilities, QoS, nor reliability-assuring mechanisms. This is where ATM comes into picture. ATM is able to provide quality of service (QoS), reliability, flow control, has monitoring and traffic engineering capabilities. But on the other hand, ATM is not widely used by end-users and have a big overhead due to small cell sizes. Therefore running IP over ATM complements IP with the features it lacks. SONET/SDH is used as a transport layer to carry traffic over fiber, because of its low delay, low error rate, inbuilt protection switching, and functionalities for management and monitoring. Finally, DWDM is used to effectively increase and share capacity of fibers [3]. To conclude: only a combination of multiple layers constitute a network architecture suitable for multi-service reliable communication backbone networks.

Unfortunately, the resource utilization in a multi-layer network is inefficient. The inefficiency is caused by an overhead added when multiplexing traffic of an upper layer into flows of a lower layer, as well as by some framing issues. Thus, for two adjacent resource layers, multiplexing upper layer traffic flows into the lower layer flows results in about 80% utilization of the lower layer link capacities [3]. And for a network with several resource layers these utilization factors multiply to yield the total utilization factor for the whole network, which can even be as low as 50%. Additionally, due to protection switching techniques in SONET/SDH, often only 50% of capacity installed in SONET/SDH layer is used to carry the payload. Therefore the utilization factor is even smaller [3]. Poor utilization of resources is not the only problem of the multi-layer networks. Since every layer acts as a stand-alone entity and has its own management and control plane, the overall management and maintenance of the network becomes very complex and results in high network cost. Besides, introduction of new services is not easy since, in some cases, a service has to be provisioned in all the layers, using, possibly, different management systems for each layer.

Large traffic volumes make this inefficiency (both, in terms of capacity and management) not acceptable, and call for new and more efficient architectures. Evolution of the network architectures is shown in Figure 2.3(b)–2.3(c). The old architecture (Figure 2.3(a)) still prevails in many backbone networks today, while newly established networks often use packet over SONET (PoS) architecture, shown in Figure 2.3(b). Future networks are seen as IP/MPLS over DWDM. Functions of ATM are being replaced by generalized MPLS (GMPLS). Also, many functions of SONET are being delegated to DWDM. Still, a thin layer between IP/MPLS and DWDM will remain. It will be used to convert the upper layer traffic into bit strings for the physical transmission, perform flow control, framing, error monitoring, etc. Several technologies, such as Thin SONET, Generalized Framing Procedure (GFP), 10 Gigabit Ethernet (10 GbE) or digital wrappers, could be used in this layer. Transport network topology will also change. SONET rings will be replaced by mesh-interconnected Optical Cross

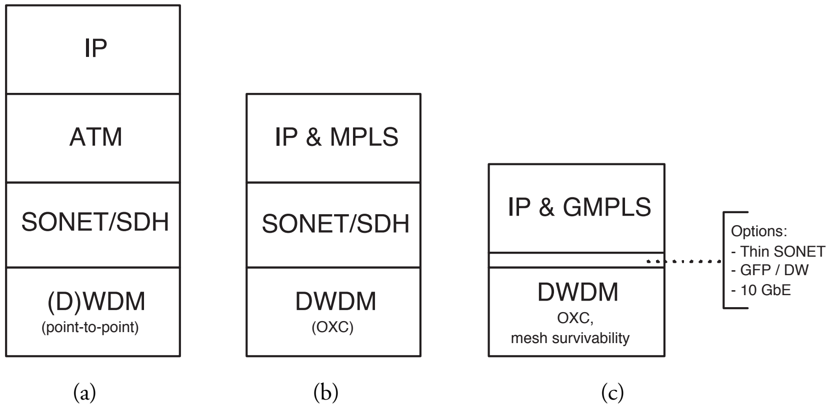


Figure 2.3: Evolution of the backbone network architecture (adapted from [3]).

Connects (OXCs), allowing for implementation of more effective recovery mechanisms.

In the sequel, when discussing different two-layer network models, the upper layer will be referred to as the *traffic layer*, and the lower layer—as *transport layer*.

2.3.1 The IP-over-optics network architecture

IP, due to its current popularity and a number of services using it, has secured its place as a service integration layer in the next generation architecture of the network. The choice of DWDM as the transmission technique for the transport layer network is also natural due to its ability of providing high throughput. However, DWDM does lack many features (such as framing, protection/restoration mechanisms, OAM&P[1] possibilities) provided by SONET/SDH in today's networks, to be able to act as a stand-alone transport layer. ITU-T addressed this issue by defining the Optical Transport Network (OTN) architecture [4], combining the advantages of SONET/SDH and DWDM. Basically, OTN uses DWDM as a physical transmission technology, complementing it by framing (wrapping) standards, as well as OAM&P and survivability functions. OTN also introduces forward error correction (FEC), which was absent in SONET/SDH, this way decreasing a number of necessary regenerators in the network and, thus, reducing the cost. Nodes in the OTN network are connected by fibers, and optical channels (*lightpaths*) are established between any pair of the nodes in the network. OTN performs multiplexing, routing, management and assures survivability of

[1] OAM&P stands for operation, administration, maintenance, & provisioning

the lightpaths. A distinct feature of OTN is its ability to provide a transport functionality for any digital signal. One feature that is missing in the OTN model is the ability to provision lightpaths automatically (i.e., without intrusion of management system) on demand from the client layer. This and other extensions to OTN, as well as abstract description of the OTN control plane functions and interfaces (within the OTN layer, as well as to the client layer) were defined by ITU-T as an Automatic Switched Optical Network (ASON) [5] model. The ASON model has been generalized for different transport technologies (not only for OTN) and defined as Automatic Switched Transport Network (ASTN) [6]. ASON/ASTN framework targets the so called *overlay model* of the control plane architecture, but defines no actual protocols to implement the defined functionality. In parallel to ITU-T, IEEE has developed a model of the multi-layer network and its control plane protocols within the Generalized Multi-protocol Label Switching framework [7]. GMPLS targets the so called *peer model* of the control plane architecture. However, protocols of GMPLS suite are also seen as the main candidates for implementing the functionality of ASON/ASTN framework. A detailed comparison of ASON and GMPLS can be found in [8]. The basic features and functions of different control plane models are presented below.

2.3.2 Network planes

Functionally, communication network can be seen as composed of three planes [9]:

1. *Data* or *user* plane. The transfer of payload (user data) takes place in this plane. In a multilayer network, every network layer has its own data plane.

2. *Control* plane manages signaling in the network required to assure functionality of the network data plane. Control plane performs connection setup, supervision and tear down, assures routing consistency, etc. In other words, control plane elements and protocols regulate how routers/switches process packets/connections. Certain routing protocol functions are also considered as a part of control plane. These are the functions responsible for building and maintaining the routing information in network nodes. Besides, control plane can define a preferential treatment for some packets/connections for which some quality of service is required. In a multilayer network every network layer has its own control plane, which typically functions in a distributed way in the network. An example of control plane is a Signaling System 7 (SS7) used for telephone networks.

3. *Management* plane is usually operating in a centralized way and typically performs monitoring of network devices and facilitates network administration. Typical usage examples of management plane functions are monitoring of network device up-times, measuring network interface throughput, collecting interface information, etc. [2]. In a multi-layer network the management plane is usually constituted by two parts: a *layer* management for each network layer and a *plane* management to assure operation between the layers. Typical examples of management plane systems are Telecommunication Management Network (TMN) [10] and Simple Network Management Protocol (SNMP) [11].

2.3.3 Control plane models for multi-layer networks

We have discussed in Section 2.3 that legacy telecommunication networks are usually composed of several layers of resources. Often each of the layers has its own control plane, making impossible to perform integrated management of network resources across the resource layers. This not only decreases the efficiency of resource utilization, increases management and operation costs, but also prevents from flexibly adapting to changing traffic patterns and capacity needs. Due to a range of new network services, both traffic patterns for given connections and locations between which traffic has to be carried changes over time. Thus there is a need for bandwidth-on-demand services which the legacy telecommunication networks can not provide. An example of an application which requires bandwidth-on-demand is a storage area network (SAN). The need for bandwidth management flexibility calls for intelligence in transport networks, leading to the concept of intelligent optical network (ION) [9] or, in general, intelligent transport network (ITN). Intelligent network control allows not only for efficient resource usage and reallocation on demand, but also makes possible to implement bandwidth-efficient recovery mechanisms. In the intelligent network it will be possible to set up (cross-layer) recovery connections after a failure has occurred, utilizing the surviving resources on all layers in the best way. Intelligence in the networks is implied by a control plane. There are three different control plane models envisaged for multi-layer networks: *overlay, peer* and *augmented*.

2.3.3.1 The overlay model and the ASON-based network architecture

In the overlay (also called *client-server*) model a network is seen as composed of two independent layers of resources: IP and ASON/ASTN (Figure 2.4). Both layers have their own independent transport, control and management planes.

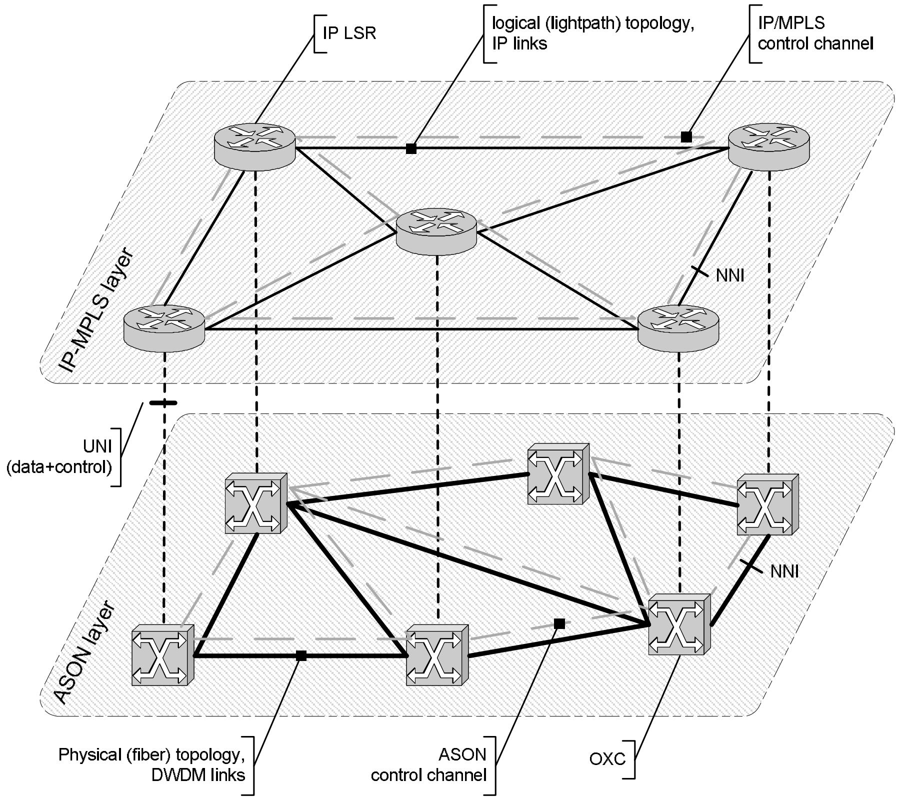


Figure 2.4: Network architecture with overlay model.

The transport plane performs switching (forwarding). The control plane performs signaling, routing, manages connections and implements a resilience mechanism. The management plane supervises the network, coordinates interworking between the layers and provides interface to an OAM&P system.

ASON/ASTN just defines a generic model of the intelligent transport network and functions of different entities, as well as interfaces between the entities. The ASON/ASTN based architecture implies the overlay network model. Since no protocols are developed by ITU-T to implement the intelligence of ASON/ASTN, ITU-T is closely collaborating with IEEE in order to adapt some GMPLS protocols for this purpose. Thus the control planes for both layers of the network can be instantiated from the same type of control plane, i.e., GMPLS.

Nodes of the upper (IP/MPLS) resource layer are Label Switch Routers (LSR). LSRs are interconnected by IP links. Besides the ordinary IP routing capabilities, LSRs are also Packet Switch Capable (PSC) or Layer Two Switch Capable (L2SC).

The PSC feature implies that routers are able to forward packets along Label Switched Paths (LSPs), based on packet formats and labels. L2SC routers are able to recognize and forward cells and frames (of ATM and FR) along LSPs. Signaling protocols extended with *traffic engineering* (TE) capabilities are to be used for configuration and control of LSPs. The two signaling protocols are the Resource Reservation Protocol (RSVP-TE) [12] and the Constrained Label Distribution Protocol (CR-LDP) [13]. Two routing protocols, extended with TE capabilities that are to be used on IP/MPLS layer are Open Shortest Path First (OSPF-TE) [14] and Intersystem to Intersystem (ISIS-TE) [15]. The IP/MPLS layer can be subdivided into routing domains. Interior Gateway Protocol (IGP) is used to distribute routing (link state) information within each domain, whereas Exterior Gateway Protocol (EGP) is used to exchange address reachability information between the domains.

The lower (ASON/ASTN) layer nodes are DWDM-enabled optical cross connects that are connected with fiber cables. Links between the nodes are formed by a number of wavelength channels. Paths in the DWDM layer are called *optical connections (lightpaths)* and are formed of a sequence of links. Lightpaths realize the capacities of the client layer links. OXCs are capable of performing wavelength conversion and switching of lightpaths depending on the incoming interface and wavelength. Lightpaths can be of one of the three types: i) *permanent* – set up by the management system; ii) *soft-permanent* – set up by the management system using control plane signaling protocols; iii) *switched* – set up directly by control plane signaling protocols on request from the client layer. The lower layer can also be subdivided into the carrier domains. Within each carrier domain of the lower layer link state information is distributed using Internal Network-Network Interface (I-NNI). Between the carrier domains address reachability is exchanged using External Network-Network Interface (E-NNI).

The layers interact according to the ASON/ASTN model. The IP layer acts like a client layer to the optical (i.e., DWDM) layer, which is a server layer. Each link of the IP layer is implemented by a lightpath in the DWDM layer. Communication between the layers uses a User-Network Interface (UNI). UNI is regarded as unsafe and therefore hides all the details (such as addressing, topology, routing) of the lower resource layer from the upper resource layer.

2.3.3.2 The peer model and GMPLS-based network architecture

All resource layers in a network with the peer model (Figure 2.5) are integrated into a single (unified) transport plane. Therefore the network also has single instances of control and management planes, which are common for the whole

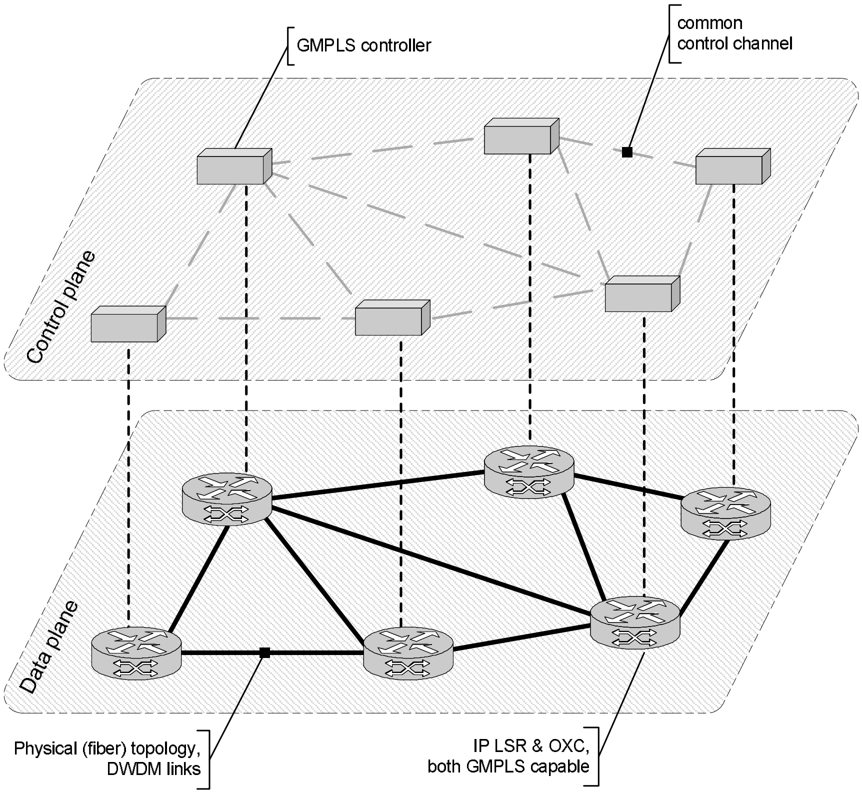


Figure 2.5: Network architecture with peer model.

network. This means that naming, addressing, topology and routing information is common for all types of resources in the network. For integrating all the resource layers it is necessary that there is a common protocol for conveying control information (such as address resolution and initiation of connection setup/release) between different resource layers. That is, the control protocol should allow IP and DWDM networks interact like peers. The peer model is targeted by the GMPLS concept. The GMPLS signaling, routing and link management protocols can be used to implement a common control plane in such a network.

The transport plane of the network is formed by nodes interconnected by fibers. The nodes in the network are GMPLS LSRs. They are able to perform ordinary IP routing, packet-oriented switching (have PSC, L2SC interfaces) and also lambda-oriented switching (have a Lambda Switch Capable (LSC or λSC) interface). The switching decision depends on the information associated with the *generalized label* (GL), which can consist in labels added to packets, cells, frames, as well as wavelength (and interface) of incoming signal or even an optical fiber. GL defines LSP encoding type (packet/digital wrapper/wavelength), switching type (switching of LSP as a TDM circuit or as a whole wavelength), and a generalized payload identifier (for example, packet over SONET) [9]. A number of wavelengths between adjacent nodes form links. And lightpaths are formed as sequences of links (a specific wavelength is used on each of the links for the specific lightpath). The network can be subdivided into domains. Link, packet LSP, and lambda LSP information is distributed within each domain using the IGP protocol, while address reachability is exchanged between the domains using the EGP protocols.

The control plane manages and supervises the transport plane. Information about network topology and status of resources is disseminated using routing protocols, i.e., OSPF-TE, IS-IS-TE, CSP and updated BGP. Connection setup and maintenance are performed by signaling protocols, such as RSVP-TE and CR-LDP. To control data link parameters the Link Management Protocol (LMP) is used. GMPLS extensions to RSVP-TE and CR-LDP signaling protocols are defined in [16; 17] and [18] respectively. These changes are implied by the introduction of the generalized label, and allow nodes to distribute GLs and perform configuration of nodes with different switching capabilities along an LSP.

2.3.3.3 Augmented control plane architecture

The augmented architecture constitutes a trade-of between the overlay and peer control plane models. Basically, the augmented model is similar to overlay model, and assumes that each resource layer has its own transport and control plane. However, some control information, such as reachability, is exchanged between the control plane instances of different layers. The augmented model allows to share the client-layer reachability information by distributing it over the transport network, but at the same time to not disclose the transport layer reachability information to the client layer.

2.3.3.4 Peer vs overlay vs augmented model

All three models have their pros and cons. A common control and management plane in a peer model allows avoiding duplication of the functions performed in these planes in each resource layer (as is the case in the overlay model). Also, due to integration of all resources into a single transport plane, there is no need for standardization of the UNI interface between IP/MPLS routers and OXCs. On the other hand, integration of different (technology) client networks into a single transport plane is difficult. A single control plane in peer model makes all the information (including such confidential information as the transport network topology) freely accessible in the client domain. Besides, the peer model assumes a single carrier owning the transport network, because no NNI interface is defined. In the overlay model, only subsets of (non-sensitive) information are available to client and other carrier's networks due to defined UNI and NNI interfaces. A trade-off between the advantages and disadvantages of the overlay and peer models is provided by the augmented architecture. However, there does not exist yet a clear definition or understanding of the augmented model.

2.4 Network resilience

In our information society more and more services require the survivability of underlying telecommunication transport networks. These are emergency services (such as 112 in Europe or 911 in USA), tele-medicine, e-banking, and stock markets, just to mention a few. Therefore requirements for transport network survivability[2] are very high. The network architectures that were/are used often imply using resilience mechanisms requiring 100% extra capacity in the network for protection purposes. As new services are introduced and demand for capacity grows, there is a need for new network architectures and resilience mechanisms with more efficient capacity utilization, allowing to decrease network cost, at the same time still providing an adequate level of network resilience. Resilience of networks is assured with help of protection and restoration mechanisms, which are discussed later in this section. Before this we will discuss shortly common causes of network failures.

2.4.1 Network failures: causes and statistics

Network failures (also called outages) can be divided into *planned* and *unplanned*. Planned network failures occur due to network maintenance actions that are performed by a network operator. Such failures can be caused, for example, by software or hardware upgrades. This kind of failures is easy to deal with since the operator knows exactly when and where will the failure occur and thus it can prepare accordingly: warn the customers, choose the low activity time (e.g., at night) and, if possible, reroute the traffic away from the affected network elements. Much more trouble is caused by the unplanned outages. These can be due to equipment failures, natural disasters or actions of third parties (such as sabotage, digging accident, etc). Failure rate figures from Telcordia (former Belcore) [19] show that most often link failures are caused by cable cuts (4.39/year/1000

[2]In the sequel, terms 'survivability', 'resilience' and 'recovery' will be used as synonyms.

Equipment type	Range of MTBF (hours)	Typical MTTR (hours)
IP interface card	$10^4 - 10^5$	2
IP router	$10^5 - 10^6$	2
ATM switch	$10^5 - 10^6$	2
SONET/SDH DXC or ADM	$10^5 - 10^6$	4
WDM OXC or OADM	$10^5 - 10^6$	6

Table 2.1: Typical values of MTBF and MTTR for network equipment failures. Adapted from [9].

sheat miles). Another study, which was initiated by regulatory bodies in USA [20], shows that about 60% of cable cuts were due to dig-ups. The study is based on 160 cases. It is also reported that in average the physical repair took 14 hours and the average service outage was 5.2 hours. In order to describe failures of network elements statistically, two parameters are introduced: the mean time between failures (MTBF) and the mean time to repair (MTBR). MTBF specifies average time interval between two subsequent failures of the same network component. After the component has failed, MTTR specifies the average time needed to repair it. Typical MTBF values for a 1000km of cable are in the range of 50 to 200 days. The MTTR for a failure depends on the severity of the failure and location of the failing component and can range from hours to weeks [9]. The failure rate of equipment is often magnitudes less than for cables. Some typical values of equipment failures are given in Table 2.1.

As transmission rates grow, amounts of data carried by telecommunication networks become huge. Therefore a breakage of even a single optical cable would have a big impact, if no recovery is employed. Thus the role of mechanisms assuring network resilience is vital. It is clear, however, that although technical means would allow, it is not economically feasible to protect network from any possible failure scenario. Therefore networks are designed to survive only a subset of all possible failure scenarios (called *accounted failures* in [9]), which have the highest probability of occurring. Taking into account that MTTR is much shorter than MTBF, it is a very low probability that several components will fail simultaneously, and thus, except only for the most critical parts, it is enough to consider only single-link or single node failures. Furthermore, as it was discussed above, cable failures are much more probable to occur than equipment failures. Thus the emphasis in network survivability should be put on resilience mechanisms, allowing to recover from cable cuts.

The following section gives a survey of different resilience options in single

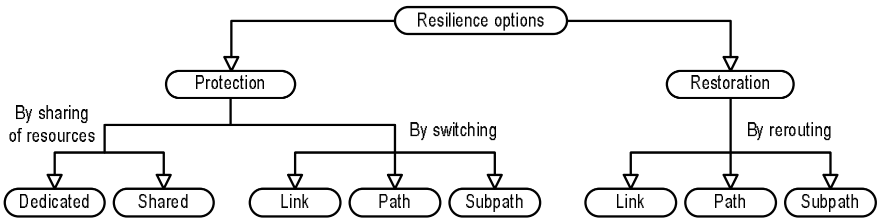


Figure 2.6: Classification of recovery mechanisms.

and multiple resource layers of the multi-layer network and gives their comparisons.

2.4.2 Protection and restoration

According to the way resilience is assured in the network, two broad types of recovery mechanisms are defined. These are 1) *protection* and 2) *restoration* [21]. Classification of different recovery mechanisms is summarized in Figure 2.6 and different mechanisms are discussed in more detail in the text below.

2.4.2.1 Protection

Protection (also called *protection switching*) means that for a primary entity (e.g., primary path) there is a pre-established backup path and that flows are switched to the latter when the former fails. Depending on the span of the backup path, few protection techniques can be defined:

1) *local protection* - when the backup path protects a single link (in this case, the technique is also called *link protection*; see Figure 2.7) or node (the technique is called *node protection*; see Figure 2.8) [22]. Also, *preconfigured protection cycles* (*p*-cycles) belong to the group of local protection mechanisms. The *p*-cycle protection mechanism consists in forming preconfigured cyclic closed paths (Figure 2.9(a)) from spare capacity in a given mesh network, which is used to protect against an on-cycle link (Figure 2.9(b)), straddling link (not on the cycle, but the end nodes of the link are on the cycle; see Figure 2.9(c)) or even node failures. The main advantages of *p*-cycles are fast (ring-comparable) protection switching and little redundancy (overhead of resources, needed for the protection mechanism) [3];

2) *path protection* - when the path spans from source (ingress) node to the destination (egress) node (Figure 2.10);

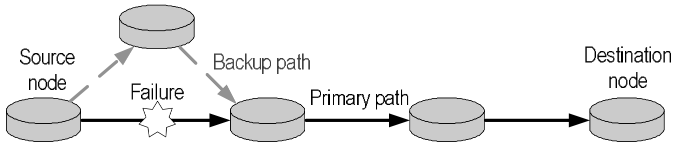


Figure 2.7: Link Protection.

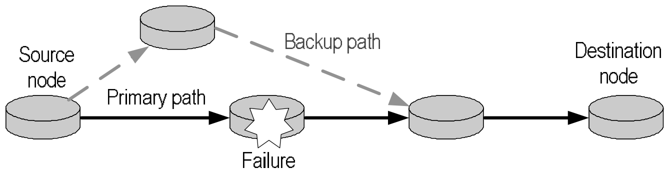
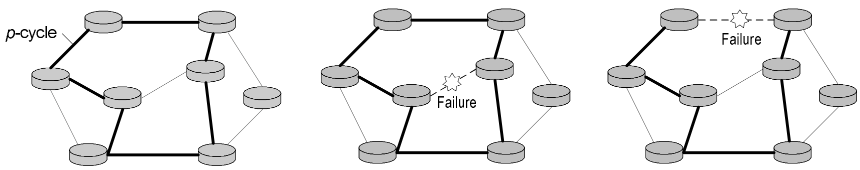


Figure 2.8: Node protection.



(a) A protection cycle. (b) Protection against an on-cycle failure. (c) Protection against a straddling link failure.

Figure 2.9: *p*-cycle protection.

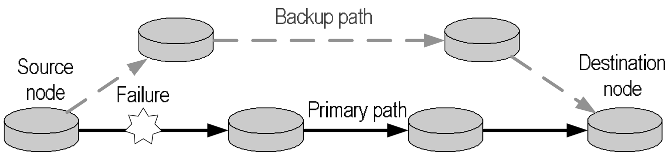


Figure 2.10: Path protection.

3) *subpath protection* - when the path is divided into segments and each segment is protected separately (Figure 2.11). Another way to perform subpath protection is to divide a network into several *protection domains*, where the path segment in each domain is protected only by the resources of the same domain (Figure 2.12). A protection domain *diameter* can be introduced, which is defined as a hop count of a shortest path between a node (belonging to the primary path), where flow is split into primary and backup path, and a node, where the two paths are merged again. Then protection domain could be seen as a generalization for

all protection techniques: setting the diameter to 1 results in link protection, setting diameter equal to a primary path length results in path protection, whereas some intermediate value between the two results in a subpath protection mechanism. Domains must overlap in order to survive link and node failures on the domain boundaries [23; 24; 25].

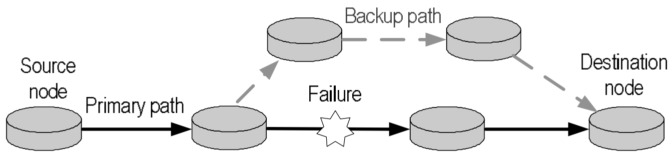


Figure 2.11: Subpath protection.

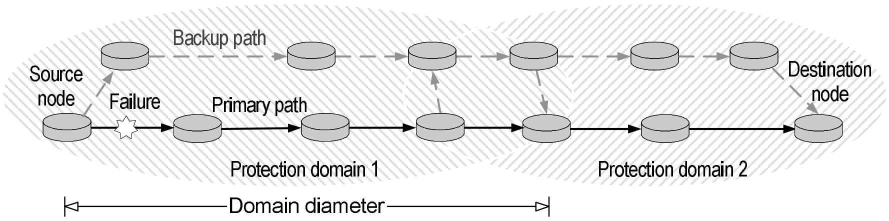


Figure 2.12: Domain protection.

A technique combining features of local and path protection is *local loop-back* (Figure 2.13). In this case, a protection path consists of two parts: a reverse part (from an upstream end node, detecting the failure, to the source) and a diverse part, which is a backup path from source to destination. Protection switching in this mechanism is performed by the detecting node, not the source [22].

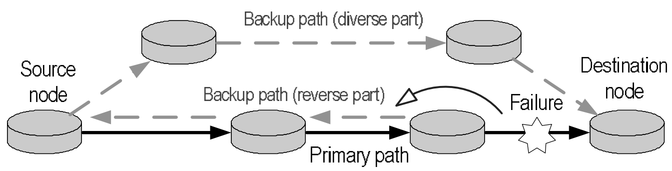


Figure 2.13: Local loop-back protection.

In path protection, primary and backup paths must be *link-disjoint* (not sharing any of the links), so that no single link failure could affect both paths. Node failures can be taken into account by requiring that primary and backup paths are *node-disjoint* (not traversing the same nodes).

A disadvantage of local protection mechanisms is that a backup path is needed per each link/node of the primary path. When network resources on a backup path are dedicated for protecting only one primary path, local protection mechanisms imply inefficient usage of the resources. On the other hand, the switching time associated with protection (from primary to backup path) for the local protection mechanisms is shorter than for path protection mechanisms, because it is performed locally at the detecting node. For path protection, switching is performed by the source node. Therefore more signaling is needed implying longer recovery time, which also means larger amounts of lost traffic, as compared to local protection mechanisms. Advantages of path protection are more efficient utilization of backup resources, as well as usually lower end-to-end propagation delay on the backup path. Subpath protection falls in-between the two techniques (local and path protection), by providing fast recovery switching and requiring modest resource utilization [22; 25].

Protection mechanisms can also be classified according to how the spare network resources are allocated for protection purposes. Based on this criteria *dedicated protection* and *shared protection* mechanisms are identified. In dedicated protection mechanism a protection entity is dedicated to each working entity. Commonly known dedicated protection mechanism are *1+1* and *1:1* [9]. In *1+1* protection the traffic, which is sent on the primary entity (e.g., primary path), is also duplicated on the protection entity (e.g., protection path). A receiving node then chooses between the primary and protection entities, based on some predefined criteria, such as signal quality. In *1:1* protection mechanism only the primary entity is used to carry traffic between the intended source and destination nodes during normal operation. The protection entity can be used to carry additional traffic that can be preempted. When the primary entity fails, traffic is switched over to the protection entity, thus preempting the low-priority traffic there [21].

In shared protection mechanism, there are M dedicated protection entities, which are shared between N primary entities, where $M < N$. Shared protection mechanism are denoted by *M:N*. In a case of failure, only one primary entity can be protected by one protection entity at a time. One of the shared protection mechanisms is *1:N*, where N working entities share one dedicated protection entity [21]. To resolve a situation of which primary entity has a right to use the shared protection entity, when multiple primary entities protected by the same protection entity fail, priorities could be assigned to the primary entities.

An advantage of *1+1* protection technique is that it is very fast, but it is also expensive, since all the resources are duplicated. *1:1* protection technique is slower than *1+1*, since a communication between end nodes is required to perform switching to the backup path. An advantage of *1:1* protection technique is that in normal operation the backup path could be used to carry lower priority traffic, which could be preempted in a case of failure on a primary path. Using *M:N* protection mechanisms can increase system availability with a small increase in total cost. An advantage of using *M:N* mechanisms is, as in a case of *1:1*, that lower priority traffic can be carried on backup paths. A disadvantage is that it is impossible to protect all N primary entities if they fail simultaneously.

Yet another protection approach is *protection by path diversity*. The idea is to split the whole demand volume into a number n of disjoint paths. Assuming a single link failure, only $1/n$ of demand's flow is lost and the rest survives.

In general, shared protection mechanisms better utilize network resources, but recovery time is longer than for dedicated mechanisms. This is due to the fact that nodes (e.g., OXCs) on the protection path cannot be configured in advance (before a failure occurs).

One more important criterion for classifying protection mechanisms concerns the reuse of non-failing resources on the failing primary path, once flows have been switched to the protection path. In this regard we can differ among the protection mechanisms with *stub release* and without it. Stub release implies that when a primary path fails, all the non-failing resources on it are released and can be reused for other paths. When stub release is not used the non-failing resources of the failing primary path remain reserved even when the demand flow is switched to the protection path.

2.4.2.2 Restoration

Restoration (also called *recovery by rerouting*) attempts to find and establish new paths or path segments on-demand after a failure has occurred, in order to restore (reroute) the affected (by the failure) flows. These new paths use the available (unassigned) surviving resources. If there are not enough resources to recover higher priority traffic, lower priority traffic can be preempted [21],[22]. When all the traffic in the network is of the same priority (or no priorities defined) and there is not enough of resources to restore the affected traffic on another paths, the affected flows are blocked. If the restoration mechanism is used in DWDM networks, then it also implies finding a new lightpath together with a wavelength. Three main types restoration techniques exist: 1) *link restoration*, 2) *subpath restoration* and 3) *path restoration* [26]. In path restoration, after detecting a failure on a working path, a source node tries to find and establish an alternative path to the destination using spare (surviving) network resources. In subpath restoration, when a failure occurs, it is the upstream end node (the source node of the failing link, with respect to the direction of the flow), not the source, which tries to find an alternative path to the destination, avoiding the failed segment of the original path. In link restoration, when a failure occurs, the upstream end node tries to find and establish an alternative path to the downstream end node, circumventing only the failed link [26].

It is shown in [27], that in optical networks path restoration is more effective in terms of restoration success ratio (restored/failed connections), while link restoration is more effective in terms of restoration time. Simulation results of an IP over WDM network architecture with GMPLS control plane [26], confirm the above result, also showing that path restoration has the highest restoration success rate, followed by subpath restoration and link restoration. On the other hand, both link and subpath restoration have shorter restoration times (link restoration is the best, followed by subpath restoration) than path restoration.

2.4.2.3 Protection vs. restoration

Usually restoration mechanisms utilize network capacity more effectively, since no spare capacity is allocated in advance. Also, restoration mechanisms are able to provide resilience against more complex (e.g., multiple link) failures, whereas protection mechanisms are designed to cope with a predefined number of simultaneous failures (typically, single link failures). Protection mechanisms, on the other hand, have faster recovery times and provide recovery (availability) guarantees. Restoration mechanisms cannot, in general, provide any restoration guarantees [26].

2.4.3 Single-layer and multi-layer survivability in a multi-layer network

Having a network with multiple layers of resources a natural question arises: in which layer(s) should resilience mechanisms be implemented? And if they are implemented in several layers, how should the coordination between them be performed? It is desired to choose a recovery mechanism which provides a good compromise between availability of network resources, restoration time, resource utilization and implementation/coordination complexity. The following two subsections discuss the options.

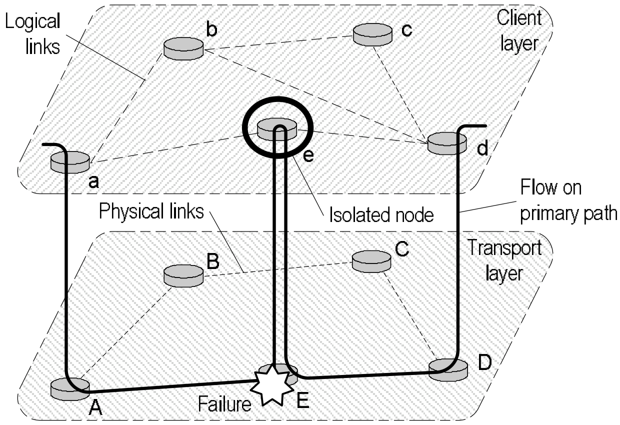


Figure 2.14: Survivability only in the lower layer. Failing lower layer node can leave an upper layer node in isolation.

2.4.3.1 Employing survivability in a single layer

Considering an IP/MPLS over DWDM network, only one of the layers could be chosen to implement a resilience mechanism. The following options are possible:

1)*Survivability only in the lower layer*. In this case, all the recovery actions are performed on the lower (DWDM) layer, independently on where a failure has occurred. Advantages of this method are that recovery actions are performed at the coarsest granularity and failures do not propagate to the upper layers. Disadvantages are that failures in upper layers may not be resolved and in case of lower layer node failure, the upper layer node(s) might be left in isolation. The isolated node problem is illustrated in Figure 2.14. In the depicted situation client layer node e becomes isolated due to the failure of transport layer node E. The lower layer alone is not able to recover the client layer flow $a - e - d$.

2)*Survivability only in the upper layer*. In this case all the recovery actions are performed in the upper layer. Advantages of this method are that it can handle lower layer node or upper layer failures more easily. Also, finer granularity of traffic in the upper layer allows for differentiation of flows for recovery speeds/priority classes, as well as may lead to better capacity utilization. A disadvantage is that several recovery actions may be needed because of the finer flow granularity, as shown in Figure 2.15.

3)*Single layer survivability combinations*. Even if it is only a single layer which has a responsibility for recovering from failure, the choice of which layer will perform the job can be made dynamically. The following two options can be identified: i) *Survivability in the lowest detecting layer* and ii) *survivability in the*

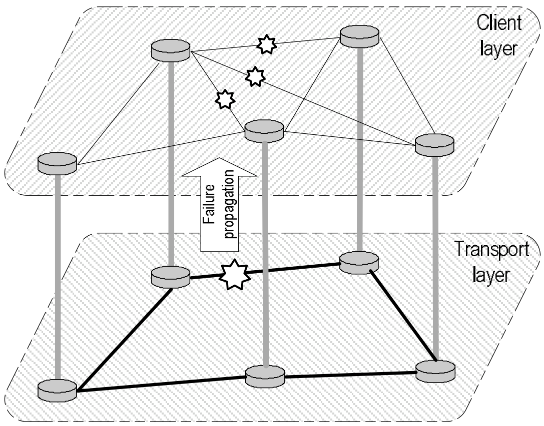


Figure 2.15: Survivability only in the upper layer. Single failure on the lower layer results in many failures on the upper layer.

highest possible layer. An advantage of using the first technique is that upper layer failures are easily handled, although the isolated upper-layer node problem is still persistent. An advantage of using the second technique is that it addresses properly the situation when traffic flows are injected in different layers by recovering flows at the proper granularity (a recovery strategy could also be to deal with a failure in the layer where the affected flows are injected). However, using the second technique many recovery actions may be needed in a case of lowest layer link failure [24].

2.4.3.2 Employing survivability simultaneously in multiple layers

By allowing recovery mechanisms of several layers to cooperate in recovering from failures and sharing the spare capacity among the mechanisms a better network resource utilization can be achieved. The following resilience options can be identified:

1) *Uncoordinated approach*. Recovery mechanisms are installed in several layers. All of them try to recover from failure, but there is no coordination between them. A possible scenario is shown in Figure 2.16. A flow between client layer nodes $a - d$ is disrupted due to a failure of a transport layer link $A - E$, carrying the client layer traffic. Since the transport layer detects the failure first, it initiates a recovery mechanism and reroutes all the flows of link $A - E$ to the protection path $A - B - E$. If the recovery process on the transport layer lasts longer than it takes for the client layer to detect the failure, the client layer initiates its recovery mechanism and reroutes the flow from the primary path $a - e - d$ to the recovery

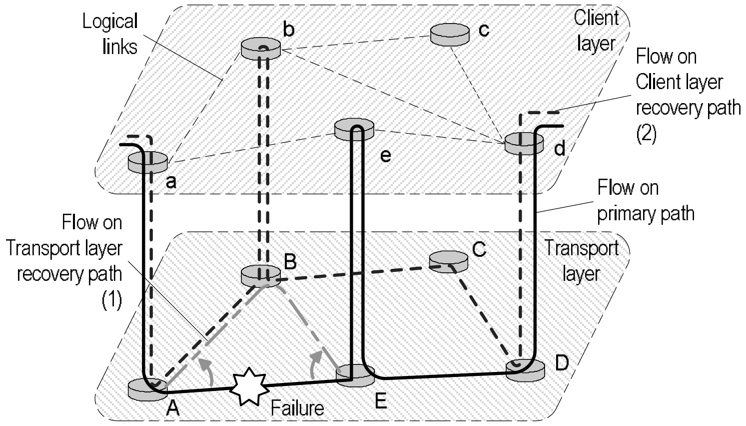


Figure 2.16: Uncoordinated approach.

path $a - b - d$. Thus the affected flow is recovered twice. A main disadvantage of such an approach is wasting of resources, since spare resources needed for a recovery mechanism have to be allocated in each layer. Besides, spare resources allocated in the lower layer (e.g., on link $A - B$ in the above example) have to be sufficient for protecting also the spare capacity of upper layers. Though this spare capacity can be used to carry lower priority traffic during normal operation, a lot of preemptable traffic may be disrupted during the recovery process due to several parallel recovery actions on different layers. An advantage of this approach is that it is simple to install and operate.

2) *Sequential approach.* This recovery approach follows the following principle: if the recovery process of the current layer is not able to recover all the affected traffic (within a predefined time), it hands over the recovery mandate to the recovery process of the next layer above, which then tries to recover the remaining flows and so on. Although the sequential approach can be used with any sequence of layers, two obvious strategies are: i) *bottom-up* - the lowest detecting layer starts the recovery, and recovery actions on the upper layers follow. An advantage of this approach is that recovery is performed at the appropriate granularity; ii) *top-down* - the upper layer starts the recovery. Only if it is not able to restore all flows, the lower layer recovery mechanism is initiated. An advantage of this approach is that higher layers can differentiate traffic and restore high-priority traffic first. An illustration of the bottom-up approach is given in Figures 2.17-2.18. Two client layer flows ($a - e - d$ and $a - c$) are disrupted by a failure of node E in the transport layer. The transport layer is able to recover only flow 2 by switching it from path $A - E - C$ to path $A - B - C$ (Figure 2.17). Then the client layer

takes over and recovers flow 1 on a path $a - b - d$, which in the transport layer takes path $A - B - C - D$ (Figure 2.18).

An obvious disadvantage of sequential recovery approach is that recovery actions in the next layers are delayed, independently of the failure. When implementing sequential recovery mechanism, hold-off timers have to be introduced to prevent several layers from counter-acting. The higher the layer is, the longer the timeout value for the hold-off timer should be. Introducing a token signal, which a recovery mechanism in one layer could use to trigger a recovery action on (hand over the recovery responsibility to) another layer, could make the recovery process faster. This is useful in situations when a recovery mechanism on a current layer identifies that it cannot do anything more to improve the situation, earlier than the hold-off timer expires.

3) *Integrated approach*. This approach assumes a common integrated recovery mechanism across all layers. This is the most flexible, but also most complicated recovery approach. It implies that recovery mechanisms in all layers are coordinated by the common control plane having information about failed connections, spare resources in each layer and capabilities of the recovery mechanisms. In this way surviving resources in all layers can be used in a most efficient way when recovering from failure [24].

Different layers can implement any of the recovery mechanisms presented in Section 2.4.2.

2.4.3.3 Choosing a recovery mechanism

Each of the discussed recovery mechanisms has its pros and cons. In general, all the recovery mechanisms present a trade-off between recovery time (which also implies the amount of lost traffic) and extra needed capacity. For example, restoration mechanisms are very capacity-efficient, but take longer time to recover failed flows compared to protection mechanisms. On the other hand, a 1+1 protection mechanism requires double the working capacity to be installed in the network, but provides instant recovery. In this thesis we are concerned only with the capacity aspect of the recovery mechanisms.

Trade-offs of different resilience schemes in IP/MPLS over DWDM networks have been studied in [28; 29]. The studies show that better network performance may be achieved if restoration mechanism are employed on the IP layer and protection mechanism are used on the DWDM layer.

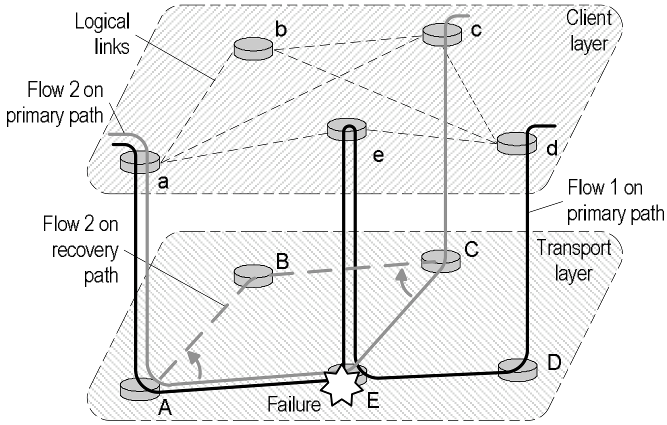


Figure 2.17: Bottom up approach, phase1.

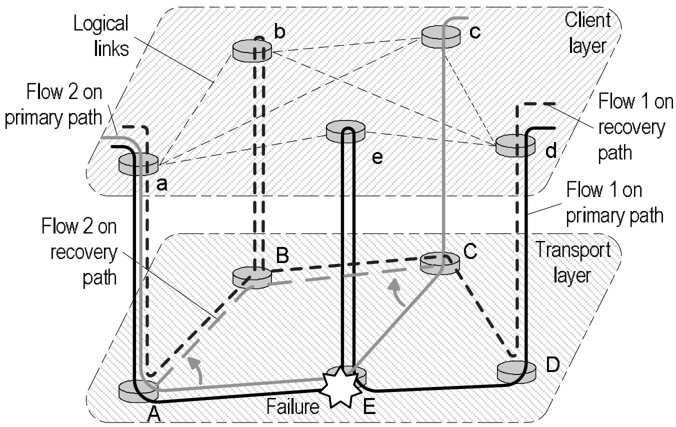


Figure 2.18: Bottom up approach, phase 2.

CHAPTER 3

Modeling framework

3.1 Optimization models and algorithms

3.1.1 Optimization models

In this thesis network design problems (NDPs) are presented as optimization models (OMs). An OM is given in a form of an objective function and a set of constraints, defining a region $\mathcal{S}$ of feasible values that problem variables can assume. The aim is to find values of problem variables (decision variable) within the feasibility region such that the objective function is minimized (alternatively—maximized). A general form of an OM is as follows:

$$
\begin{aligned}
&\text{Objective:} \qquad && \min \ \boldsymbol{f}(\boldsymbol{x}) && && (3.1)\\
&\text{Subject to:} && g_i(\boldsymbol{x}) \geq 0 && i = 1, 2, \ldots, I && (3.2)\\
& && h_j(\boldsymbol{x}) = 0 && j = 1, 2, \ldots, J && (3.3)\\
& && \boldsymbol{x} \in \boldsymbol{X}. && && (3.4)
\end{aligned}
$$

In the formulation above, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathcal{S}$, $\mathcal{S} \subseteq \mathbb{R}^n$, is a vector of optimization (also called *decision*) variables. Values that decision variables can assume are limited by problem constraints. It is convenient to divide problem constraints into *inequality constraints* (3.2) and *equality constraints* (3.3). Other constraints, such as those implying non-negativity or integrality of variables, are grouped and represented by set $\boldsymbol{X}$, as in (3.4).

An *objective* (also called *goal*) function (3.1) is in general a vector function composed of m optimization criteria, that is, $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$, where each $f_k(\boldsymbol{x})$ is a real-valued function ($f : \mathcal{S} \rightarrow \mathbb{R}^1$).

For $m = 1$ we deal with a scalar objective function and then objective (3.1) is well defined. Clearly, minimization and maximization problems are equivalent, since min$\boldsymbol{f}(\boldsymbol{x})$ is the same as $-$max$(-\boldsymbol{f}(\boldsymbol{x}))$. The problems with more than one criterion to be optimized are called *multi-criteria* (also called *multi-objective*) optimization problems [30]. Multi-criteria (MC) optimization implies optimizing several conflicting criteria subject to given constraints. For a multi-criteria optimization problem there usually does not exist a solution that simultaneously minimizes all the criteria to the lowest possible value within $\mathcal{S}$. Thus, a solution to a multi-objective optimization problem is a tradeoff between competing criteria. Typically, if an MC optimization problem reduced to any single objective criterion $f_k(\boldsymbol{x})$ has a unique solution, the solution to the full MC problem is in general a (possibly infinite) set of Pareto solutions. Recall that a Pareto solution is a point in $\mathcal{S}$ for which any single criterion in the multi-criteria objective can be improved only at the expense of making some other criterion worse. The main difficulty arising when solving MC optimization problems is how to judge which objective criteria vector is the best (optimal).

There is a number of ways for solving MC problems. First of all, a single aggregate objective function can be formed, by summing all the criteria functions possibly multiplied by weighting coefficients. Obviously, the solution will depend on the chosen weights, thus choosing them is in general difficult. Second simple approach is to use *lexicographical optimization*. In lexicographical minimization we are in first place interested in minimizing the first criterion $f_1(\boldsymbol{x})$. Then, among all the points that attain the minimum of $f_1(\boldsymbol{x})$ we minimize $f_2(\boldsymbol{x})$, and so on. In effect, we solve a sequence of single-criterion optimization problems one by one, and the vector of goal functions is optimized component by component. Yet another technique for solving an MC optimization problem is to look for a *min-max* solution vector. This approach is similar to lexicographical optimization, except that what we need to find is the lexicographically minimal criteria vector $\boldsymbol{f}(\boldsymbol{x})$ *sorted* in non-decreasing order. The lexicographical and min-max (in fact max-min) optimization techniques will be described in more details in the following chapters. The techniques are extensively covered in [31].

In this thesis most of the optimization models employ MC optimization, and all the three approaches for resolving the problems are used.

Optimization problems can be divided into several classes. For the clarity of the discussion let us assume for now that in an OM we have single-criterion objective function $f(\boldsymbol{x})$. If f is convex, g_i is concave and h_j are linear functions of $\boldsymbol{x} \in \mathbb{R}^n$, the problem is called a *convex programming problem* (CXP). One important property of convex problems is that local optimality coincides with global optimality. The sufficient conditions of optimality for a convex problem are defined by the *Karusch-Kuhn-Tucker conditions* [32].

When all $f,\ g_i,\ h_j$ are linear and $\boldsymbol{X} = \mathbb{R}$, the OM is called the *linear programming problem* (LP). For problems in this class the solution space $\mathcal{S}$ is a convex polytope defined by the linear constraints [33]. A *simplex algorithm* created by G.B. Dantzig in 1947 is widely used for resolving LPs.

Integer programming problems (IPs) are those were $\boldsymbol{X} = \mathbb{Z}$ and, as before, all $f,\ g_i,\ h_j$ are linear. Problems, where some variables are continuous and some are integers, are called *mixed integer programming problems* (MIPs). Problems with discrete variables forming a finite set are called *combinatorial optimization problems*. Both LP and MIP problems are used in this thesis to model network design tasks. For an in-depth coverage of network modeling the reader is referred to [31].

3.1.2 Overview of resolution methods for MIP and IP problems

A general framework for exact solving of MIPs is *branch-and-bound* (B&B) [32], [34], [35], and its cutting plane-based enhancement called *branch-and-cut* (B&C) [36], [37], [38]. Efficiency of B&B greatly depends on the quality of the lower bounds provided by the linear relaxation of the MIP subproblems solved in the nodes of the B&B tree. In general, the quality of the lower bounds depends on a particular MIP formulation; for example the use of so called "big M" is disadvantageous in this aspect. However, sometimes a problem formulation cannot be stated without the "big M". The reader will encounter several such formulations in the thesis.

The quality of lower bounds can often be improved by the use of *Lagrangian relaxation* (LR) instead of linear relaxation (the use of LR for solving multi-layer problems was studied in e.g., [39]). Still, in many case the lower bounds can be most effectively improved by introducing additional linear inequalities (called *cuts* or *valid inequalities*), preferably defining facets of the MIP solution polyhedron. Such cuts can be derived either from the general *cutting plane method* [32], [40]), or using problem-specific valid inequalities [41] (note that using problem-dependent inequalities gives a better opportunity to find the facets). The possible types of general cuts are for example Gomory fractional cuts, mixed integer rounding, disjunctive inequalities, super-additive inequalities (see [40]), and those used in the lift-and-project method for binary MIPs [37]. In the case of the network design problems studied in this thesis, the problem-dependent cuts are basically knapsack-based inequalities, considered in the literature both for modular dimensioning ([42], [43], [44]) and unsplittable flows ([45], [46]). An example of such type of cuts are cover inequalities and cover-lifted inequalities, see [47]. Additionally, for unsplittable flows, super-additive inequalities have recently been applied [48].

General purpose MIP solvers, such as CPLEX [49] or XPRESS-MP [50], use the B&C approach with in-built general purpose cuts. Problem-specific valid inequalities can be used either in own-developed B&B algorithms, or can be added to the general purpose solvers. It should be emphasized that although problem-specific algorithms based on B&C are potentially capable of attaining good results in exact (or near-exact) resolving of the OMs considered in this thesis, in practice it is very hard to beat general purpose MIP solvers (for example CPLEX) both in terms of execution times and the quality of the final solution. Still, even such solvers may not be effective enough already for solving problem instances of medium size, because of intrinsic difficulty of the studied network design problems.

For practical purposes it is usually sufficient to solve a problem in hand by means of some approximate (or heuristic) algorithm if the gap between the quality of the approximate solution (called the upper bound) and the exact solution (usually estimated with its lower bound) is small enough, for example not greater than 5%. Such approximate algorithms can be time efficient in contrast to their exact counterparts. Also, the exact and approximate solution methods can be combined, as it will be illustrated in Chapter 7 of the thesis.

Since capacity in telecommunication networks is often over-provisioned (to account for the future traffic growth), exact solutions to network design problems are not always used directly, but merely as a support tool in taking a decision during network planning process. Therefore, in some cases finding a lower bound on the solution (for example, by solving the relaxed problem) may already provide the necessary information and radically reduce the solution time.

Due to the above-mentioned reasons in the thesis we mostly study problem relaxations, decomposition methods, and an approximate problem resolution.

3.2 Network modeling

3.2.1 Single layer network model

For modeling purposes a telecommunication network is typically represented as a graph (see Figure 3.1). Vertices of the graph, representing network nodes, are interconnected by edges, representing links. Links can be *undirected* and *directed*. An undirected link allows flow transfers in both directions, while a directed link allows the transfers in only one predefined direction. Links have a limited capacity, which can be given (fixed), or variable, leading to *capacitated* or *uncapacitated* problems, respectively. Directed links can be of two types: links with shared capacity among flows in both directions, and those with dedicated capacity in each

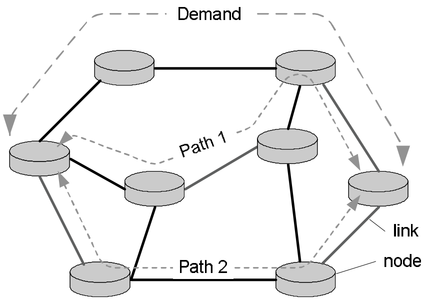


Figure 3.1: Network model.

of the directions.

A *demand* between a pair of nodes is a requirement for a certain amount of bandwidth called *demand volume* between two nodes of the network. For non-elastic demand a demand volume is a given constant. In the case of elastic traffic, a demand represents a requirement for available bandwidth (possibly bounded from below and above) between the two nodes. It is assumed that a list of demands is given. For each demand, a list of *candidate paths* (routing list) is specified. A *path* (route) is a an ordered list of links making possible to transfer a flow between the end nodes of a demand. If the required demand volume of a demand is realized by flows on several paths (from the candidate path list), then the flows are called *bifurcated*. Contrary, when for each demand its demand volume is allocated to one single path, the resulting flow pattern is called *non-bifurcated*, and the corresponding network design problem is called *unsplittable*.

In all the models presented in this thesis the so called link-path (L-P) problem formulation is used. The L-P formulation is convenient, since it allows to model both *directed* and *undirected* links and flows, as well as to have effective control on the type of allowable paths. Thus, only networks with undirected links are considered in this thesis.

3.2.2 Two-layer network model

A multi-layer network is modeled by means of a set of interconnected single-layer models and a specific feature of multi-layer network modeling is that relationship between the layers is be taken into account.

For the two-layer case the network model is represented by two overlaid graphs corresponding to each of the layers (see Figure 3.2). Like in the single layer network case, a network for each of the layers is modeled as an undirected graph. Links of the upper layer (also called client/traffic/data layer) are labeled with e, where $e = 1, 2, \ldots, E$ (alternatively we will denote this as $e \in \mathcal{E}$). Each link e is characterized by its capacity. If the capacity is a given constant we denote it by c_e, and if it is a variable of an optimization problem we denote it by y_e. A demand d ($d = 1, 2, \ldots, D$, alternatively $d \in \mathcal{D}$) between source and destination (S-D) nodes is a requirement for a certain amount of bandwidth. In the elastic traffic case, lower bound h_d and upper bound H_d for the demand volume are defined, and the demand can be assigned any bandwidth between the bounds. D is the number of demands in the upper layer. Each demand d is realized by flows x_{dp}, where index $p = 1, 2, \ldots, m(d)$ (also denoted as $p \in \mathcal{P}_d$) labels paths for flows realizing demand d. The total flow X_d, realizing demand d, is the sum of flows assigned to all paths of the demand and is calculated as $X_d = \sum_{p=1}^{m(d)} x_{dp}$, $d = 1, 2, \ldots, D$. Having defined the upper layer, it is recognized that the link capacities of this layer are the demand volumes for the lower layer (usually the transport layer) that have to be realized by the path flows in the lower layer.

Entities of the lower (for example optical transport) layer are defined as follows. The lower layer network is interconnected by (for example optical) links labeled with g, $g = 1, 2, \ldots, G$ (or simply $g \in \mathcal{G}$), of capacity u_g. If the capacity is a given constant then we denote it by c_g. There is a *unit cost* ξ_g associated with each link, which is the cost of installing one capacity unit on the link. Demands for the lower layer are the link capacities of the upper layer. Therefore demands of the lower layer are indexed with e and flows of the lower layer realizing demands e are denoted by z_{ek}. Index $k = 1, 2, \ldots, n(e)$ (also denoted as $k \in \mathcal{K}_e$) labels the paths for flows realizing demand volume (i.e., capacity) associated with link e. In this model all nodes of the upper layer must exist in the lower layer as well, but the reverse implication is not necessarily true. This is because the transport network can in general have some transit nodes that are not terminating end-to-end traffic of the upper layer. The nodes can be either the routers that have double functionality (e.g., they act as IP routers as well as DWDM OXCs), or the terminating nodes in optical network (e.g., pure OXC nodes or WDM terminal nodes).

An example of routing in such a two-layer network is given in Figure 3.3. Say that we have a demand in the upper layer between nodes a and d. In the upper layer it is realized by the flow routed on a path $a - b - d$. However, the capacities of the IP links are realized by the flows in the lower layer. More precisely, capacity of the upper layer link $a - b$ is realized by lower layer flow $A - B$, and the capacity of link $b - d$ is realized by the flow $B - E - D$. Thus, the actual path that flow between nodes a and d takes is $a - A - B - b - B - E - D - d$. The two-layer network model, presented above can be easily extended to more layers.

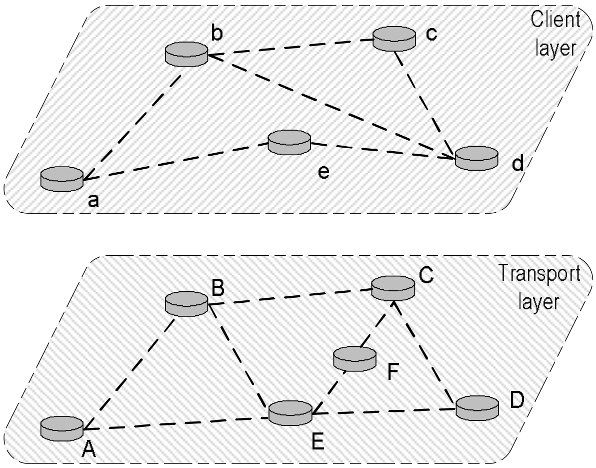


Figure 3.2: A network with two resource layers.

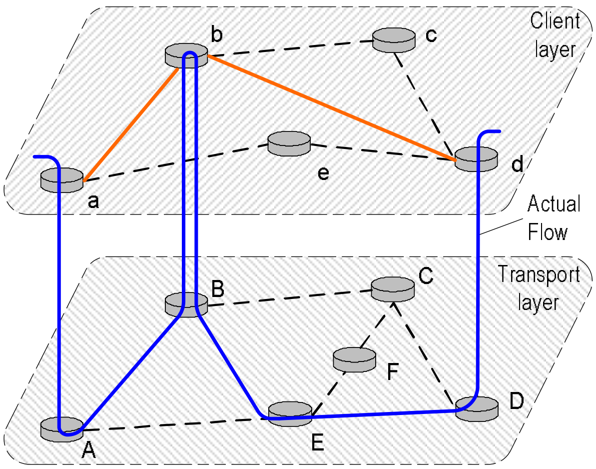


Figure 3.3: Routing in a two-layer network.

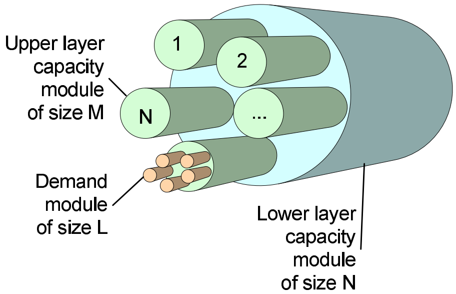


Figure 3.4: Relation between capacity modules.

3.2.3 Modeling of modular link capacities

In reality both network demands and link capacities are typically installed in modules of a fixed size. To model this we denote by L the size of the upper layer demand module. Also, the upper layer link capacities are installed in modules of size M, and the lower layer link capacities are installed in modules of size N.

Let us consider an IP/MPLS over DWDM network. In general, module sizes M and N have the following interpretation (See Figure 3.4 for illustration). If demands in the IP/MPLS layer are imposed in units of size L, then M represents the number of the L-size units realized by one module of capacity of the upper layer links. Consequently, the upper layer link capacity y_e expresses the amount of the modules installed on a link e; observe that in total the link can carry up to $L \cdot M$ demand modules. A module size N represents a number of M-size modules. The lower layer link capacity u_g gives an amount of modules N installed on link g. For example, if demands in the IP/MPLS layer are imposed in STM-1 sizes, i.e., $L = 155, 52$ Mbps, and $M = 64$, then installing $y_e = 1$ module M on link e results in capacity of the link of $9, 952$ Gbps (which corresponds to STM-64), and in DWDM layer it requires one OC-192 channel, i.e., 1 wavelength. Then, if $N = 128$, and $u_g = 1$ module of size N is installed on link g, it corresponds to installing a fiber with 128 wavelengths, carrying approximately 10 Gbps each.

Hence the size of module M can be chosen to be equal to capacity carried by one lightpath in DWDM network. Then, N represents the number of wavelengths per fiber and u_g gives the number of fibers to be installed on a link.

3.2.4 Modeling of failures

In this thesis network resource failures are modeled by link *availability coefficients* $\alpha_{gs} \in [0, 1]$ assigned to the lower layer links, where s ($s = 0, 1, \ldots, S$, or simply

$s \in \mathcal{S}$) labels the *failure states* (also called *failure situations*) and S is the number of the considered states. State $s = 0$ is called the *normal state* and corresponds to the case when all links in the lower layer are fully functional and available. A value $\alpha_{gs} = 0$ means that link g is totally failed (unavailable) in state s, whereas $\alpha_{gs} = 1$ implies that link g is fully available in state s. A fractional value of α_{gs} represents a partial link failure. Since availability coefficients are defined for links, to model a node failure, coefficients α_{gs} of the links incident to the failed node have to be set to a value which represents the level of the node failure. For example, setting $\alpha_{gs} = 0$ for all links incident to a certain node v would mean that node v has completely failed. Note that if node v appears in the upper layer as well, then in this case all the demands originating (terminating) in node v cannot be realized and must be discarded.

The notions presented in the previous section can be extended and made state-dependent. Flows of the upper and lower layers can now be made state-dependent and defined as x_{dps} (flow realizing demand d on path p in state s) and z_{eks} (flow realizing demand e on path k in state s), respectively. The total flow, realizing demand d in state s is then defined as $X_{ds} = \sum_{p=1}^{m(d)} x_{dps}\ ,\ d = 1, 2, \ldots, D\ ,\ s = 1, 2, \ldots, S$. Similarly, the capacities of the upper and lower layer links can be defined as y_{es} and $u_{es} = \alpha_{gs} u_g$, respectively.

CHAPTER 4

Fairness in data networks

4

4.1 Elastic traffic

Traffic in packet networks can be roughly divided into two broad categories: i) *non-elastic traffic* and ii) *elastic traffic* [51]. Non-elastic traffic is generated by services and applications that require strict quality of service guarantees for minimum available bandwidth, maximum allowed packet loss, delay, jitter, etc. This type of traffic is, for example, generated by streaming (real-time) services and applications such as traditional telephony and IP telephony (VoIP), video conferencing, video broadcasting over IP (IPTV), video on demand (VOD), etc.

The other type of traffic is elastic traffic which, in fact, constitutes the majority of the traffic in the Internet. The elastic traffic is generated by "greedy" services and applications, i.e., those that can and do adapt their instantaneous packet rate to the available bandwidth in the network. Elastic traffic is, for example, generated by applications using Transmission Control Protocol (TCP) over the Internet, or by Available Bit Rate service in ATM networks. Examples of applications and services generating this type of traffic are web browsing, newsgroups, file downloads, and peer-to-peer applications.

The on-going evolution of telecommunications networks into *multi-service* networks imply that traffic of both types (elastic and non-elastic) is carried by the same network. This fact is taken into account when defining network models in this thesis.

4.2 Demand and supply of bandwidth

4.2.1 Traffic growth

The share of elastic traffic in today's networks constantly grows, mainly due to the P2P traffic. Recent studies [52; 53] show that the P2P traffic constitutes as much as 60% of all the traffic in the backbone networks of broadband service providers and in the Internet. According to another study [54] analyzing the Internet traffic in Germany, depending on the time of day, the P2P traffic constitutes between 30% (during daytime) and 70% (at night) of the total traffic. Also, more and more software companies, open software projects, and content providers choose to distribute their software products or multimedia content using the P2P technology, or use the existing P2P networks directly. For example, many Linux distributions such as Debian, Gentoo, Knoppix, Fedora/RedHat, Ubuntu, Mandrake, Slackware etc., can be downloaded via P2P. TV/video distribution is also moving towards using the P2P technology. Examples are the recently launched TV/video content distribution network Joost [55], and BBC development efforts of a VOD system using P2P technology [56; 57]. Thus the requirement for bandwidth is moved from the servers of the software/content providers, to the broadband network providers of the end users, and so are the costs of the capacity. All these new network usage paradigms imply that elastic traffic in the backbone networks constantly grows, and thus network operators have to address it.

4.2.2 Capacity expansion and traffic management

The network operators used to address the growing need for bandwidth by constantly extending the capacity of their networks. According to [58] the installed bandwidth in the networks increases at a rate of 300% per year. The leading service providers report that bandwidth in their networks doubles about every 6 to 9 months [59]. The larger network capacity allows the operators to sell faster broadband connections, which stimulate yet wider development and usage of P2P and other bandwidth-demanding services and applications, that in turn create a demand for even more bandwidth. However, in this closed loop between network providers, service providers, and end users, network providers typically gain only from selling the connections, but not from the services that are using their networks for transporting the traffic. Thus the network capacity expansion induces the capital expenditure (CAPEX) costs for the operators, but does not create additional revenue from the broadband services (just for the connection itself) which their networks are used for, and therefore give no adequate return on investment. In order to both satisfy constantly growing demand for bandwidth and limit CAPEX many network providers oversubscribe their networks. The oversubscription is based on the statistical models and observed network usage patterns, which allow for an assumption that all the users will never be connected to the network at the same time to consume all its resources [53]. Thus the users can expect that sometimes they will not be getting the maximum throughput that was promised, because they have to share the network capacity with other users. In the case when users connect to the Internet via a cable television (CATV) [60] or residential/community/campus LAN network, the bottleneck is in the premises of the provider's network. If, on the other hand, they connect via any type of digital subscriber line (DSL), they have to share the resources in the core of the network. And if some of the users engage in heavy bandwidth usage, the other users that share the same bottleneck resource can experience degradation of a service due the bandwidth shortage. The situation tends to be of the type which is described by the *Pareto principle* (a.k.a. the $80 - 20$ *rule*) [2], i.e., where 80% of resources are consumed by less than the 20% of users, applications and services [53]. The problem is especially profound on the upstream links, where capacity is particularly limited [61]. This is due to the symmetrical behavior of the P2P applications which on average consume about 80% of the upstream capacity in access networks [52]. The symmetrical capacity consumption implied by P2P creates especially many problems in legacy access networks that were designed to address a usage scenario, where a user mainly downloads and does not upload much. The current situation is not only unprofitable for the operators, but is also unfair towards some users of the network. Thus there is a need for mechanisms enabling fair access to the available resources for all users and services.

Therefore the operators have taken various efforts in order to limit the users' and the "free-riding" service providers' traffic. Some of the operators are charging their users for additional traffic, once some total (upload and download) monthly quote is exceeded. This business model is based on the idea that there is no incentive for users not to be greedy, unless the over consumption costs them extra. However, the competition for the customers on the market leads to the situation where more and more operators offer the connection at a flat monthly rate. Other operators try to throttle or limit the P2P traffic [62; 63; 64; 65]. Still, such efforts can lead to loosing the customers, since most of the customers buy the fast broadband connection in order to use P2P services/applications in the first place. Attempts to filter the traffic of the free-riding service providers often raise the discussions of the antimonopoly flavor. Besides, filtering of the traffic using traditional firewalls does not come easy either, since many of the P2P-related services are able to tunnel the traffic over the ports of services (like HTTP[1]) that may not be filtered. The recent achievements in deep packet inspection (DPI) technology [2], allow to extract information from each packet on all protocol layers (Layer 2 through Layer 7 of the OSI model) at a wire speed. As an outcome, each packet may be redirected, tagged for some priority handling, rate-limited, blocked, etc.,

[1] Hyper text transfer protocol; the data transfer protocol used for web.

depending on the operator's policies. With this tool at hand operators are able to manage the traffic of different types, and even for each user separately. However, the possibilities that DPI opens raise a lot of discussions concerning user privacy. Also, operators' intention to use the DPI technology for limiting or blocking traffic of some types (e.g., P2P, or traffic from some content providers) resulted in protests from some content providers and public interest groups that are requiring legislation assuring "net neutrality" [66; 67]. Net neutrality would mean that network providers are not allowed to discriminate traffic of any type in their networks, unless it is malicious. This includes filtering or limiting the traffic of P2P, as well as traffic for services of third parties. Despite all the ongoing debates the DPI based solutions [68; 69] are available from many equipment vendors and are finding their way into the operators' networks.

Regardless of which mechanism network providers will choose for managing the traffic in their networks, it is clear that they have to assure fair flow allocation, both among users and applications. First of all, the fair bandwidth allocation is necessary to be able to give equal opportunity to all users to benefit from their broadband connection, thus assuring their satisfaction with the network. Secondly, the fair distribution of the available bandwidth between services and applications is needed for assuring their operation with certain QoS at all network states. Furthermore, all these guarantees have to be provided while conserving network resources.

The rest of this chapter is devoted to presentation of models and algorithms for designing networks with fair flow allocation. They can be used as a support tool for network operators in the network planning process. We will start by first introducing several fairness criteria that are often encountered in the context of networking.

4.3 Concept of fairness in data networks

Though the concept of fairness has been widely studied in the field of economics (as early as in 1912 [70]), its application to telecommunication networks is rather new and traces its origins back to late 80's [71]. It should be clear from the discussion above, that the idea of fair bandwidth sharing is currently gaining more and more interest in the context of both access and backbone networks. A need for fairness is prominent for example in situations when congestion arises due to excessive offered traffic so that some of the traffic can be entirely rejected. It is important then that all users/services/applications still get their fair share of bandwidth and none is discriminated by total rejection of its traffic. Fairness is important since it increases users' satisfaction with the network and its services. Still, fairness can decrease the total network throughput.

As opposed to total network throughput maximization (TM), profitable from the network operator's point of view (although likely greatly unfair for some customers), many alternatives exist that make the allocation more fair for the users than TM, still maintaining cost efficiency of the TM mechanism. For example, probably the simplest fair flow allocation principle would be to assign the same amount of bandwidth to all users, at the same time trying to make the assigned amount as large as possible. However, this principle can degrade the total network throughput. The reason for this is that there may be some spare capacity left after such an assignment, which could be used to assign more bandwidth (than this minimum) to some of the flows. The issues of economical and at the same time fair allocation of bandwidth in networks carrying elastic traffic have been studied in [31; 71; 72; 73; 74; 75; 76] and many other publications.

Fair allocation can be defined in many ways. Since fairness is a very subjective measure, it is not at all clear which fairness criterion reflects user expectations in a best way, and yet more—how to define it. Often, utility and welfare functions are introduced to facilitate definition and comparison of different fairness criteria.

4.3.1 Utility and welfare functions

As in most of other systems, resources in telecommunication networks are limited and thus there is a number of users competing for these resources. When we intend to distribute the resources fairly among the competing users, a question arises of how to define fair allocation. A common sense definition of fair allocation, which stems from the extensive studies of fairness in economics, is that such an allocation is envy-free [77]. This means that nobody prefers someone else's allocation over ones own. It is clear that distribution of resources among users according to this definition of fairness depends on every user's understanding/preferences of what is fair. Thus, it implies that every user has to communicate his preferences to some entity responsible for resource distribution in the network, if such fairness model is adopted. Yet more, users would have to signal their preferences to the network as soon as any change in resource distribution has occurred. Clearly, this is not a practical mechanism for large telecommunication networks due to scalability reasons. Therefore a concept of *utility function* has been introduced. Utility function $U_d(\cdot)$ is defined for every network user (demand[2]) d and maps a

[2]Note that different utility functions may be associated with user's different connections or particular applications. In order to have a generic definition, we assume that the utility function is defined per demand. Further, terms 'user' and 'demand' will be used interchangeably.

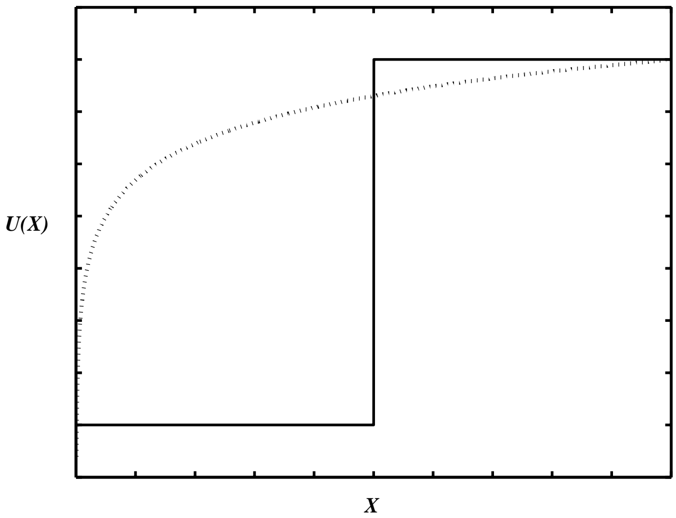


Figure 4.1: Examples of utility functions: for elastic traffic (dotted line) and real-time traffic (solid line).

service delivered by the network into some satisfaction measure (utility value) of the user. The utility value is used only for ordering of user's preferences, and in general has no meaning in absolute terms. For each user, received service (resource allocation) giving higher utility value is preferred. For example, if a user prefers an allocation amount x over $\hat{x}$, then the utility value $U(x) > U(\hat{x})$. An example of a utility function is $U(X_d) = \log(X_d)$. We will return to this utility function when discussing proportional fairness principle.

A network service can be characterized by any relevant subset of measures, such as transmission rate, delay, jitter, loss, etc. Further in this thesis we will be concerned mainly with assignment of transmission rate (bandwidth) to demands, thus we assume that a service is described by this characteristic alone. Therefore we can write a utility function for demand d as $U_d(X_d)$, where X_d is the transmission rate. Assuming that users are willing to pay in order to get better service, the utility can also be seen as a measure of how much a user is willing to pay for the particular service level [51].

Figure 4.1 illustrates utility functions for two types of demands: elastic (best-effort) traffic generated by e.g., file transfer application (depicted by the dotted line), and real-time traffic generated by streaming audio or video application (depicted by the solid line). In general, the utility function for elastic traffic is increasing, concave, and possibly differentiable function of X, while the function for a real-time traffic is a step function, with the step occurring at the minimal transfer rate required for the real-time service to be of acceptable quality.

Given a set of utility functions for the users, it is not at all straightforward how to find the allocation of resources to users, because in general there exist many feasible solutions. Therefore it is useful to introduce a *welfare function* $W(U_1, U_2, \ldots, U_D)$, where D is the number of network demands, that aggregates the individual utility functions U_d of the demands. A welfare function has to be increasing in all of its arguments, and it reflects a network-wide resource distribution strategy [78]. An example of welfare function is $W(\boldsymbol{U}) = \sum_d U_d$, which, when maximized, implies maximization of a total utility in a network.

In the domain of telecommunication networks, fairness and fair resource allocation usually refers to allocation according to some collection of utility and welfare functions. Two of the most often encountered fairness principles in the field of telecommunications are max-min fairness and proportional fairness.

4.3.2 Max-min fairness

Max-min fairness (MMF) corresponds to the welfare function $W(U_1, U_2, \ldots, U_D) = min(U_1, U_2, \ldots, U_D)$, where the users' utility functions are $U(X_d) = X_d$ for all d. Maximizing the welfare function W results in a max-min allocation $max(min(X_1, X_2, \ldots, X_D))$. However, this is just a first step of the max-min allocation, which guarantees that each user gets the same minimal utility, and this minimum is as large as possible. However, the welfare can be possibly improved by applying the operation of maximization iteratively. The iterative procedure consists in performing the following in each of the iterations: first fixing the assignment of resources for these users for which it can not be improved any more, and then trying to maximize it for the rest. Value of welfare achieved by iterative application of the max-min procedure is in general greater than the value resulting from the first-step only. When referring to MMF allocation in this thesis we assume the allocation achieved by the iterative procedure.

The max-min fairness criterion was proposed in 1971 by John Rawls in his book "*A theory of Justice*" [79], a fundamental work in the area of social justice. In the context of data networks the MMF notion can informally be stated as follows: a vector of bandwidth allocations $\boldsymbol{X}^* = (X_d^* : d \in D)$ is *max-min fair* if it is feasible (that is: $\boldsymbol{X}^* \geq 0$ and the link loads generated by $\boldsymbol{X}^*$ do not exceed link capacities), and for each $d \in D$, X_d^* cannot be increased (while maintaining the feasibility) without decreasing $X_{d'}^*$ for some d', for which $X_{d'}^* \leq X_d^*$.
Formally this can be expressed as follows (see Figure 4.2 for illustration) [71]:

Definition 4.1. $\boldsymbol{X}^* = (X_d^* : d \in D)$ is *max-min fair* if it is feasible (that is: $\boldsymbol{X}^* \geq 0$ and the link loads generated by $\boldsymbol{X}^*$ do not exceed link capacities), and for each feasible (as above) $\bar{\boldsymbol{X}}$, and for each $d \in D$, if $X_d^* \leq \bar{X}_d$, then for some d', $X_d^* \geq X_{d'}^*$ and $X_{d'}^* > \bar{X}_{d'}$.

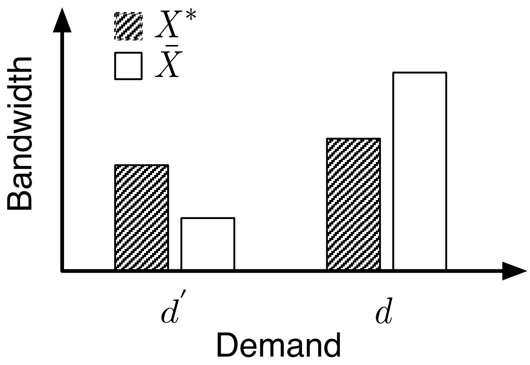


Figure 4.2: Defining MMF allocation vector $\boldsymbol{X}^*$.

Thus, bandwidth allocation is MMF, if it is not possible to increase allocated bandwidth for any user, without decreasing it for another user with equal or smaller allocated bandwidth. Since the MMF principle maximizes the minimum allocation, which is assured for everyone, it is argued to be the most fair principle.

In some cases it can be of interest to consider a *weighted max-min fair* allocation. The weighted MMF allows to take into account different need for bandwidth among the demands. Given a vector of weights $\boldsymbol{\omega} = (\omega_d : d \in D)$, where for each demand d the corresponding weight ω_d gives the demand's relative need for bandwidth, the weighted MMF allocation is defined as follows:

Definition 4.2. An allocation of resources $\boldsymbol{X}^*$ with weights $\boldsymbol{\omega}$ is *weighted max-min fair* if it is feasible (that is: $\boldsymbol{X}^* \geq 0$ and the link loads generated by $\boldsymbol{\omega X}^*$ do not exceed link capacities), and for each feasible (as above) $\bar{\boldsymbol{X}}$, and for each $d \in D$, if $\frac{X_d^*}{\omega_d} \leq \frac{\bar{X}_d}{\omega_d}$, then for some d', $\frac{X_d^*}{\omega_d} \geq \frac{X_{d'}^*}{\omega_{d'}}$ and $\frac{X_{d'}^*}{\omega_{d'}} > \frac{\bar{X}_{d'}}{\omega_{d'}}$.

From the definition of MMF, it is clear that MMF is a Pareto optimal allocation.

Definition 4.3. A flow allocation is called *Pareto efficient* or *Pareto optimal*, if no further Pareto improvements to it can be made, i.e., when allocation to any individual cannot be made better without making it worse to some other individual [2].

In mathematical algorithms one of the ways to achieve MMF allocation is by means of lexicographical maximization.

4.3.2.1 Lexicographical maximization

Maximization of lexicographically ordered vectors can be used as a means to achieve a MMF allocation. As it is shown in [31; 80], MMF allocation of certain entities is equivalent to finding lexicographically maximal vector of these entities (sorted in non-decreasing order) within the space of all feasible solutions of the considered problem. Commonly known examples of lexicographical order are ordering of words in a dictionary and ordering of names in a telephone directory.

Definition 4.4. Vector $\boldsymbol{a} = (a_1, a_2, \ldots, a_S)$, sorted in the non-decreasing order, is *lexicographically greater* than the sorted vector $\boldsymbol{b} = (b_1, b_2, \ldots, b_S)$, $\boldsymbol{a} \succ \boldsymbol{b}$, if and only if there exists s', $0 \leq s' < S$, such that $a_s = b_s$ for $s = 1, 2, ..., s'$, and $a_{s'+1} > b_{s'+1}$ (it is possible that $a_i < b_i$ for some $i > s' + 1$). If $\boldsymbol{a} \succ \boldsymbol{b}$ or $\boldsymbol{a} = \boldsymbol{b}$ then we denote this by $\boldsymbol{a} \succeq \boldsymbol{b}$.

Suppose $\boldsymbol{X}$ is a solution vector of a problem, and $\bar{X}$ is the set of all (feasible) n-vectors $\boldsymbol{X}$. Consider a vector of real-valued objective functions/criteria:
$\boldsymbol{f}(\boldsymbol{X}) = (f_1(\boldsymbol{X}), f_2(\boldsymbol{X}), \ldots, f_J(\boldsymbol{X}))$, where $f_j : \bar{X} \rightarrow \mathbb{R}^n, j = 1, 2, \ldots, J$.

Definition 4.5. The *lexicographical maximization* problem for given $\bar{X}$ and $\boldsymbol{f}$ is defined as:

$$\text{lex max} \quad \{\boldsymbol{f}(\boldsymbol{X}) : \boldsymbol{X} \in \bar{X}\},$$

and consists in finding a vector $\boldsymbol{X}^* \in \bar{X}$ for which $f(\boldsymbol{X}^*)$ is *lexicographically maximal* over all $\boldsymbol{X} \in \bar{X}$, i.e., for all $\boldsymbol{X} \in \bar{X}$, $\boldsymbol{f}(\boldsymbol{X}^*) \succeq \boldsymbol{f}(\boldsymbol{X})$.

Finding a MMF allocation is equivalent to finding a lexicographically maximal vector among all feasible solution vectors sorted in non-decreasing order. Say that $\hat{\boldsymbol{X}}$ denotes an allocation vector $\boldsymbol{X}$ sorted in non-decreasing order, and $\bar{Y}$ denotes a set of all such feasible vectors $\hat{\boldsymbol{X}}$. Then the MMF allocation results from solving the following problem:

$$\text{lex max} \;\; \{\hat{\boldsymbol{X}} : \hat{\boldsymbol{X}} \in \bar{Y}\}.$$

4.3.3 Proportional fairness

Another important fairness concept is *proportional fairness* (PF) [81]. PF corresponds to a welfare function $W(U_1, U_2, \ldots, U_D) = \sum_d U_d$, where the individual utility functions are $U(X_d) = \log(X_d)$ for all d. Maximizing the welfare function W gives proportionally fair allocation. PF constitutes a trade-off between the two extremes: MMF (maximum fairness) and total TM (maximum throughput). Proportional fairness is defined as follows.

Definition 4.6. Given a set D of demands, to which bandwidth has to be allocated, a vector of bandwidth allocations $\boldsymbol{X}^* = (X_d^* : d \in D)$ is *proportionally fair* if it is feasible (that is: $\boldsymbol{X}^* > 0$ and the link loads generated by $\boldsymbol{X}^*$ do not exceed link capacities), and maximizes $\sum_d \log X_d^*$.

If each demand $d \in D$ has an associated weight w_d which reflects relative bandwidth need, then the *weighted proportional fairness* criterion is defined as follows:

Definition 4.7. Given a set D of demands, to which bandwidth has to be allocated and a vector of weights $\boldsymbol{w} = (w_d : d \in D)$, a vector of bandwidth allocations (rates) $\boldsymbol{X}^* = (X_d^* : d \in D)$ is *weighted proportionally fair* for a weight vector $\boldsymbol{w}$ if it is feasible (that is: $\boldsymbol{X}^* > 0$ and the link loads generated by $\boldsymbol{X}^*$ do not exceed link capacities), and maximizes $\sum_d w_d \log X_d^*$.

Further in the thesis we will mainly use the weighted PF criterion. For brevity, we will, however, refer to both the non-weighted and weighted versions of the criterion by PF.

The PF principle does not allow zero bandwidth to be allocated to any demand, as well as makes it non-profitable to allocate too much bandwidth to any single demand. Both these observations stem from the properties of the logarithmic function. The idea underlying PF was introduced as early as 1854 in the book of Hermann Heinrich Gossen "*The Development of the Laws of Exchange among Men and of the Consequent Rules of Human Action*". Gossen's First Law states that the satisfaction from consumed each additional amount of the same commodity diminishes until satiety is reached [82]. The logarithmic function captures this idea very well.

4.3.4 Linear approximation of log function

If the same utility function is assumed for all demands, i.e., $U(X_d) = \log(X_d)$ for all d, then the PF allocation is implied by using the following objective function in network design problems:

$$\max \ R(\boldsymbol{X}) = \sum_d w_d \log(X_d). \tag{4.1}$$

The presence of the logarithm in the objective function implying PF allocation makes the objective function (and thus the associated problem) non-linear. Non-linear problems are much more difficult to solve, than their LP counterparts. The objective function (4.1) can be linearized using the following piece-wise linear approximation $\Gamma(X_d)$ of the logarithmic function $\log(X_d)$ (see Figure 4.3):

$$\Gamma(X_d) = min\{F_i(X_d) = a_i X_d + b_i : i = 1, 2, \ldots, I\} \ , \tag{4.2}$$

for some I to be specified. The LP approximation makes the problems solvable with standard LP solvers and consists in introducing auxiliary variables f_d, corresponding to X_d, and a set of I constraints,

$$f_d \ \leq \ a_i X_d + b_i \ , \qquad d = 1, 2, \ldots, D \ , \qquad i = 1, 2, \ldots, I \tag{4.3}$$

$$f_d \qquad \text{unconstrained in sign.} \tag{4.4}$$

corresponding to the linear pieces of approximation (4.2), which replace the logarithm of the flow ($\log(X_d)$). Then the objective function (4.1) can be replaced with:

$$\max \ R(\boldsymbol{X}) = \sum_d w_d f_d. \tag{4.5}$$

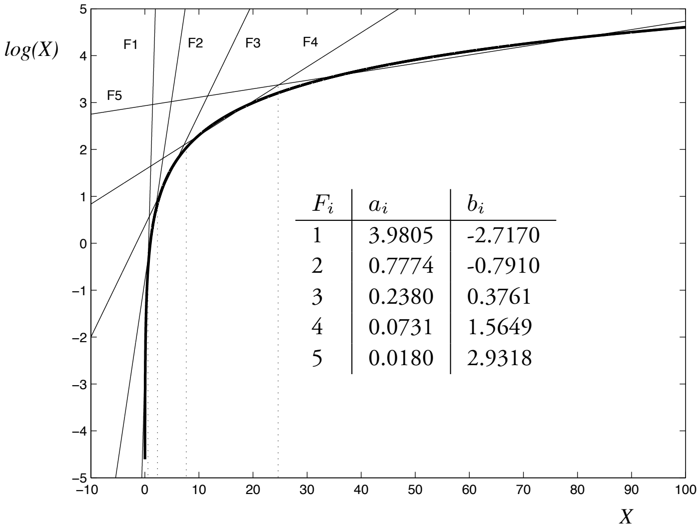
In the numerical experiments, reported in this thesis, the approximation $\Gamma(X_d)$ with five ($I = 5$) linear pieces has been used. The consecutive pairs of coefficients (a_i, b_i) that were used are shown in Figure 4.3.

The presented approximation of the objective function introduces some error to the solution. However, the error can be made as small as desired by using sufficiently many linear pieces for the approximation and assuring that they are sufficiently well spread out in the desired region. Besides, using logarithm or its piece-wise approximation is generally not important in practice.

4.3.5 Bandwidth efficiency of MMF and PF

It has been mentioned that increasing fairness in bandwidth distribution comes at a cost of decreased total network throughput. Let us illustrate the bandwidth efficiency of MMF and PF by means of an example. It will also show how do the MMF and PF bandwidth allocations compare to TM.

Example 4.1. Consider a simple linear network depicted in Figure 4.4, containing three nodes and two links. Capacity of each of the links is 1.5 units. Say that there are three demands in this network, $d = 1, 2, 3$, as shown in the figure. Let x_d be the flow allocated to demand d. Traffic for each of the demands is routed on the single-possible (loop-free) path. Assume that this network carries elastic traffic, and our task is to distribute the available bandwidth (limited only by the link capacities) among the three demands, given that users behind each of the



F_i	a_i	b_i
1	3.9805	-2.7170
2	0.7774	-0.7910
3	0.2380	0.3761
4	0.0731	1.5649
5	0.0180	2.9318

Figure 4.3: Piece-wise linear approximation of the logarithmic function.

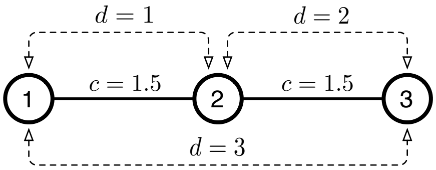


Figure 4.4: A simple network.

demands want to get as much bandwidth as possible and are willing to pay for it. Certainly, there are many possible ways to distribute the bandwidth among the demands. Say, that we want to maximize the total network throughput. This is usually of interest to the network operator, since such a bandwidth assignment maximizes the operator's profit. It can be easily seen that maximum throughput is attained by assigning $x_1 = x_2 = 1.5$ and $x_3 = 0$. Total network throughput in this case is equal to $x_1 + x_2 + x_3 = 3$. However, such an assignment is very unfair towards demand $d = 3$, which is assigned zero bandwidth.

If fairness is the main concern, one can choose to allocate bandwidth according to MMF principle. Recall that the idea of MMF is to allocate at least the same minimum bandwidth to all of the demands, at the same time making this minimum as large as possible. The resulting MMF allocation in this case is $x_1 = x_2 = x_3 = 0.75$ and the total network throughput is $x_1 + x_2 + x_3 = 2.25$.

Thus MMF implies maximum fair solution, since all of the demands are allocated the same minimum amount of bandwidth. However, the total network throughput in this case is degraded, as compared to the throughput maximal solution.

A good tradeoff between the two solutions presented above is provided by distributing bandwidth according to the PF principle. The PF concept implies maximization of the sum of the logarithms of the flows, i.e., $\log x_1 + \log x_2 + \log x_3$. The PF maximal solution in the case of this example is implied by assigning flows $x_1 = x_2 = 1$ and $x_3 = 0.5$. The resulting network throughput is $x_1 + x_2 + x_3 = 2.5$. Thus we can see that PF implies a less fair solution than MMF, but with greater total network throughput. One can also observe, that PF favors shorter flows.

From the above example one can see that bandwidth distribution can be made fair at a price of decreased total network throughput. A trade-off between fairness and throughput can be attained by means of choosing a proper fairness criterion, for example, the PF principle. Table 4.1 summarizes and qualitatively compares typical features of MMF, PF and TM.

Bandwidth sharing objective	Throughput	Fairness
TM	very good	poor
PF	good	good
MMF	satisfactory	very good

Table 4.1: Comparison of TM, MMF and PF.

4.3.6 Other welfare functions

As illustrated by Example 4.1, increasing fairness of bandwidth distribution among the demands comes at a cost of decreased efficiency, i.e., total network throughput. Thus the bandwidth efficiency of different fairness criteria is often compared to the maximal throughput result, achieved by the *throughput maximization* (TM). TM corresponds to the welfare function $W(U_1, U_2, \ldots, U_D) = \sum_d U_d$, where the individual utility functions are $U(X_d) = X_d$ for all d. Throughput-maximal allocation is achieved by maximizing the welfare function, which can be simply written as the following objective function:

$$\max \sum_d X_d. \tag{4.6}$$

Yet another bandwidth sharing principle is called *potential delay minimization* (PDM) [83] and is similar to PF. PDM corresponds to a welfare function $W(U_1, U_2, \ldots, U_D) = \sum_d U_d$, where the individual utility functions are $U(X_d) = -1/X_d$ for all d. The idea behind this principle is that assigning higher transfer rate to a demand makes the potential delay, expressed as $1/X_d$, lower. This is achieved by maximizing the welfare function.

Expressions for MMF, PF, PDM and many other fairness relations can be obtained from a generalized definition of fairness, so called (p, α)*-proportional fairness*, which was introduced in [84].

4.4 Fairness and flow/congestion control

In data networks the notion of fairness is closely related to congestion control. *Congestion* is a situation when due to excessive load on some bottleneck resource its performance deteriorates. The bottleneck resource can be a node or a link (i.e., specific interface on a node to which link is connected). Typical effects that network experiences under congestion are packet loss, delay, and blocking of new connections. Congestion can affect either a part or the whole network. A situation in a network when, due to congestion, little or none of useful information exchange is taking place is called *congestion collapse*. Different mechanisms exist for avoiding congestion, and recover from it once the congestion has occurred. Their taxonomy is presented in [85; 86]. From the perspective of flow/congestion control[3] (CC) mechanisms fairness is important in two ways. First of all, it is very important that both under normal operation, and, especially, under congestion all the flows have a fair share of the bottleneck resource. Secondly, it is very desirable that all sources are cooperative and not-selfish, in a sense that they adjust their rate of sending in order to avoid congestion. In other words, the sources should be *TCP-friendly* [87]. *TCP-firendliness* of a non-TCP flow control protocol means that it implies using the same or similar rate of transmission under certain congestion conditions, as TCP would use under the same conditions. For details on TCP-friendliness refer to [88].

A rough classification of mechanisms that are/can be used for implying fair resource sharing in a network is depicted in Figure 4.5.
Mechanisms that are most widely studied from the fairness perspective, or implementing some fairness principle often are end-to-end congestion control mechanisms and cell or packet scheduling and queue management mechanisms. Congestion control in the Internet is mainly performed end-to-end by the Trans-

[3] Flow control can be seen as a special case of congestion control [86].

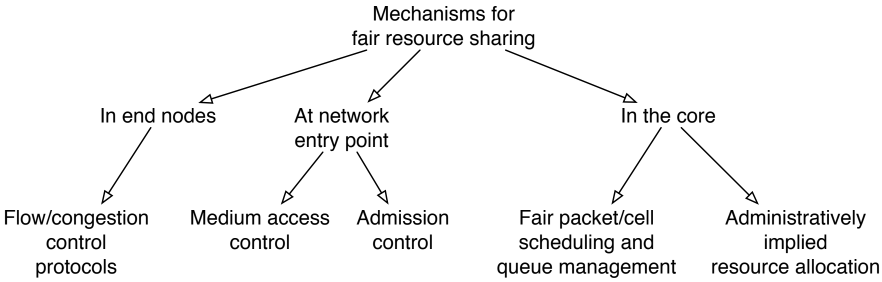


Figure 4.5: Classification of mechanisms for imposing fair resource sharing.

mission Control Protocol. TCP is based on the Additive Increase Multiplicative Decrease (AIMD) algorithm [89]. It has been shown in [75] that congestion control using AIMD mechanism and first-in-first-out (FIFO) queueing with tail dropping in the network nodes under certain circumstances tends to lead to PF bandwidth allocation. This result holds for the Vegas version of TCP, but does not apply for the (today used) Reno version of TCP [90; 91], which instead aims at minimizing the potential delay [2]. Despite this, TCP implies some sort of fair sharing of resources, since all TCP sources that share some bottleneck adjust the rate of sending according to level of congestion at the bottleneck. In this way, every TCP source using the bottleneck gets approximately fair share of the resources. It has been shown in [89] that in the network composed of a single bottleneck link and a number of users sharing it the AIMD algorithm converges to an efficient and fair equilibrium point which corresponds to MMF allocation. However, the approximate fairness implied by using TCP holds globally only if all sources in the network are TCP-friendly, which is not a reality in todays networks. Due to existence of *unfriendly* (i.e., not TCP-friendly) flows, such as those initiated by applications using User Datagram Protocol (UDP), the fairness in a network is destroyed. Misappropriation of resources by high bandwidth, unfriendly flows increases congestion at a bottleneck resource in a network and pushes congestion-reactive (TCP) flows to the state of bandwidth starvation. In order to address this greatly unfair situation, the end-to-end CC mechanisms are not sufficient, and mechanisms in network routers are needed in order to identify and penalize the unfriendly flows by dropping or limiting the rate of their packets. A number of such mechanisms have been proposed in the form of either *active queue management* (AQM) algorithms or *fair queueing scheduling* (FQS) algorithms, implying approximately fair sharing of router's resources by all flows that are active at a router at a given time. AQM mechanisms consist in first estimating the current rate of each flow, to which packets belong, and then dropping packets with probability derived form the rate estimates. A big advantage of AQM mechanisms is that packets that are not dropped are transmitted using a simple first in first out (FIFO) queueing discipline [92]. Examples of AQM mechanisms include *flow random early drop* (FRED) [93] and *core-stateless fair queueing* (CSFQ) [94].

A line of research for fair queueing scheduling algorithms has started with a pioneering work of John Nagle, namely, *fair queueing* (FQ) mechanism [95]. The FQS mechanisms consist in first classifying the packets into multiple queues, where packets of each flow are placed into a separate queue, and then removing packets from each queue according to a schedule which assures a fair share of resources to each of the flows. *Weigthed fair queueing* (WFQ) associates a weight with each of the queues and allows to give priority to certain (classes of) flows. Examples of fair queueing scheduling algorithms are *start-time fair queueing* (SFQ) [96] and *deficit round robin* (DRR) [97], which implements the WFQ discipline. A significant side benefit of the fair queueing mechanisms is that they are able to assure queueing and processing time guarantees for time-critical flows. Experiment results presented in [94] illustrate effectiveness of DRR, FRED and CSFQ to ensure fair flow allocation in different congestion situations with both TCP and UDP flows present. Especially good results were achieved by DRR.

In general, the AQM and FQS mechanisms require that each router keeps the state information for each active flow, thus implying changes in the network equipment. The number of in-progress flows on a backbone link can be counted in hundreds of thousands. Due to hardware requirements and cost the scalability of solutions based on these mechanisms was widely regarded as questionable. It should be mentioned, however, that some mechanisms, like the CSFQ, release the core nodes from the requirement of keeping the status information for each flow, but instead it needs some packet labeling actions to be performed at network boundary nodes, as well as it requires changes in IP header. A recent study [92], taking into account known characteristics of IP traffic on flow level and using data from real backbone link, showed that the FQS and AQM mechanisms are scalable, since the complexity does not increase with the link capacity. This is because the number of active flows that actually have to be scheduled is small, measured at most in hundreds. With the increase of the processing power and memory in the network equipment, the FQS mechanisms have found their way into the routers. For example, WFQ is now implemented in most of the Cisco routers. It is enabled by default on all serial interfaces of Cisco routers working at or below E1 speeds (2.048 Mbps) [98]. Solutions for higher speed links are also available. Queue management and scheduling mechanisms are used as an enabling technology for *differentiated services* (DiffServ) for enforcing resource reservation.

When comparing the end-to-end CC mechanisms with the queue management and scheduling mechanisms in the network routers from the perspective of fairness, the following can be observed. The end-to-end mechanisms result in approximate fairness, while the queue management and scheduling mechanisms allow for more exact fair rate allocation and are able to adapt faster to new network congestion situations [78].

A problem of fair medium access (MAC) arises both in wireless ad-hoc networks and in all-optical WDM networks. In wireless ad-hoc networks a situation can occur when some stations occupy the shared channel while other stations experience bandwidth insufficiency. Fair MAC schemes are developed (for example, see [99]) to address these issues. In multichannel slotted WDM ring networks each node has a tunable-wavelength transmitter and fixed wavelength receiver. Thus there is one wavelength associated with each node. A problem of fair medium access arises when a node grabs too many slots on a wavelength leading to a given destination. Then the nodes which are further downstream have less opportunities to get access to the wavelength. In order to provide fair access to the medium for all nodes, MAC scheme is proposed [100] combining mechanisms assuring fairness, scheduling and resource reservation.

Fair admission control (AC) is often associated with connection-oriented networks, such as ATM. Calls (connection requests) in ATM have different QoS requirements. An AC mechanism determines if a new connection can be admitted to the network by finding out if the QoS requirements of the new connection can be fulfilled at the same time preserving the QoS requirements for the existing connections. The aim is to have an AC mechanism that treats all types of calls fairly, i.e., assures equal probability of admittance for all types of calls. The tradeoff that AC mechanism resolves is between link utilization and fairness of admission for all types of calls. If the fairness requirement is skipped, the tendency is that calls with large bandwidth get hardly admitted. Different mechanisms for AC have been proposed, assuring different levels of fairness and improving utilization. See [101; 102] and literature therein for examples. It is also interesting to note that allocation resulting from the rate control algorithms for the Available Bit Rate service class in ATM is usually argued to be MMF [103; 104].

Other areas where fairness is discussed in the context of admission control are mobile/wireless networks and optical networks. In wireless multimedia networks resources have to be shared among different types of services, such as voice, video and data traffic. Call admission control schemes aim at satisfying QoS requirements for every type of traffic and assuring good network utilization. This often leads to a situation where broadband calls are blocked by narrowband calls. The problem can be alleviated by fair admission control schemes like the one discussed in [105].

In optical WDM networks with traffic grooming capabilities a question of fairness in admission control concerns dynamic wavelength assignment (DWA) algorithms. In such networks connection requests with different capacity requirements arrive at network nodes, where nodes try to assign each request to some wavelength. Some requests get blocked. If no special mechanism is employed for assuring fairness in the admission control, then the connection requests asking for a capacity which is near to the wavelength capacity experience higher blocking. Fairness of existing DWA algorithms is studied in [106], while one of the mechanisms for the fair admission control is presented in [107].

The discussed mechanisms for assuring fair resource allocation in telecommunication networks operate mostly only on the local knowledge, and thus assure only locally fair allocation of resources. Globally fair (network-wide) resource allocation can be achieved by administratively assigning resources to users. This approach requires a planning stage using data from the whole network, which is often carried off-line. The planning tools are based on optimization algorithms. The algorithms presented in this thesis belong to this category. Pre-planned allocations can then be realized in a network with the help of e.g., MPLS LSPs. This approach of resource allocation is suitable only for long-lasting connections, and thus is executed as a part of long-term network planning and optimization activities.

A general discussion of fairness issues and challenges in data networks can be found in [78], while [108] gives a detailed discussion about the MMF fairness principle and its application to telecommunication systems. A good insight in congestion control and fairness issues is given in [87]. It should be noted that research literature discussing fairness in the scope of data networks concentrates on fair distribution of bandwidth among flows, while issues of fair delay management, fair loss rate distribution and fair jitter control have hardly been considered [78].

CHAPTER 5

Fair bandwidth sharing in resilient single-layer networks

5.1 Fairness in a single-layer network at the normal state

Having introduced different fairness principles we will now develop mathematical models and algorithms for designing networks with fair bandwidth sharing among demands. As illustrated in Example 4.1, the PF principle presents a good trade-off between fair allocation and resulting network throughput. Recall, that PF does not allow zero bandwidth to be assigned to any demand, as well as makes it unprofitable to assign too much bandwidth to any single demand. Therefore we will assume that PF is used for bandwidth allocation among demands.

First we will consider only the network at the normal state. Therefore, for the beginning, we will disregard all the failure states, thus assuming $S = 1$, and will focus only on the case when all the the resources in the network are available. After presenting a mathematical model for this case, we will extend it by also taking failure states into account.

We will start by presenting the basic network design problem, which is relevant for a situation when an IP network provider realizes the streams of elastic traffic (i.e., demands) between its customers over IP links. The capacity of the IP links is realized by means of the transmission capacity leased from a transmission carrier (transmission network provider). We may assume that the IP network provider has a fixed (e.g., monthly) budget, B, for leasing the capacity from the carrier, knowing the per month cost, ξ_e, of leasing one unit of capacity on the IP link e. Bandwidth available on the IP network paths is to be shared between the demands according to the PF principle. The objective (which will be called *revenue*) of the IP network operator is to distribute all the available bandwidth among the demands according to the PF principle. The implied IP network design problem involves both allocation of bandwidth to demands, and dimensioning of IP links.

We will assume that for each demand a list of possible candidate paths is given, and it is allowed to split arbitrarily the bandwidth allocated to a demand on any number of its paths. Consequently, the paths to be used for each demand and a portion of demand's flow to be routed on each of them are outcomes of this design problem. The problem is formulated as an IP program and is presented in the next section.

5.1.1 Mathematical model

Let us first consider the case with only one state — the normal state. The Proportional Fairness Nominal Design Task (NDT) with the budget constraint is formulated as follows:

NDT:

$$\max \quad R = \sum_d w_d \log(X_d) \tag{5.1}$$

$$\text{s.t.:} \quad \sum_e \xi_e y_e \ \le B \ , \tag{5.2}$$

$$\sum_p x_{dp} \ = X_d \ , \qquad \forall d \in \mathcal{D} \tag{5.3}$$

$$\sum_d \sum_p \delta_{edp} x_{dp} \ \le y_e \ , \qquad \forall e \in \mathcal{E} \tag{5.4}$$

$$x_{dp}, \ y_e \in \mathbb{Z}^+ \ , \qquad \forall d \in \mathcal{D}, \ \forall p \in \mathcal{P}_d, \ \forall e \in \mathcal{E}. \tag{5.5}$$

This is an uncapacitated network design problem, because we are interested in finding both the flow allocation and link capacities. Constraint (5.2) limits a total cost of capacities installed on all links to a given budget B. Total flow for each demand d is given by constraint (5.3). Variable X_d and constraint (5.3) are auxiliary, and are included in the formulation just for the clarity of presentation. In order to make the formulation more efficient the left-hand side of (5.3) can be directly used in the objective function. The capacity constraint (5.4) forces the consistency between the flow assignments and allocated link capacities. It states that all flows x_{dp} traversing link e must not exceed its capacity. Since capacity in links is installed in modules, constraint (5.4) should be modified as follows to take link modularity into account:

$$\sum_d \sum_d \delta_{edp} x_{dp} \leq y_e M \;, \qquad \forall e \in \mathcal{E} \;, \tag{5.6}$$

where:
$y_e \in \mathbb{Z}^+$, $\forall e \in \mathcal{E}$ number of capacity modules installed on link e;
M size of the link capacity module.

The entity $R = \sum_d w_d \log(X_d)$ in (5.1) is referred to as the (logarithmic) revenue, and should be familiar to the reader by now from Section 4.3.3 as an objective implying PF allocation. As it was discussed in Section 4.3.3 the use of such an objective has two advantages regarding assignment of total flows to demands: (i) it prohibits the assignments of zero flow, and (ii) it does not allow for over-consumption of bandwidth by individual demands.

A relaxation of the above task (assuming variables $x_{dp} \in \mathbb{R}^+$ and $y_e \in \mathbb{R}^+$) has been considered in [31; 109; 110; 111] where it is shown how the explicit solution to NDT can be found by exploiting dual properties of convex programming. In particular it is shown there that

$$X_d^* = \frac{w_d B}{\zeta_d \sum_d w_d} \tag{5.7}$$

where $\zeta_d = \min\{\sum_e \xi_e \delta_{edp} : p = 1, 2, \ldots, m(d)\}$ is the cost (length) of the shortest path realizing demand d. Note, that the lists of candidate paths are not necessary for solving the problem—it is enough to find the shortest path for each demand by some shortest path algorithm and allocate flow to it according to formula (5.7).

The problem can be extended by introducing lower and upper bounds on the total demand flows. This is achieved by adding the following constraint to the problem:

$$h_d \leq X_d \leq H_d \;,\; d \in \mathcal{D} \;, \tag{5.8}$$

where h_d and H_d are the given bounds on total flows. With the help of the bounds we can also express the required fixed bandwidth assignment explicitly. This can be done by setting $h_d = H_d$ in (5.8). By setting the appropriate values for demands' lower and upper bounds one can also combine demands with fixed and with fairly distributed bandwidth assignment within the same model. It should be noted, that the explicit formula (5.7) does not hold when bounds are added to the problem. There is, however, an explicit way of solving the problem with bounds, which is described in [31].

5.2 Path generation for the single-layer network at the normal state

Problem NDT formulated in Section 5.1.1 assumes a set of *candidate paths* (denoted by $\mathscr{P}$) for each demand to be given beforehand. If all possible paths (denoted by $\bar{\mathscr{P}}$) are allowed in the problem, then NDT can be formulated in the *node-link* formulation (which implicitly considers all possible paths). A more realistic scenario is, however, when a set of *allowable paths* (denoted by $\widehat{\mathscr{P}}$) is limited by some requirement, for example by a limit on the number of hops (links) in a path. When using the node-link formulation it is in most cases not possible to effectively take such additional requirements into account[1]. An example of such a situation is the hop-limit criterion. For such cases *link-path* formulations with the candidate path (i.e., routing) lists matching the path criteria are better suited, still, formally, the routing lists for all demands must be specified in advance in the problem formulation. This is not always a practical approach, especially for large networks, since a number of paths, and hence variables, in the problem instance can grow exponentially with the network size (for example the number of all (simple) paths in a network graph grows exponentially with the network size). In such a case a problem formulation in the link-path notation can become *non-compact* which means that the number of variables (columns) of the problem is exponential with the network size. On the other hand, there can be cases when the number of allowable candidate paths does not increase exponentially but polynomially with the network size. Such a nice behavior is for example exhibited by the hop-limit requirement when the maximal number of hops is fixed and does not increase with the increase of the network size.

Certainly, large instances of non-compact formulations, although linear, are virtually impossible to resolve, unless an effective algorithm for generating the paths required in the final optimal solution can be applied while solving the main problem. The technique used for *path generation* (PG) can be based on the general method of column generation [112] (also called *generalized LP* in [32]) in the revised simplex method. The PG approach allows to start solving a network optimization problem with a minimal set of candidate paths (initially it can contain even a single path) for each demand. New, possibly much better, paths are generated and included in routing lists dynamically, as needed. There are several strategies of adding new paths to the problem using PG, namely: 1) *add-all*, when for every demand a path shorter than all paths on a routing list for the demand, is

[1] In general for node-link formulations it holds that $\mathscr{P} = \widehat{\mathscr{P}} = \bar{\mathscr{P}}$, whereas for link-path formulations $\mathscr{P} \subseteq \widehat{\mathscr{P}} \subseteq \bar{\mathscr{P}}$.

added; 2) *add-best*, when a shortest path is added only for one demand, for which the difference in length between the considered path and paths on a demand's routing list is biggest; 3) *add-first*, when only one, first encountered path shorter than all candidate paths for all demands is added. The work in this thesis focuses only on add-all strategy. Comparison of the strategies can be found in [31].

The goal of using PG to resolve a network optimization problem is not only to make the resolution process more efficient, but also to find an ultimate optimal solution defined as follows:

Definition 5.1. A solution of a restricted (with a limited number of predefined candidate paths in $\mathscr{P}$) optimization problem is called *optimal in a wider sense* if it is also the optimal solution for the problem with all allowable paths from $\widehat{\mathscr{P}}$ included [31].

Below we will show how PG can be applied for resolving problem NDT. Although there is no need to carry the path generation procedure for problem NDT, since the explicit solution to the problem exists (see Section 5.1.1), we will anyway develop the PG algorithm for two reasons. First of all, it will serve as an illustrative example which will allow reader to better understand the PG algorithms and their properties for other, more complex, problems presented in this thesis. The second reason is that we will also develop an expression allowing to evaluate what gain one can get by shifting (a part of) a flow to a path which is shorter than the currently used one.

The path generation technique is directly applicable only to linear problems. Therefore we will consider a relaxation of NDT, assuming $x_{dp} \in \mathbb{R}^+$ and $y_e \in \mathbb{R}^+$. Let us restate problem NDT taking all the modifications described in Section 5.1.1, into account, and include the lower and upper bounds on demand volumes. The relaxed problem is denoted by RNDT and is presented below.

RNDT:

$$\max \qquad R = \sum_d w_d \log\left(\sum_p x_{dp}\right) \tag{5.9}$$

$$\text{s.t.:} \qquad \sum_e \sum_d \sum_p \delta_{edp} x_{dp} \xi_e \ \le B \tag{5.10}$$

$$h_d \ \le \ \sum_p x_{dp} \ , \qquad \forall d \in \mathcal{D} \tag{5.11}$$

$$\sum_p x_{dp} \ \le \ H_d \ , \qquad \forall d \in \mathcal{D} \tag{5.12}$$

$$x_{dp} \in \mathbb{R}^+ \qquad \forall d \in \mathcal{D},\ \forall p \in \mathcal{P}_d. \tag{5.13}$$

Note that constraints (5.2) and (5.4) were merged by substituting the left-hand side of (5.4) for variable y_e in (5.2), resulting in constraint (5.10). Also, in the objective function and constraints (5.11) and (5.12), X_d was substituted by the explicit expression given by the left-hand side of (5.3).

In Section 4.3.4 we have discussed how objective function (5.9) can be linearized with the help of a piece-wise linear approximation of the logarithm function in order to convert problem RNDT to a linear problem, and hence to enable PG. The approximation of objective function (5.9) consists in adding the following constraints to problem RNDT

$$f_d \ \le\ a_i\left(\sum_p x_{dp}\right) + b_i\ , \qquad \forall d \in \mathcal{D},\ \forall i \in \mathcal{I} \tag{5.14}$$

$$f_d \in \mathbb{R}\ , \qquad \forall d \in \mathcal{D}, \tag{5.15}$$

and substituting the objective function with

$$\max\ R = \sum_d w_d f_d. \tag{5.16}$$

Thus, we finally arrive at the following linear programming formulation:

LRNDT:

$$\max \qquad R = \sum_d w_d f_d \tag{5.17}$$

$$\text{s.t.:} \tag{5.18}$$

$$[\pi \ge 0] \qquad \sum_e \sum_d \sum_p \delta_{edp} x_{dp} \xi_e \ \le B \tag{5.19}$$

$$[\sigma_d \ge 0] \quad h_d \ \le\ \sum_p x_{dp}\ , \qquad \forall d \in \mathcal{D} \tag{5.20}$$

$$[\tau_d \ge 0] \quad \sum_p x_{dp} \ \le\ H_d\ , \qquad \forall d \in \mathcal{D} \tag{5.21}$$

$$[\beta_{di} \ge 0] \quad f_d \ \le\ a_i \sum_p x_{dp} + b_i\ , \qquad \forall d \in \mathcal{D}, \forall i \in \mathcal{I} \tag{5.22}$$

$$f_d \in \mathbb{R}\ , \qquad \forall d \in \mathcal{D} \tag{5.23}$$

$$x_{dp} \in \mathbb{R}^+\ , \qquad \forall d \in \mathcal{D}, \forall p \in \mathcal{P}_d. \tag{5.24}$$

PG works with dual variables of the original (primal) LP problem LRNDT. Therefore, we will first write the Lagrangian function for LRNDT and examine its dual variables. The dual variables associated with each of the constraints (5.19)-(5.22) are indicated in the brackets in front of each constraint. For convenience we transform the objective function (5.17) into minimization:

$$\min \quad F = -R = -\sum_d w_d f_d. \tag{5.25}$$

The Lagrangian function for LRNDT with the objective transformed to (5.25) takes the form:

$$\begin{aligned} L(\boldsymbol{f}, \boldsymbol{x}\ ;\ \pi, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}) &= \sum_d \Big(\sum_i \beta_{di} - w_d \Big) f_d + \\ + \sum_d \sum_p \Big(\tau_d - \sigma_d - \sum_i a_i \beta_{di} &+ \pi \sum_e \delta_{edp} \xi_e \Big) x_{dp} \\ + \sum_d \Big(h_d \sigma_d - H_d \tau_d - \sum_i b_i \beta_{di} \Big) &- B\pi. \end{aligned} \tag{5.26}$$

By definition, the dual objective function reads:

$$\begin{aligned} W(\pi, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}) &= \min_{\boldsymbol{x} \geq \boldsymbol{0}\ ,\ \boldsymbol{f} \gtreqless \boldsymbol{0}} L(\boldsymbol{f}, \boldsymbol{x}\ ;\ \pi, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}) \\ &= \sum_d \Big(h_d \sigma_d - H_d \tau_d - \sum_i b_i \beta_{di} \Big) - B\pi. \end{aligned} \tag{5.27}$$

Constraining values of the dual variables only to those resulting in a bounded value of the dual function W, the dual problem becomes:

DLRNDT:

$$\max \quad W(\pi, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}) = \sum_d \Big(h_d \sigma_d - H_d \tau_d - \sum_i b_i \beta_{di} \Big) - B\pi \tag{5.28}$$

s.t.:

$$\sum_i \beta_{di} \ = \ w_d\ ,\ \forall d \in \mathcal{D} \tag{5.29}$$

$$\sum_i a_i \beta_{di} + \sigma_d - \tau_d \ \leq \ \pi \sum_e \delta_{edp} \xi_e\ ,\ \forall d \in \mathcal{D},\ \forall p \in \mathcal{P}_d \tag{5.30}$$

$$\pi, \beta_{di}, \sigma_d, \tau_d \in \mathbb{R}^+\ ,\ \forall d \in \mathcal{D},\ \forall i \in \mathcal{I}. \tag{5.31}$$

It is well known [113] that any optimal primal solution ($\boldsymbol{x}^*$,$\boldsymbol{f}^*$) and any optimal dual solution $(\pi^*, \boldsymbol{\beta}^*, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*)$ form a saddle point of the Lagrangian function.

Such a saddle point (i.e, an optimal solution and the corresponding optimal multipliers) can be obtained from an LP simplex-based solver. The complementary slackness property states that the following condition is satisfied for any pair of optimal primal and dual solutions:

$$\left(\sum_i a_i\beta^*_{di} + \sigma^*_d - \tau^*_d - \pi^*\sum_e \delta_{edp}\xi_e\right)x^*_{dp} \;=\; 0, \quad \forall d \in \mathcal{D}, \forall p \in \mathcal{P}_d. \tag{5.32}$$

Thus, if for some path p candidate for demand d and for at least one optimal set of multipliers, the value in the parentheses in (5.32) is strictly negative, then the corresponding flow variable x^*_{dp} must be equal to zero for all optimal primal solutions. This implies that path (d, p) cannot carry a positive flow in *any* optimal primal solution of problem LRNDT with the given candidate paths.

Let $\mathscr{P}_d$ be the given list of candidate paths for demand d and let $\mathcal{L}_d$ be the length/cost of the shortest path on this list:

$$\mathcal{L}_d = \min_{p\in\mathcal{P}_d}\left(\sum_e \delta_{edp}\xi_e\right), \qquad \forall d \in \mathcal{D}. \tag{5.33}$$

where $\mathcal{P}_d$ is the set of indices of paths in set $\mathscr{P}_d$. Certainly, the expression enclosed in parentheses on the right-hand side of (5.33) gives a cost of path p for demand d according to metrics $\boldsymbol{\xi}$, i.e., to the real (primal) link costs.

From (5.29) and complementary slackness it follows that for any saddle point $(\boldsymbol{x}^*, \boldsymbol{f}^*; \pi^*, \boldsymbol{\beta}^*, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*)$ of Lagrangian (5.26) the following conditions hold:

$$\sum_i \beta^*_{di} \;=\; w_d\,, \qquad \forall d \in \mathcal{D} \tag{5.34}$$

$$x^*_{dp} > 0 \Rightarrow \sum_e \delta_{edp}\xi_e = \mathcal{L}_d\,, \qquad \forall d \in \mathcal{D}, \forall p \in \mathcal{P}_d \tag{5.35}$$

$$\sum_p x^*_{dp} > 0 \Rightarrow \sum_i a_i\beta^*_{di} + \sigma^*_d - \tau^*_d \;=\; \pi^*\mathcal{L}_d\,, \quad \forall d \in \mathcal{D}. \tag{5.36}$$

There are several interesting properties associated with optimal multipliers. First of all, for every demand d one of variables σ^*_d and τ^*_d will always be zero, i.e.,

$$\sigma^*_d\tau^*_d = 0\,, \qquad \forall d \in \mathcal{D} \tag{5.37}$$

if only $h_d \neq H_d$. This again follows from the complementary slackness property which implies that σ_d^* or τ_d^* can be non-zero only if the total flow $X_d^* = \sum_p x_{dp}^*$ for demand d is equal to the lower bound h_d, or to the upper bound H_d, respectively. Unless both bounds are equal, one of the variables must always be zero. Furthermore, if $h_d < X_d^* < H_d$ holds for at least one optimal primal solution, i.e., the total flow is between the bounds, then $\sigma_d^* = \tau_d^* = 0$.

The second observation, which also follows from complementary slackness, is that for each demand d at most two variables β_{di}^* can be non-zero (typically only one variable β_{di}^* is non-zero). This is due to the fact that at most only two (adjacent) linear pieces of the logarithm approximation are used, i.e., at most only two constraints (5.22) for some i and $i+1$ can be active (and in most cases only one linear piece will be active). For example, if there is only one active piece (of number i) then $\beta_{di}^* = w_d$, because of (5.34). This case is illustrated in Figure 5.1(a) where for demand d only approximation piece $i = 2$ is active, and therefore $\beta_{d2}^* = w_d$. If, on the other hand, a bandwidth of some demand corresponds to the point where two approximation pieces intersect, then it holds that $\beta_{di}^* + \beta_{d,i+1}^* = w_d$. An example of such a situation is given in Figure 5.1(b), where both pieces $i = 2$ and $i = 3$ are active, implying that $\beta_{d2}^* + \beta_{d3}^* = w_d$.

Thus, if for some demand $\hat{d}$, the inequality $h_{\hat{d}} < X_{\hat{d}}^* < H_{\hat{d}}$ holds for some optimal $X_{\hat{d}}^*$, and only one piece $i(\hat{d})$ of the approximation is active, conditions (5.34) – (5.36) can be combined into the following simple expression:

$$w_{\hat{d}} a_{i(\hat{d})} \;=\; \pi^* \mathcal{L}_{\hat{d}}. \tag{5.38}$$

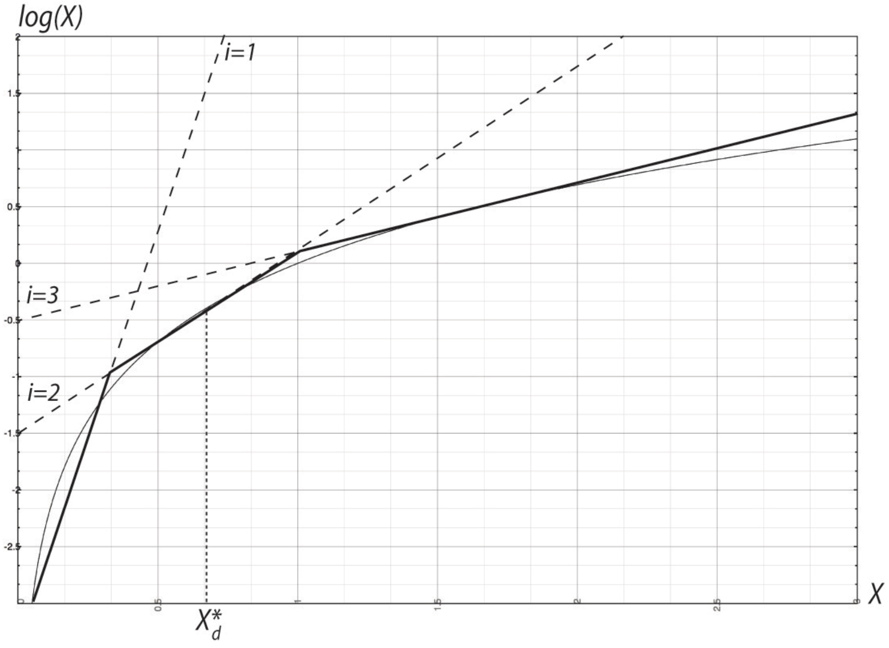
Hence, if we consider a special case with all $h_d = 0$ and all $H_d = +\infty$ and that there exists an optimal primal solution such that for each demand $d \in \mathcal{D}$ there is exactly one active approximation linear piece (piece number $i(d)$, say), and $X_d^* > 0$, then for any set of optimal multipliers it holds that:

$$\frac{w_d a_{i(d)}}{\mathcal{L}_d} \;=\; \pi^* , \qquad\qquad \forall d \in \mathcal{D} \tag{5.39}$$

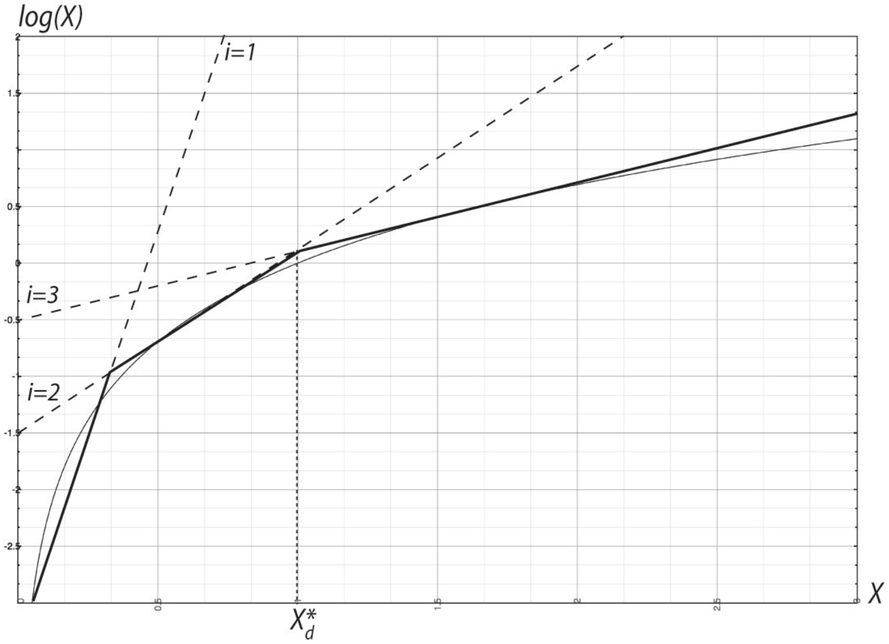
and consequently the optimal value W^* of the dual function (7.15) can be expressed as:

$$W^* = -\sum_d w_d \Big(a_{i(d)} \frac{B}{D\mathcal{L}_d} + b_{i(d)} \Big). \tag{5.40}$$

This expression corresponds to a situation when the total flow $X_d^* = \frac{B}{D\mathcal{L}_d}$ is assigned to one of the shortest paths of demand d and the linear function $a_{i(d)} X_d + b_{i(d)}$ is used for computing its revenue.



(a) One approximation piece is active.



(b) Two approximation pieces are active.

Figure 5.1: Cases of flow approximation.

We will now state and prove a proposition underlying the path generation algorithm. Suppose that for a given instance of problem LRNDT all allowable paths for every demand d form a set $\widehat{\mathscr{P}}_d$. For each demand d, let $\wp_d \in \widehat{\mathscr{P}}_d$ be one of the shortest paths among all allowable paths, and let ℓ_d be its cost with respect to the link cost metrics $\boldsymbol{\xi}$, i.e.,

$$\ell_d = \|\wp_d\|_{\boldsymbol{\xi}} = \sum_{e \in \wp_d} \xi_e = \min_{p \in \widehat{\mathcal{P}}_d} \left(\sum_e \delta_{edp} \xi_e \right) \qquad \forall d \in \mathcal{D}, \qquad (5.41)$$

where $\widehat{\mathcal{P}}_d$ is the set of indices of paths in set $\widehat{\mathscr{P}}_d$. We consider two versions of problem LRNDT:

- *full* problem LRNDT-FULL with the set of candidate paths for each demand d equal to $\widehat{\mathscr{P}}_d$
- *limited* problem LRNDT-LIMITED with the set of candidate paths for each demand d equal to $\mathscr{P}_d$ where $\mathscr{P}_d \subseteq \widehat{\mathscr{P}}_d$ for all d and $\mathscr{P}_d \subsetneq \widehat{\mathscr{P}}_d$ for at least one demand d.

Proposition 5.1. *Let $(\boldsymbol{f}^*, \boldsymbol{x}^* \,;\, \pi^*, \boldsymbol{\beta}^*, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*)$ be a saddle point of the Lagrangian (5.26) for an instance of the limited problem LRNDT-LIMITED with the given set of candidate paths $\mathscr{P}_d$ for each demand d. Let $\wp_d \in \widehat{\mathscr{P}}_d$ be a shortest path for demand d and let ℓ_d denote its length (5.41). Let $\mathcal{L}_d$ denote the length of the shortest path on the list of candidate paths $\mathscr{P}_d$ for demand d of the limited problem LRNDT-LIMITED. Then:*

- *If $\mathcal{L}_d = \ell_d$ for all $d \in \mathcal{D}$, then any optimal solution to LRNDT-LIMITED is also optimal for LRNDT-FULL. Notice that this is the case when problem LRNDT-FULL has no paths shorter than LRNDT-LIMITED.*
- *Otherwise, if $\mathcal{L}_d > \ell_d$ for some demand d, then adding path $\wp_d$ into $\mathscr{P}_d$ can possibly improve the current optimal solution, and the maximal rate of improvement is equal $\pi^*(\mathcal{L}_d - \ell_d)$.*

Proof:
The first part of the proposition follows from the fact that the dual problems to LRNDT-FULL and LRNDT-LIMITED have exactly the same domains given by constraints (5.29) – (5.31). In particular, constraint (5.30) defines the same set of dual variables since $\mathcal{L}_d = \ell_d$ for all $d \in \mathcal{D}$. Hence the two dual problems are identical and thus have the same optimal objective implying that the LRNDT-FULL and LRNDT-LIMITED have the same optimal primal objective function value.

Now let us proceed to the second part of the proposition. Suppose that for some demand $\hat{d}$ there exists a path $\wp_{\hat{d}} \in \widehat{\mathscr{P}}_{\hat{d}} \setminus \mathscr{P}_{\hat{d}}$ with the length equal to $\ell_{\hat{d}}$ (where $\ell_{\hat{d}}$ is calculated as in (5.41), i.e., $\ell_{\hat{d}} = \sum_{e \in \wp_{\hat{d}}} \xi_e$, and denotes the cost of routing one unit flow on path $\wp_{\hat{d}}$) which is strictly less than $\mathcal{L}_{\hat{d}}$ (recall that $\mathcal{L}_{\hat{d}}$ is the length of the shortest path on the candidate list $\mathscr{P}_{\hat{d}}$ of problem LRNDT-LIMITED). Consider a new, perturbed, problem LRNDT-ε obtained for some fixed $\varepsilon > 0$ from the limited problem LRNDT-LIMITED by the following modifications:

$$h^{\varepsilon}_{\hat{d}} = h_{\hat{d}} - \varepsilon \tag{5.42}$$

$$H^{\varepsilon}_{\hat{d}} = H_{\hat{d}} - \varepsilon \tag{5.43}$$

$$B^{\varepsilon} = B - \varepsilon \ell_{\hat{d}}. \tag{5.44}$$

The modifications (5.42)-(5.44) express the situation when for demand $\hat{d}$ a flow of fixed size ε is routed on path $\wp_{\hat{d}}$, and otherwise the demands are realized according to the solution of LRNDT-LIMITED. Consequently, the lower and the upper bounds on the total flow for $X_{\hat{d}}$ are decreased by ε. Besides, the budget is decreased by the cost $\ell_{\hat{d}}$ required to route flow ε on path $\wp_{\hat{d}}$.

When evaluating the primal objective function for the perturbed problem, flow ε has to be taken into account. Essentially, if the original objective (5.9) with the logarithm function is considered, then for the perturbed problem the objective function at optimum becomes:

$$R^* = \sum_d w_d \log\Big(\sum_p x^*_{dp} + \eta_d \varepsilon\Big), \tag{5.45}$$

where $\eta_d = 1$ only for $d = \hat{d}$, and 0 otherwise. In LRNDT-ε we assume piecewise linear approximation of the logarithmic function and therefore we adjust constraint (5.22) for demand $\hat{d}$ to account for flow ε as follows:

$$f_{\hat{d}} \;\leq\; a_i\Big(\sum_p x_{\hat{d}p} + \varepsilon\Big) + b_i\,, \qquad \forall i \in \mathcal{I}. \tag{5.46}$$

The dual function of the perturbed problem LRNDT-ε is obtained analogously to the dual function (5.27) for problem LRNDT-LIMITED taking modifications

(5.42)-(5.44) and (5.46) into account:

$$\begin{aligned}
W^{\varepsilon}(\pi, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}) &= \sum_{d \neq \hat{d}} \left(h_d \sigma_d - H_d \tau_d - \sum_i b_i \beta_{di} \right) + (h_{\hat{d}} - \varepsilon) \sigma_{\hat{d}} \\
&- (H_{\hat{d}} - \varepsilon) \tau_{\hat{d}} - \sum_i b_i \beta_{\hat{d}i} - \sum_i a_i \varepsilon \beta_{\hat{d}i} - (B - \varepsilon \ell_{\hat{d}}) \pi \\
&= \sum_d \left(h_d \sigma_d - H_d \tau_d - \sum_i b_i \beta_{di} \right) - B\pi + \varepsilon \left(\tau_{\hat{d}} - \sigma_{\hat{d}} - \sum_i a_i \beta_{\hat{d}i} + \pi \ell_{\hat{d}} \right) \\
&= W(\pi, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}) + \varepsilon \left(\tau_{\hat{d}} - \sigma_{\hat{d}} - \sum_i a_i \beta_{\hat{d}i} + \pi \ell_{\hat{d}} \right),
\end{aligned} \tag{5.47}$$

where $W(\pi, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau})$ is the dual function of LRNDT-LIMITED. We also note that the dual problems to both LRNDT-ε and LRNDT-LIMITED have the same domain.

Let $(\pi^*, \boldsymbol{\beta}^*, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*)$ be an optimal dual solution of LRNDT-LIMITED. Then, the optimal revenue for LRNDT-LIMITED, R^*, calculated according to (5.17)

$$R^* = -F^* = -W^* = -W(\pi^*, \boldsymbol{\beta}^*, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*), \tag{5.48}$$

where F^* is the value of the optimal primal function (5.25) of LRNDT-LIMITED (and equality $F^* = W^*$ follows from the strong duality theorem, see [32]). Further, let $(W^{\varepsilon})^*$ denote the optimal objective for the problem dual to LRNDT-ε. Then, because multipliers $(\pi^*, \boldsymbol{\beta}^*, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*)$ are in general not optimal for the dual of LRNDT-ε, we have that

$$\begin{aligned}
(R^{\varepsilon})^* &= -(W^{\varepsilon})^* \leq -W^{\varepsilon}(\pi^*, \boldsymbol{\beta}^*, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*) \\
&= -W^{\varepsilon}(\pi^*, \boldsymbol{\beta}^*, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*) - \varepsilon \left(\tau^*_{\hat{d}} - \sigma^*_{\hat{d}} - \sum_i a_i \beta^*_{\hat{d}i} + \pi^* \ell_{\hat{d}} \right) \\
&= R^* + \varepsilon \left(\sum_i a_i \beta^*_{\hat{d}i} + \sigma^*_{\hat{d}} - \tau^*_{\hat{d}} - \pi^* \ell_{\hat{d}} \right),
\end{aligned} \tag{5.49}$$

where $(R^{\varepsilon})^*$ is the optimal revenue for LRNDT-ε calculated according to (5.17). Thus, assuming that (5.36) holds for LRNDT-LIMITED, we finally obtain the following inequalities:

$$(R^{\varepsilon})^* \leq R^* + \varepsilon \pi^* (\mathcal{L}_{\hat{d}} - \ell_{\hat{d}}). \tag{5.50}$$

This completes the proof. □

Inequality (5.50) shows what is the possible maximal revenue gain. This maximum is attained when the optimal dual variables (π^*, $\boldsymbol{\beta}^*$, $\boldsymbol{\sigma}^*$, $\boldsymbol{\tau}^*$) for the dual to LRNDT-LIMITED remain optimal for the dual to LRNDT-ε, i.e., when the simplex basis does not change from problem LRNDT-LIMITED to problem LRNDT-ε: in such a case including the new path $\wp_{\hat{d}}$ to the list of the candidate paths for demand $\hat{d}$ and assigning flow ε to it will improve the primal objective function by $\varepsilon\pi^*(\mathcal{L}_{\hat{d}} - \ell_{\hat{d}})$.

In fact, this is the case for example when only one piece of the approximation is active for the total flow of demand $\hat{d}$ (the case illustrated in Figure 5.1(a)), and the demand volume is between the bounds. For this case the expression for π^* is given by (5.38), and in this case the gain in revenue can be expressed as:

$$(R^\varepsilon)^* - R^* = \varepsilon w_{\hat{d}} a_{i(\hat{d})} \frac{\mathcal{L}_{\hat{d}} - \ell_{\hat{d}}}{\mathcal{L}_{\hat{d}}}. \tag{5.51}$$

Path generation procedure for LRNDT-LIMITED is very simple and consists in putting, for each demand, one the shortest path computed according to real link costs $\boldsymbol{\xi}$ on the candidate path list.

5.3 Fairness in a resilient single-layer network

The fair network design model presented in the previous sections, as well as models considered in literature [109; 110], referred to as the normal case fair designs, do not take resilience into account. The network design problem that we will present now extends the NDT problem, and explicitly considers the predefined set of failures at the network design stage. Combining fair bandwidth allocation and robustness is an original contribution of this work.

The problem, as in the case of NDT, consists in finding the optimal PF allocation of bandwidth to demands, however, in this case–also in each of the considered network operation states. Resulting link capacities must be fixed to the same values in all the states (they are not state-dependent), except that they effectively are diminished by failures. Thus the resulting link capacities should be such that they imply maximum possible revenue in each of the operation states. This requirement makes the problem of the *multi-criteria optimization* type, because revenue functions should be maximized simultaneously for all the states. Besides, in order to be able to compare the resulting revenue vectors and selecting the best one in some preferred sense, a global objective function must be decided. It is by far not obvious how to formulate an adequate corresponding optimization problem — this can most likely be done in several reasonable ways. We start by presenting one such formulation using the rule of Max-Min Fairness applied to the revenue values for all the considered operation states. Later we introduce other combinations of criteria and compare them by numerical study.

5.3.1 Combining MMF and PF

The problem employs a kind of two-dimensional fairness: bandwidth between demands within each state is allocated according to PF principle, whereas revenues among all states are allocated according to the MMF principle. Advantages of combining the two fairness principles are several. First of all, none of the flows may be assigned zero bandwidth (due to the log function implying PF allocation) in any of the failure states, no matter how severe the failure is (assuming that the design problem is still feasible). However, the volume of bandwidth assigned to each demand in each of the states may vary (within eventual bounds, if they are specified). Secondly, as a result of imposing MMF allocation, the revenues in each situation attain at least the same minimum value, which is maximized. And since the revenues reflect total network bandwidth in a given situation, the higher the revenue, the higher is the bandwidth that is available for demands to share. This means that in each situation there will be at least the same minimum bandwidth available, and bandwidth allocations will not be discriminated in any state just to have higher bandwidth allocations in some other state. This is, of course, assuming that all demand weights (w_{ds}) are the same, and the not weighted MMF is used. Changing the weights allow to give priorities when allocating bandwidth to some demands in some situations.

5.3.2 Recovery mechanism

The problem assumes full reconfiguration of flows as a mean of recovery from failures. Thus, in a case of a failure, even the unaffected flows may be torn down and then all the flows are rerouted in order to recover the affected by failure flow within the surviving network resources. Although this is not really feasible mean of recovery in practice, it can be seen as a relaxation of different practical recovery mechanisms, and can be used as such for obtaining the upper bound for the solution. Besides, our intention here is to develop models and algorithms for fair flow allocation in resilient networks, and not the recovery mechanisms themselves.

5.3.3 Mathematical model

Suppose $\bar{X}$ is the set of all (feasible) vectors $\boldsymbol{X} = \{X_{ds} : d = 1, 2, \ldots, D, s = 1, 2, \ldots, S\}$ defined by the following constraints:

$$\sum_e \xi_e y_e \ \leq \ B\,, \tag{5.52}$$

$$\sum_p x_{dps} \ = \ X_{ds}\,, \qquad \forall d \in \mathcal{D}\,, s \in \mathcal{S} \tag{5.53}$$

$$\sum_d \sum_p \delta_{edp} x_{dps} \ \leq \ \alpha_{es} y_e\,, \quad \forall e \in \mathcal{E}\,, s \in \mathcal{S} \tag{5.54}$$

$$x_{dps},\ y_e \in \mathbb{Z}^+\,, \qquad \forall d \in \mathcal{D},\ p \in \mathcal{P}_d,\ s \in \mathcal{S},\ e \in \mathcal{E}. \tag{5.55}$$

We will call a problem with a set of feasible vectors $\bar{X}$ defined as above by Robust Design Task (RDT). The constraints for the RDT are similar to those of problem NDT, presented in Section 5.1.1. The only thing which is different is that flow variables (x_{dps}) and associated constraints are now also indexed on situations. The right hand side of constraint (5.54) gives the capacity available on link e in failure situation s, which is defined as $\alpha_{es} y_e$. Thus the constraint imposes that the total bandwidth for flows traversing link e in situation s does not exceed the available capacity of the link in this situation.

For each $\boldsymbol{X} \in \bar{X}$ let $\boldsymbol{R}(\boldsymbol{X}) = (R_1(\boldsymbol{X}), R_2(\boldsymbol{X}), \ldots, R_S(\boldsymbol{X})$ denote the vector of revenues,

$$R_s(\boldsymbol{X}) \ = \ \sum_d w_{ds} \log(X_{ds})\ ,\ \forall s \in \mathcal{S}\,, \tag{5.56}$$

sorted in non-decreasing order. Then the Proportional Fairness RDT is formulated as follows [114]:

$$\textbf{RDT-MMF-PF:} \qquad \text{lex max}\ \{\boldsymbol{R}(\boldsymbol{X}) : \boldsymbol{X} \in \bar{X}\}. \tag{5.57}$$

It is denoted RDT-MMF-PF in order to reflect which criteria are used in the objective function.

5.3.4 Extensions of the basic problem

Problem RDT-MMF-PF posed in Section 5.3.3 can be extended in several ways. Such variations of the problem can be achieved by introducing additional constraints characterizing the solution set $\bar{X}$. First, we may assume that the capacities of links, y_e, are limited (e.g., because of the shortage of transmission capacity), adding constraints

$$y_e \leq C_e \ , \ e = 1, 2, ..., E \ , \tag{5.58}$$

where C_e are fixed upper bounds on link capacities. Secondly, we can introduce lower and upper bounds on the total demand flows realized in failure situations;

$$h_{ds} \leq X_{ds} \leq H_{ds} \ , \ d = 1, 2, ..., D \ , \ s = 1, 2, .., S \ , \tag{5.59}$$

where h_{ds} and H_{ds} are the given bounds on total flows. With the help of the bounds we can also to specify the required fixed bandwidth assignment explicitly. This can be done by setting $h_{ds} = H_{ds}$ in (5.59). By setting the appropriate values for demands' lower and upper bounds one can also combine demands with fixed and fairly distributed bandwidth assignment within the same model. Finally, we can impose constraints on the flow reconfiguration in the case of failure. So far we have assumed that individual flows, x_{dps}, are only situation-dependent and can be freely reconfigured using surviving resources. This allows us to not explicitly distinguish the nominal situation in the formulation of RDT-MMF-PF. If we wish to consider a more realistic case when the nominal flows which are not affected by a failure (i.e., the nominal flows that survive a given failure situation s) are not disconnected (and hence still work in situation s) we proceed as follows. We label the "true" failure situations with $s = 1, 2, ..., S$, and use the index $s = 0$ for the normal state. Then we assume that if a link fails then it fails totally (i.e., $\alpha_{es} \in \{0, 1\}$), and introduce the binary path availability coefficients:

$$\theta_{dps} = \prod_{\{e:\delta_{edp}=1\}} \alpha_{es} \ , \qquad \forall d \in \mathcal{D}, \ \forall p \in \mathcal{P}_d, \ s = 1, 2, ..., S. \tag{5.60}$$

Now, the considered requirement for flow reallocation can be taken into account by introducing new constraints:

$$x_{dps} \geq \theta_{dps} x_{dp0} \ , \qquad \forall d \in \mathcal{D}, \ \forall p \in \mathcal{P}_d, \ s = 1, 2, ..., S. \tag{5.61}$$

Since in the backbone optical networks link capacities are installed in modules, problem formulations can be adjusted to account for the modularity by modifying constraint (5.54) as follows:

$$\sum_d \sum_d \delta_{edp} x_{dps} \ \leq \ \alpha_{gs} y_e M \ , \qquad \forall e \in \mathcal{E} \ , \forall s \in \mathcal{S} \ , \tag{5.62}$$

where:
$y_e \in \mathbb{Z}^+$ number of capacity modules installed on link y;
M size of the link capacity module.

Other ways of taking similar restrictions in flow reallocation (e.g., the use of single-backup paths) and the corresponding constraints are described in Section 9.3 of [31].

It should be noted that all the above described modifications of RDT-MMF-PF allow for the application of the algorithms presented in Section 5.3.5 without any adjustments. In fact, the algorithms can be used to solve the problems with other criteria for bandwidth distribution among demands, instead of PF. Thus, they, in fact, are frameworks for resolving the MMF problems.

5.3.5 The algorithms

The robust design task formulated above is not trivial to solve since it involves lexicographical order maximization (see Section 4.3.2.1 and [115; 116]). Below we describe an iterative algorithm for resolving a relaxed problem RDT-MMF-PF (i.e., when $x_{dps}, y_e \in \mathbb{R}^+$) using the idea of [117] and then show how, using dual theory, we can improve its efficiency. This further investigation will result in an improved algorithm, which is presented later.

5.3.5.1 Basic algorithm

The structure of the basic algorithm is illustrated by the block diagram shown in Figure 5.2. As it will become clear, the basic and the improved algorithms will mainly differ only by the *non-blocking test* performed in Step 2.

Basic Algorithm for resolving RDT-MMF-PF

Step 0 (Initialization):
Put $n := 0$, $Z_0 := \emptyset$, $Z_1 := \{1, 2, \ldots, S\}$.
Step 1:
Solve the following convex programme:
BS1:

$$\textbf{maximize} \quad t \tag{5.63}$$

$$\textbf{subject to} \quad (5.52) - (5.54) \text{ and}$$

$$R_s = \sum_d w_{ds} \log(X_{ds}) \geq t \; , \; s = 1, 2, \ldots, S \; . \tag{5.64}$$

Let $(t^* \; , \; \boldsymbol{x}^* \; , \; \boldsymbol{X}^* \; , \; \boldsymbol{R}^* \; , \; \boldsymbol{y}^*)$ be the optimal solution to the problem above. Put $t_s := t^*$ for each $s \in Z_1$.

Step 2:
Put $n := n + 1$ and

$$\begin{aligned} Y &:= \{ s \in Z_1 : R_s = t_s \} \quad \text{and} \\ Z_1 &:= \{ s \in Z_1 : R_s > t_s \} . \end{aligned} \tag{5.65}$$

For each[2] $s' \in Y$ solve the following convex programme:

BS2:

$$\textbf{maximize} \quad R_{s'} = \sum_d w_{ds'} \log(X_{ds'}) \tag{5.66}$$

$$\textbf{subject to} \quad (5.52) - (5.54) \text{ and}$$

$$R_s = \sum_d w_{ds} \log(X_{ds}) \geq t_s \ , \ s = 1, 2, \ldots, S \ . \tag{5.67}$$

Let $R_{s'}$ be the solution to the task for s'. Redefine the sets:

$$\begin{aligned} Z_0 &:= Z_0 \cup \{ s' : R_{s'} = t_{s'} \} \ , \\ Z_1 &:= Z_1 \cup \{ s' : R_{s'} > t_{s'} \} \ . \end{aligned} \tag{5.68}$$

Step 3:
If $Z_1 = \emptyset$ then stop; otherwise solve the following convex programme:

BS3:

$$\textbf{maximize} \quad t \tag{5.69}$$

$$\textbf{subject to} \quad (5.52) - (5.54) \text{ and}$$

$$R_s = \sum_d w_{ds} \log(X_{ds}) \geq t_s \ , \ s \in Z_0 \ , \tag{5.70}$$

$$R_s = \sum_d w_{ds} \log(X_{ds}) \geq t \ \ , \ s \in Z_1 \ . \tag{5.71}$$

Let $(t^* \ , \ \boldsymbol{x}^* \ , \ \boldsymbol{X}^* \ , \ \boldsymbol{R}^* \ , \ \boldsymbol{y}^*)$ be the optimal solution to the problem above. Put $t_s := t^*$ for each $s \in Z_1$. Go to Step 2.

5.3.5.2 Comments to the Basic Algorithm

In the algorithm above $t \in \mathbb{R}$ and $t_s \in \mathbb{R}, s \in \{1, 2, \ldots, S\}$ are auxiliary variables. Sets $Y, Z_0, Z_1 \subseteq \{1, 2, \ldots, S\}$ are used to keep situation indices.

[2] Of course, if maximization of $R_{s'}$, $s' \in Y$, indicates that $R_{s''} > t_{s''}$, $s' \neq s'' \in Y$, there is no need to carry out this task for s''.

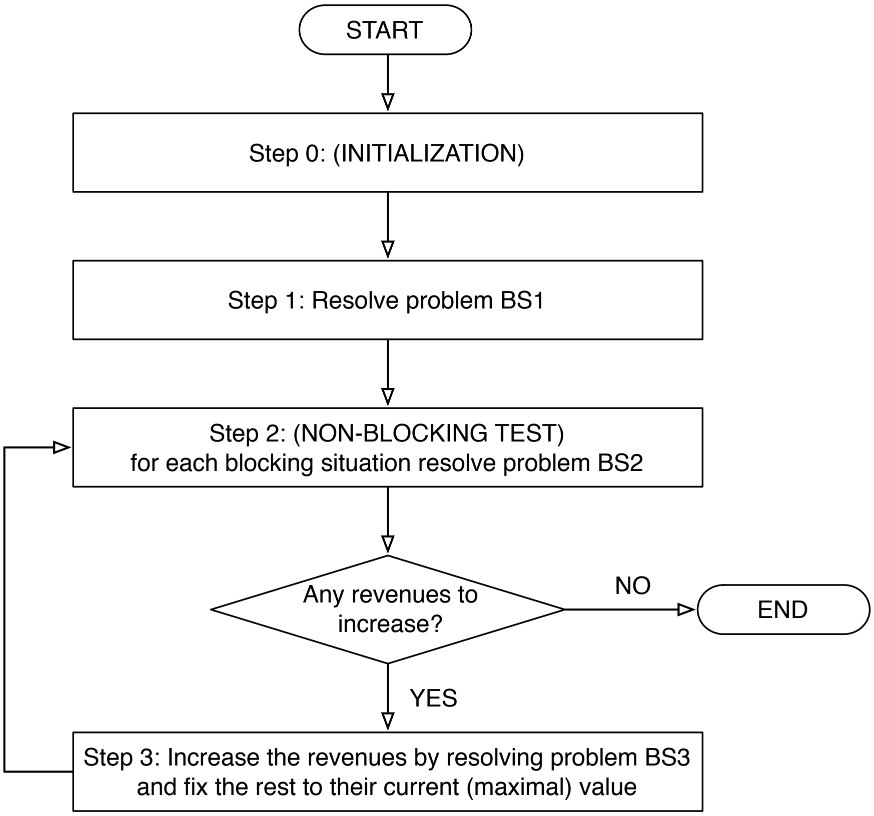
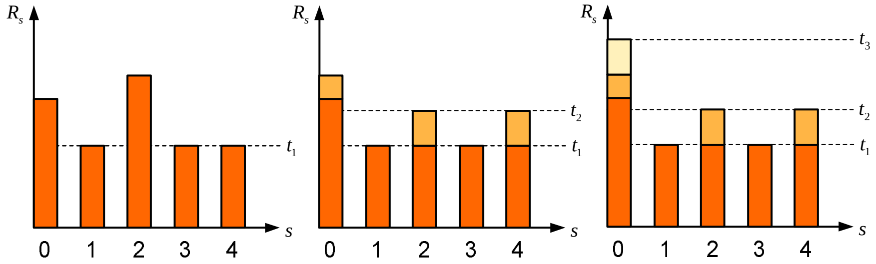


Figure 5.2: The block diagram for the basic algorithm.

The test performed in Step 2 of the algorithm, which is aimed at identifying the situations for which the revenue can be improved, will be called the *non-blocking test*. The set Y is a temporary set which, during execution of Step 2, is used to store situations for which it is not known whether or not the current bound t_s is the upper bound for R_s. The sets Z_1 and Z_0 have the following interpretation after completion of Step 2 in iteration n ($n = 1, 2, \ldots, N$, where N is the number of iterations performed until the termination of the algorithm):

Z_0 : The set of situations for which it is known that R_s is equal to t_s and can not be made any larger.

Z_1 : The current set of situations for which it is known that R_s can be made larger than the current bound t_s.



(a) After iteration $n = 1$. (b) After iteration $n = 2$. (c) After iteration $n = 3$.

Figure 5.3: Evolution of revenues. The revenue vector is not sorted.

The algorithm presented above finds an assignment of link capacities and situation-dependent flows that are max-min fair from the R_s values viewpoint. So it is in fact solving an MMF problem for the PF networks. This means that any solution of the algorithm has the property that the resulting vector $\boldsymbol{R}(\boldsymbol{X})$ is the (unique) lexicographically maximal vector in the space $\left\{ \boldsymbol{R}(\boldsymbol{X}) : \boldsymbol{X} \in \bar{X} \right\}$. The following example illustrates how the allocation of revenues evolve through the iterations of the algorithm, to arrive at the MMF solution.

Example 5.1. Assume we design a network for which there are five states of operation (denoted $s = 0, 1, 2, \ldots, 4$) identified. We want to allocate revenues R_s among these states according to the MMF principle. For this we are going to use the basic algorithm presented in Section 5.3.5.1. Say, in the first iteration $n = 1$ an allocation is found with revenue values as shown in Figure 5.3(a). Assume that the non-blocking test in Step 2 of the algorithm indicates that situations 1 and 3 are blocking. Thus the revenues for these situations are fixed at value t_1 and are increased for the other situations, at least till the value of t_2, which is blocking for some of the remaining situations. This is illustrated in Figure 5.3(b). Say, that after second visit to Step 2 (iteration $n = 2$) it appears that situations 2 and 4 are blocking. Thus in Step 3 their revenue values are fixed at level t_2, and the revenue for the remaining situation 0 is increased yet further, till value t_3, as shown in Figure 5.3(c). After iteration $n = 3$ it appears that value t_3 is blocking for situation $s = 0$, thus the revenue is fixed at the value t_3 and the algorithm terminates.

5.3.5.3 Improving the algorithm efficiency

From the time efficiency viewpoint the main drawback of the algorithm presented in Subsection 5.3.5.1 is the multiple solving of the optimization tasks in Step 2 during each iteration in order to detect whether or not the revenue in a particular situation from set Y can be further increased. This, however, can be checked in a much simpler way provided that the optimal values of the dual variables (Lagrange multipliers) corresponding to constraints (5.64) in Step 1 and to constraints (5.71) in Step 3 are known. We shall illustrate this approach, described in detail in [31], for the optimization task of Step 1.

Let U be the set of all feasible vectors $\boldsymbol{u} = (\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{y})$ defined by constraints (5.52)-(5.55). Then it can be shown, that the dual function (cf. [113], Section 8.3) of the task of Step 1 is equal to

$$W(\boldsymbol{\lambda}) = \max_{\boldsymbol{u} \in U} \{ \ \textstyle\sum_s \lambda_s R_s(\boldsymbol{u}) \ \} \ , \ \boldsymbol{\lambda} \geq \mathbf{0} \ , \tag{5.72}$$

where $R_s(\boldsymbol{u}) = \sum_d w_{ds} \log(X_{ds})$ and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_S)$ is a vector of the dual variables corresponding to constraints (5.64) rewritten as

$$t - \sum_d w_{ds} \log(X_{ds}) \leq 0 \ , \ s = 1, 2, \ldots, S \ . \tag{5.73}$$

Let $\boldsymbol{\lambda}^*$ denote a vector of optimal multipliers, i.e. the multipliers with the property

$$W(\boldsymbol{\lambda}^*) = \min_{\boldsymbol{\lambda} \geq \mathbf{0}} W(\boldsymbol{\lambda}) \ . \tag{5.74}$$

Let t^* denote the optimal value of the primal function (5.63) and let U^* denote the set of all optimal solutions of the considered task. Finally, let $\mathcal{S}$ denote the set of the *blocking situations with respect to* t^*.

Definition 5.2. A situation is called *blocking with respect to* t^* if the following holds:

$$s \in \mathcal{S} \text{ if and only if } R_s(\boldsymbol{u}) = t^* \text{ for all } \boldsymbol{u} \in U^* \ . \tag{5.75}$$

The following fact forms the basis for improving the basic algorithm (through improving the *non-blocking test* (5.66)-(5.67)) from Section 5.3.5.1.

Proposition 5.2. *For each situation* s *(*$s = 1, 2, \ldots, S$*), if* $\lambda_s^* > 0$ *then* $s \in \mathcal{S}$.

Proof: Although the implication in Proposition 5.2 is a direct consequence of the complementary slackness property of convex programming, we will prove it here from scratch, as this can be instructive for a reader not familiar with the dual optimization theory.

Let us dualize constraints (5.73), and form the Lagrangian,

$$\begin{aligned} L(\boldsymbol{u};\boldsymbol{\lambda}) &= t - \sum_s \lambda_s \left(\ t - R_s(\boldsymbol{u}) \ \right) \\ &= t - \sum_s \lambda_s \left(\ t - \textstyle\sum_d w_{ds} \log(X_{ds}) \ \right) \\ &= t \left(\ 1 - \textstyle\sum_s \lambda_s \ \right) + \sum_s \lambda_s \sum_d w_{ds} \log(X_{ds}) \ , \end{aligned} \tag{5.76}$$

and the dual function,

$$W(\boldsymbol{\lambda}) = \max_{\boldsymbol{u} \in U} L(\boldsymbol{u};\boldsymbol{\lambda}) \ , \ \lambda_s \geq 0 \ , \ s = 1, \ldots, S \ , \tag{5.77}$$

cf. [113], Chapter 8. Since we are considering a convex optimization problem, any vector of optimal dual variables (multipliers) $\boldsymbol{\lambda}^*$, satisfies the following equality

$$W(\boldsymbol{\lambda}^*) = \min_{\boldsymbol{\lambda} \geq \mathbf{0}} W(\boldsymbol{\lambda}) = t^* \ . \tag{5.78}$$

It follows that the set Λ in which the dual function assumes its minimum can be limited to

$$\Lambda = \left\{ \ \boldsymbol{\lambda} : \textstyle\sum_s \lambda_s = 1 \ , \ \lambda_s \geq 0 \ , \ s = 1, \ldots, S \ \right\} \ , \tag{5.79}$$

because if $\sum_s \lambda_s \neq 1$ then $W(\boldsymbol{\lambda})$ may be made infinitely large by assuming $t = +\infty$ or $t = -\infty$. Hence, as we are interested in the minimization of the dual function, we may restrict the domain of (5.77) to

$$W(\boldsymbol{\lambda}) = \max_{\boldsymbol{u} \in U} \sum_s \lambda_s R_s(\boldsymbol{u}) \ , \ \ \boldsymbol{\lambda} \in \Lambda \ . \tag{5.80}$$

Now let $\boldsymbol{\lambda}^*$ be a vector of optimal multipliers and suppose that for some $\boldsymbol{u}^* \in U^*$, $(\boldsymbol{u}^*, \boldsymbol{\lambda}^*)$ is a saddle point of the Lagrangian function (5.76). Then

$$t^* \ = \ W(\boldsymbol{\lambda}^*) = L(\boldsymbol{u}^*;\boldsymbol{\lambda}^*) = \max_{\boldsymbol{u} \in U} L(\boldsymbol{u};\boldsymbol{\lambda}^*) = \max_{\boldsymbol{u} \in U^*} L(\boldsymbol{u};\boldsymbol{\lambda}^*) \ , \tag{5.81}$$

so that

$$\sum_s \lambda_s^* R_s(\boldsymbol{u}^*) = t^* \ . \tag{5.82}$$

Since $\sum \lambda_s = 1$, equality (5.82) implies that $R_s(\boldsymbol{u}^*) = t^*$ for all s such that $\lambda_s^* > 0$. Otherwise the sharp inequality $\sum_s \lambda_s^* R_s(\boldsymbol{u}^*) > t^*$ would hold, since equalities $R_s(\boldsymbol{u}^*) \geq t^*$ hold for any optimal solution. This in fact implies that

λ_s^* cannot be positive if s is a non-blocking situation. Indeed, let $\hat{\boldsymbol{u}} \in U^*$ be an optimal solution with the following property

$$\begin{aligned} R_s(\hat{\boldsymbol{u}}) &= t^* \text{ for all } s \in \mathcal{S} \text{ and} \\ R_s(\hat{\boldsymbol{u}}) &> t^* \text{ for all } s \notin \mathcal{S}. \end{aligned} \tag{5.83}$$

Then it is clear that if $\boldsymbol{\lambda}^* \in \Lambda$ and there exists a situation $s \notin \mathcal{S}$, such that $\lambda_s^* > 0$ then

$$\sum_s \lambda_s^* R_s(\hat{\boldsymbol{u}}) > t^*, \tag{5.84}$$

which is a contradiction. Therefore,

$$\text{if } s \notin \mathcal{S} \text{ then } \lambda_s^* = 0, \tag{5.85}$$

and this completes the proof. □

It is important to observe that the inverse implication is not always true, i.e., in general $\lambda_s^* = 0$ does not imply that $s \notin \mathcal{S}$ (see Example 8.3 in [31]). However, the implication in Proposition 5.2 makes it possible to construct an improved version of the basic algorithm. The improved algorithm for solving RDT-MMF-PF is based on a dual non-blocking test and is as presented in the following section.

5.3.5.4 The improved algorithm

The block diagram for the improved algorithm is shown in Figure 5.4. Compared to the basic algorithm for resolving problem RDT, the modified algorithm increases effectiveness of the problem resolution in two ways. The most important advancement is the efficient non-blocking test. In the modified algorithm the test consists in simple checking for the values of dual variables, whereas in the basic algorithm it required the time-consuming solution of the optimization problem for each of the situations. Furthermore, the algorithm efficiency is increased by having the optimization problem resolved only in Step 1, as opposed to Steps 1 and 3 in the basic algorithm. However, this change in the algorithm structure has only a small impact on the total algorithm efficiency, as compare to the gains due to the more efficient non-blocking test. The basic algorithm could also be formed in a similar way.

Improved Algorithm for resolving RDT-MMF-PF

Step 0 (Initialization):
Put $n := 0$, $Z_0 := \emptyset$, $Z_1 := \{1, 2, \ldots, S\}$, $t_s := 0$ for all s.

Step 1:
Solve the following convex programme (denote it by **MS1**):

maximize t

subject to (5.52)-(5.54) and (5.73) and

$$R_s = \sum_d w_{ds} \log(X_{ds}) = t_s \ , \ s \in Z_0 \tag{5.86}$$

$$R_s = \sum_d w_{ds} \log(X_{ds}) \geq t \ \ , \ s \in Z_1. \tag{5.87}$$

Let t^* be the optimal solution of the above problem and $\lambda_s^*, s \in Z_1$ be the optimal dual variables corresponding to constraints (5.87).

Step 2:
Put $n := n+1$ and $t_s := t^*$ for each $s \in Z_1$. Put $Z_0 := Z_0 \cup \{s \in Z_1 : \lambda_s^* > 0\}$ and $Z_1 := \{s \in Z_1 : \lambda_s^* = 0\}$.

If $Z_1 = \emptyset$ then STOP (the vector $\boldsymbol{R} = (R_1, R_2, \ldots, R_S) = (t_1, t_2, \ldots, t_S)$, sorted in non-decreasing order, is the solution of the problem); else go to Step 1.

5.3.5.5 Comments to the improved algorithm

The entities in the improved algorithm have the same meaning as those in the basic algorithm, except of the set Z_1. In the improved algorithm set Z_1 denotes the current set of situations, for which it is not known if t_s is the maximal value for R_s.

If optimal t^* in Step 1 happens to be strictly greater than the optimal solution obtained in the previous iteration, then all situations s in set Z_1 are non-blocking. If not, then one or more situations in set Z_1 are blocking (they are “false" non-blocking situations) and t^* cannot be increased. Still, among the newly obtained optimal dual variables λ_s^* for $s \in Z_1$, at least one with $\lambda_s^* > 0$ will appear, and hence, at least one new blocking situation will be detected. This follows from the fact that in the optimal dual solution of the problem solved in Step 1, the dual variables, λ_s^*, corresponding to (5.87), expressed as in (5.73), have the property analogous to the property defined in (5.79), i.e.,

$$\sum_{s \in Z_1} \lambda_s^* = 1 \text{ and } \lambda_s^* \geq 0 \text{ for } s \in Z_1. \tag{5.88}$$

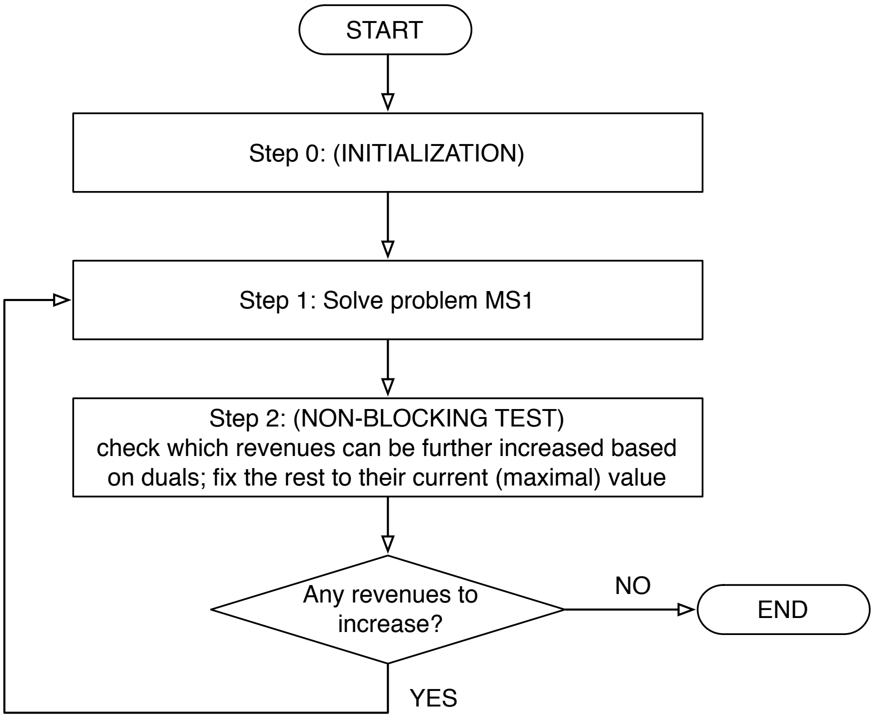


Figure 5.4: The block diagram for the improved algorithm.

Let us notice that although it would be rather difficult to find optimal multipliers for the above considered optimization tasks, the linear approximation presented in Section 4.3.4 makes this feasible, since most of the linear programming solvers provide the optimal dual variables along with primal optimal solution.

Now we will investigate what is really required for the implication $\lambda_s^* = 0 \Rightarrow s \notin \mathcal{S}$ to hold. Suppose that the network has the following (natural) property:

Property 5.3. *If any proper subset $\mathcal{S}_0$ of $\mathcal{S}$ is removed from the formulation of the considered optimization problem, then in the resulting reduced optimization problem at least one of the situations $s \in \mathcal{S} \backslash \mathcal{S}_0$ will become non-blocking with respect to t^*.*

Proposition 5.4. *Property 5.3 holds if, and only if, for each situation s it is true that $\lambda_s^* = 0 \Rightarrow s \notin \mathcal{S}$.*

Proof: We will first show that Property 5.3 implies that $\lambda_s^* = 0 \Rightarrow s \notin \mathcal{S}$ holds for each situation s. To do this we note that property (5.85) implies that from the dual problem (i.e. minimization of the dual function) viewpoint, the non-blocking situations (i.e. situations not in $\mathcal{S}$) are not important: the primal problem (5.52)-(5.55), (5.63), (5.73), reduced to the situations from set $\mathcal{S}$, and the original problem taking into account all situations $s \in \{s_1, s_2, \ldots, s_S\}$, are identical from the dual problem viewpoint. The common dual function is given by:

$$W(\boldsymbol{\lambda}) = \max_{\boldsymbol{u} \in U} \sum_{s \in \mathcal{S}} \lambda_s R_s(\boldsymbol{u}) \ , \ \ \boldsymbol{\lambda} \in \Lambda', \tag{5.89}$$

where

$$\Lambda' = \left\{ \ \boldsymbol{\lambda} \in \Lambda : \lambda^*_s = 0 \text{ for } s \notin \mathcal{S} \ \right\}. \tag{5.90}$$

Now suppose that $\lambda^*_{s'} = 0$ for some $s' \in \mathcal{S}$. This leads to the following further restriction of the dual function:

$$W(\boldsymbol{\lambda}) \ = \ \max_{\boldsymbol{u} \in U} \sum_{s \in \mathcal{S} \setminus \{s'\}} \lambda_s R_s(\boldsymbol{u}) \ , \ \ \boldsymbol{\lambda} \in \Lambda' \ , \ \ \lambda_{s'} = 0 \ . \tag{5.91}$$

The form of (5.91) implies that it is the dual function of the considered optimization task further reduced by deleting situation s' from the list of blocking situations. Then, by Property 5.3, one of the situations in the set $\mathcal{S} \setminus \{s'\}$, say situation s'', becomes non-blocking with respect to t^*. This means that in the task reduced to set $\mathcal{S} \setminus \{s'\}$, the set of blocking situations is a subset of $\mathcal{S} \setminus \{s', s''\}$. This in turn implies that in the new task $\lambda^*_{s''} = 0$ for any optimal $\boldsymbol{\lambda}^*$, and we can further modify the dual function arriving at

$$W(\boldsymbol{\lambda}) \ = \ \max_{\boldsymbol{u} \in U} \sum_{s \in \mathcal{S} \setminus \{s', s''\}} \lambda_s R_s(\boldsymbol{u}) \ , \ \ \boldsymbol{\lambda} \in \Lambda' \ , \ \ \lambda_{s'} = \lambda_{s''} = 0 \ . \tag{5.92}$$

Continuing the above procedure of reducing the set $\mathcal{S}$ we will eventually reach the empty set of blocking situations, contradicting that t^* is the optimal solution of the original task. Hence, $\lambda^*_{s'} > 0$ and this completes the first part of the proof.

The proof that: if for each situation s, $\lambda^*_s = 0 \Rightarrow s \notin \mathcal{S}$, then Property 5.3 holds, is omitted here and can be found in Section 13.1.3 of [31]. □

To summarize: if Property 5.3 holds then all situations with $\lambda^*_s = 0$ are non-blocking and the optimal solution of Step 1 in the improved algorithm will always result in increasing the value of t^*.

5.3.6 Other combinations of criteria for RDT

Certainly, combining MMF and PF fairness (as it was presented in the previous sections) is just one of the ways for designing resilient networks with fair bandwidth distribution. Recall that using two criteria (in this case–MMF and PF) for network design allows us not only to fairly distribute the bandwidth among demands, but also to do so in each of the (considered) network states through the revenues. In the previous sections we have mainly focused on the MMF-PF allocation scheme, here we will present how the discussed network design models can be extended for considering other combinations of criteria. Assuming that it is still desired to have PF bandwidth distribution between demands, the alternatives to govern the revenues consist in, but are not limited to, PF allocation and total revenue maximization (RM). RM is the same kind of objective as TM (4.6) , just applied to the revenue values, instead of the flows. We will give mathematical models for both of these options here and provide some comparative results for networks N_{12} and N_{41} in the numerical results section below.

Let us now define a resilient network design problem with PF bandwidth allocation, both among the demands and among the revenues. We will refer to this allocation scheme as PF-PF.

RDT-PF-PF:

$$\max \quad R = \sum_s \log R_s = \sum_s \log(\sum_d w_{ds} \log(X_{ds})) \tag{5.93}$$

$$\text{s.t.:} \quad (5.52) - (5.55).$$

The problem RDT-PF-PF is identical to problem RDT-MMF-PF (see Section 5.3.3), except of the objective function (5.93), which is the PF objective applied to situation-dependent revenues (5.56).

Similarly, a network design problem with PF bandwidth allocation among demands and RM applied to revenues we will refer to this scheme as RM-PF) is defined as:

RDT-RM-PF:

$$\max \quad R = \sum_s R_s = \sum_d \sum_s w_{ds} \log(X_{ds}) \tag{5.94}$$

$$\text{s.t.:} \quad (5.52) - (5.55).$$

Applying the linear approximation of the logarithmic function discussed in Section 4.3.4 and relaxing variables, so that $x_{dps}, y_e \in \mathbb{R}^+$, makes both problems linear. As such, they can be resolved using generic LP solvers, such as CPLEX [49], and do not require special algorithms. Such linearized and relaxed problems are considered in the numerical section below.

5.3.7 Numerical experiments

A number of experiments have been carried out to test the algorithms for different network examples. The following subsections contain selected input data and the results of the algorithms for four different size networks (see Figures 5.5-5.8). The network models used are presented in Table 5.1.

The networks of particular interest are the "Polish backbone network", N_{12}, and the artificial network, N_{41}, shown in Figures 5.7 and 5.8, respectively. Because of the limited space it is not feasible to give complete numerical results for these networks. For illustrative purposes, a complete description of the obtained numerical results and algorithm parameters for N_3 is given in Section 5.3.7.1. A summary of selected numerical values obtained by the algorithms applied to N_{12} is given in Subsection 5.3.7.2. Extensive results for networks N_{12} and N_{41} are given when comparing different combined criteria. For N_5 only execution times are reported. In network cases N_3, N_5 and N_{12} the budget B is equal to 1000, while for N_{41} it is 5000.

ref. code	# nodes	# links	# demands	# paths per demand	# failure situations
N_3	3	3	3	2	4
N_5	5	7	10	2-3	8
N_{12}	12	18	66	6-13	19
N_{41}	41	72	100	3	35

Table 5.1: The networks used for experiments.

5.3.7.1 Simple network, N_3

We start by presenting results achieved by the algorithms applied to the simple 3-node network N_3 depicted in Figure 5.5. In this experiment the revenue coefficients w_{ds}, and cost coefficients ξ_e are all put equal to 1. Tables 5.2 and 5.3 give the input data.

Results from these trials are found in Table 5.4 (empty fields indicate that an entity remains unchanged with respect to the preceding step). Values of flows,

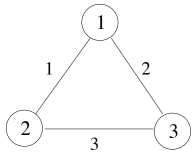


Figure 5.5: The N_3 network.

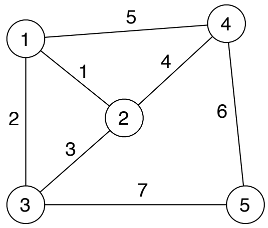


Figure 5.6: The N_5 network.

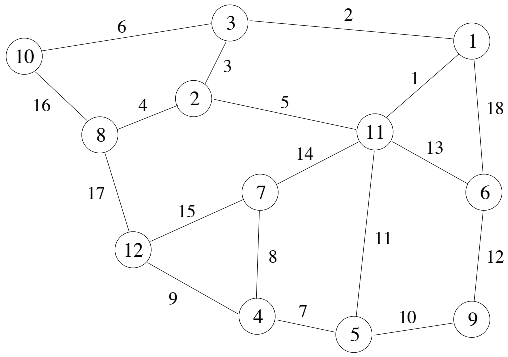


Figure 5.7: The N_{12} network.

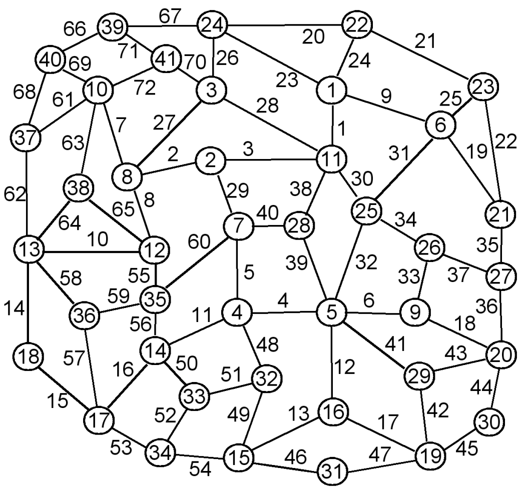


Figure 5.8: The N_{41} network.

Demand		$d=1$	$d=2$	$d=3$
Nodes		1-2	1-3	2-3
Links (e)	$p=1$	1	2	3
	$p=2$	2,3	1,3	1,2

Table 5.2: Demands and paths for network N_3.

Links	α_{es}				
	$s=1$	$s=2$	$s=3$	$s=4$	ξ_e
$e=1$	1	0	1	1	1
$e=2$	1	1	0	1	1
$e=3$	1	1	1	0	1
w_{ds}	1	1	1	1	

Table 5.3: α_{es}, w_{ds} and ξ_e for network N_3.

x_{dps}, are given only if not equal to zero. Final results (after the algorithm completion) are shown in the rightmost position. Note that $R_{s'}$ (in contrast to R_s) is relevant only for the basic algorithm, and that λ_s on the other hand is relevant only for the modified algorithm.

5.3.7.2 Backbone network, N_{12}

Already network N_{12} (medium-size Polish backbone network, Figure 5.7) is quite challenging for the algorithms in terms of computation times. Since for instance the number of the non-zero flows, x_{dps}, is quite big in this trial, only selected numerical results are given. As in the N_3 case, the revenue coefficients, w_{ds}, are all put equal to 1. Link costs are given in Table 5.5. Situations are defined in accordance to the N_3 case, i.e. each situation corresponds to one exclusively, completely failed link, except for the nominal situation, when no links are broken. In effect, α_{es} are simply obtained by expanding Table 5.3. The evolution of R_s over (selected) iterations n, along with the evolution of the auxiliary variable t and the resulting link capacities y_e, is illustrated in Table 5.6. The table reveals at which iteration (n) each individual situation was blocked. By studying the second column (t^*) and the 8th column ($R_s, n = 15$) it is concluded that situations 4,5,14 and 17 are blocked at $n = 0$, $t^* = 91.62$. Situation 7 is blocked for $n = 1$, $t^* = 92.15$ and 16 for $n = 2$, $t^* = 92.32$, respectively, and so on.

n	0	1		2
Instant	After Step 1	After Step 2	After Steps 3	After Step 2
t^*	20.35		26.8	
t_s	$t_s = 20.35$, $s = 0, 1, 2, 3$		$t_0 = 26.8$, $t_s = 20.35,\ s = 1, 2, 3$	
R_s	$R_s = 20.35$, $s = 0, 1, 2, 3$	$R_{0'} = 26.80$, $R_{s'} = 20.35$, $s' = 1, 2, 3$	$R_0 = 26.8$, $R_s = 20.35,\ s = 1, 2, 3$	$R_{0'} = 26.8$
y	$y_1 = y_2 = y_3 = 333.33$		$y_1 = y_2 = y_3 = 333.33$	
x_{dps}	$x_{110} = 24.81$ $x_{210} = 24.81$ $x_{310} = 283.72$ $x_{320} = 308.53$ $x_{121} = 24.81$ $x_{211} = 308.53$ $x_{311} = 308.53$ $x_{112} = 308.53$ $x_{222} = 24.81$ $x_{312} = 308.53$ $x_{113} = 308.53$ $x_{213} = 308.53$ $x_{323} = 24.81$		$x_{110} = 333.33$ $x_{210} = 333.33$ $x_{310} = 333.33$ $x_{121} = 24.81$ $x_{211} = 308.53$ $x_{311} = 308.53$ $x_{112} = 308.53$ $x_{222} = 24.81$ $x_{312} = 308.53$ $x_{113} = 308.53$ $x_{213} = 308.53$ $x_{323} = 24.81$	
Z_1	$s = 0, 1, 2, 3$	$s = 0$		$\emptyset$
Z_0	$\emptyset$	$s = 1, 2, 3$		$s = 0, 1, 2, 3$
λ_s	For constraint (5.73): $\lambda_0 = 0,$ $\lambda_1 = \lambda_2 = \lambda_3 = 0.33$		For constraint (5.71): $\lambda_0 = 1$	

Table 5.4: Numerical results for the basic and the modified algorithms applied to N_3.

e	1	2	3	4	5	6	7	8	9
ξ_e	1.85	3.4	1	1.45	2.3	2.9	1	1.6	2.2
e	10	11	12	13	14	15	16	17	18
ξ_e	1.5	2.3	1.4	1.65	1.25	2.3	1.55	1.35	1.7

Table 5.5: Link marginal costs for network N_{12}.

5.3.7.3 Network N_{41}

Due to its size the network N_{41} was the most resource demanding in computations. As in the case with other networks, for N_{41} the revenue coefficients, w_{ds}, are all put equal to 1. Link costs varied, dependent on link length, between 5.7 and 16.4, with the mean being 9.9, the 25^{th} percentile of 7.8 and the 75^{th} percentile of 11.7. The costs are shown in Table 5.7. Situations are defined in accordance to the N_3 case, i.e. each situation corresponds to one exclusively, com-

n	t^*	s	R_s, n = 0	1	2	14	15	e	y_e
0	91.62	0	91.62	92.15	92.32	97.83	101.64	1	25.89
1	92.15	1	91.62	92.15	92.32	96.22	96.22	2	33.32
2	92.32	2	91.62	92.18	92.32	92.33	92.33	3	42.54
3	92.33	3	91.62	92.15	92.32	93.96	93.96	4	49.76
4	93.11	4	91.62	91.62	91.62	91.62	91.62	5	40.46
5	93.56	5	91.62	91.62	91.62	91.62	91.62	6	13.70
6	93.96	6	91.62	92.15	92.32	97.37	97.37	7	54.97
7	94.06	7	91.62	92.15	92.15	92.15	92.15	8	27.97
8	94.27	8	91.62	92.15	92.32	95.93	95.93	9	22.43
9	95.93	9	91.62	92.15	92.32	94.27	94.27	10	31.64
10	96.22	10	91.62	92.15	92.32	93.56	93.56	11	16.55
11	96.26	11	91.62	92.15	92.32	97.83	97.83	12	38.13
12	96.68	12	91.62	92.15	92.32	93.11	93.11	13	22.43
13	97.37	13	91.62	92.15	92.32	96.68	96.68	14	55.60
14	97.83	14	91.62	91.62	91.62	91.62	91.62	15	22.24
15	101.64	15	91.62	92.15	92.32	96.26	96.26	16	25.24
		16	91.62	92.15	92.32	92.32	92.32	17	40.45
		17	91.62	91.62	91.62	91.62	91.62	18	29.56
		18	91.62	92.15	92.32	94.06	94.06		

Table 5.6: Selected results and parameters for network N_{12}.

pletely failed link, except for the nominal situation, when no links are broken. As in the case with network N_{12}, α_{es} are simply obtained by expanding Table 5.3. The number of demands was limited to 100 and the number of situations to 35 in order to still get reasonable computation times. The situations were selected randomly from the set of 73 single link failure situations possible.

5.3.7.4 Comparing the combinations of fairness criteria

In order to illustrate the effects of the different fairness schemes used in resilient network design problem, below we provide some results for networks N_{12} and N_{41}. There are two sets of results for network N_{12} presented: Table 5.8 and Figures 5.9 and 5.10 present some results for the case when all the link costs are assumed equal to one, i.e., $\xi_e \equiv 1$. This allows for a comparison of the different fairness schemes without going into the discussion of how do different cost factors bias the results. The second set of results for N_{12}, comprised of

e	1	2	3	4	5	6	7	8	9
ξ_e	7.8	9.4	13.7	10.6	9.7	9.3	10.1	8.9	12.9
e	10	11	12	13	14	15	16	17	18
ξ_e	14.2	10.2	11.1	11.8	11.9	10.7	9.9	12.2	10.9
e	19	20	21	22	23	24	25	26	27
ξ_e	12.0	16.4	15.6	14.6	15.2	7.8	6.6	7.3	13.5
e	28	29	30	31	32	33	34	35	36
ξ_e	15.5	8.1	6.9	12.6	12.0	7.3	8.3	7.1	8.7
e	37	38	39	40	41	42	43	44	45
ξ_e	8.5	8.3	10.3	6.9	12.1	9.2	9.7	7.8	7.8
e	46	47	48	49	50	51	52	53	54
ξ_e	10.7	11.1	8.3	9.3	6.8	8.2	7.6	7.7	8.9
e	55	56	57	58	59	60	61	62	63
ξ_e	5.7	5.7	11.7	9.7	8.2	12.5	9.9	12.5	11.4
e	64	65	66	67	68	69	70	71	72
ξ_e	8.8	10.9	8.1	11.2	9.3	6.4	6.2	7.0	8.3

Table 5.7: Link marginal costs for network N_{41}.

Table 5.9 and Figures 5.11 and 5.12, are for the case when "real"[3] link costs (shown in Table 5.5) are used. Results for network N_{41} are produced with the real costs and are shown in Table 5.10 and Figure 5.13. The two cases with the real network costs illustrate the outcomes of the considered schemes when the allocation is affected also by the link cost factor. The tables present main statistical measures calculated for revenues R_s and link capacities y_e for each of the considered schemes. The preferred values are bolded in the tables. Figures 5.9, 5.11 and 5.13 show revenue vectors sorted in the non-decreasing order (in order to compare them to the MMF-PF approach). The sorting implies that the order of the revenues in the resulting sorted vector may in general be different for the different schemes. Thus it is not relevant to give situation indices in the figures. Therefore there are no values on the abscise in these figures. Different schemes are named in the figures just after the criterion which governs the allocation of revenues among the situations, since it was assumed that in all the cases bandwidth is allocated among the demands according to the PF principle. There are also two values of revenues indicated in the figures. These as R^{PF}_{min}, the minimum

[3]In this context, *real* does not mean that the costs come from some real-life scenario, but that they somehow reflect the lengths of the links.

revenue for the PF-PF option, and R_{min}^{MMF}, the minimum revenue for the MMF-PF option. These are the two values we are mostly interested in for illustrating the effects of the MMF versus the PF allocation schemes.

Figures 5.10 and 5.12 show the resulting link capacities. For convenience, the real link costs are indicated as well on the second abscise in Figure 5.12. There is no corresponding figure presented for the N_{41} network due to the large set of values. Instead, the reader is referred to Table 5.10 presenting statistical measures for these results.

Comparing the thee schemes in terms of statistical measures, the preferred solution in the context of fair flow allocation is the one which has the highest minimum revenue value (Min), highest maximum revenue value (Max), lowest gap between the two (Range), lowest variation (Var), and highest values of mean (Mean), median (Median), 25^{th} and 75^{th} percentiles (P25 and P75, respectively), as well as highest total sum of the revenues.

Examining the first set of results (for network N_{12} with $\xi_e \equiv 1$), we notice that for the RM-PF and the PF-PF cases the values are very close. The same was observed also for the cases (both for N_{12} and N_{41}) with the real costs. Therefore RM-PF is omitted in the subsequent two sets of results. Besides, as it can be seen from Table 5.8, the PF-PF scheme performs even better in some aspects. Specifically, both median and P25 are higher for the PF-PF case, and are the best scores among the three schemes for this network case. We can also see that both the RM-PF and the PF-PF schemes achieve higher mean and maximal revenues, as well as result in the larger sum of revenues, compared to MMF-PF case. However, this comes at a cost of smaller minimum revenue value and greater range/variation, which are best for the MMF-PF case. The gap between the revenues for the three schemes and all the situations can be seen in Figure 5.9. The minimum revenues achieved by the PF-PF and MMF-PF cases are denoted in the figure by R_{min}^{PF} and R_{min}^{MMF}, respectively. As expected, it is common to all the network scenarios studied that $R_{min}^{MMF} > R_{min}^{PF}$. Together with the lowest variation, this is the biggest advantage of the MMF-based scheme. The same results are observed also for the other two network cases, namely N_{12} and N_{41} with real costs. The sorted revenue vectors, with the gap between R_{min}^{PF} and R_{min}^{MMF} indicated, for these two cases are shown in Figures 5.11 and 5.13.

In practice a network operator may want to have at least the same minimum revenue preserved in all the failure situations, and this revenue to be as large as possible. As it was discussed, the revenue reflects (straightforward, if $w_{ds} \equiv 1$) total network bandwidth in a given network operation state. Thus, the minimum revenue value that is attained in any of the situations translates into the minimum

	R_s			y_e		
	RM-PF	PF-PF	MMF-PF	RM-PF	PF-PF	MMF-PF
Min	126.074	126.066	**127.840**	35.748	35.030	32.446
Max	**137.957**	**137.957**	134.381	91.127	91.180	82.304
Range	11.883	11.890	**6.541**	55.379	56.150	49.858
Var	7.076	7.052	**4.496**	205.460	202.007	225.093
Mean	**131.105**	**131.105**	130.145	55.556	55.556	55.556
P25	129.497	**129.559**	127.840	45.288	45.419	45.513
Median	131.600	**131.646**	130.516	53.452	53.811	52.799
P75	**132.527**	**132.527**	131.404	64.883	63.692	69.015
Sum	**2490.989**	**2490.989**	2472.751	1000	1000	1000

Table 5.8: Revenue and capacity statistics for N_{12} with $\xi_e \equiv 1$.

guaranteed bandwidth in the network in any of the (considered) network states. Knowing this value not only facilitates the decisions of the operator about what bandwidth guarantees it can provide to the customers, but also, if this value is maximized, it can imply increased customers' satisfaction with the network, and in turn add marketing value to the operator's network services. Another argument, supporting the MMF-PF allocation scheme is the lowest (among the three considered alternatives) variation of revenues across the network states, as can be seen from Tables 5.8, 5.9 and 5.10 This shows certain stability in revenue allocation.

If, on the other hand, the operator is interested in maximizing its overall profit, directly linked to the total revenue (see Sum row in the tables), then either the RM-PF or the PF-PF option is the way to go. However, since for the considered test cases a better median value for revenues was observed when the PF-PF scheme was used, PF-PF should be the preferred alternative.

No general conclusions can be drawn regarding the resulting link capacities, and it is up to an operator to decide which allocation pattern is preferable depending on the network expansion strategy, forecasted demands, etc. For the network instances with real costs it has been observed (see Tables 5.9 and 5.10) that minimum range between the assigned capacities is achieved by PF-PF option, while minimum variance is achieved by MMF-PF. The maximum 75^{th} percentile, as well as maximum capacity value were observed for the PF-PF option.

5.3.7.5 Platforms

The introduced algorithms have been implemented in three variants, all on the same hardware, namely a Dell Precision 220 PC, with an Intel Pentium III-

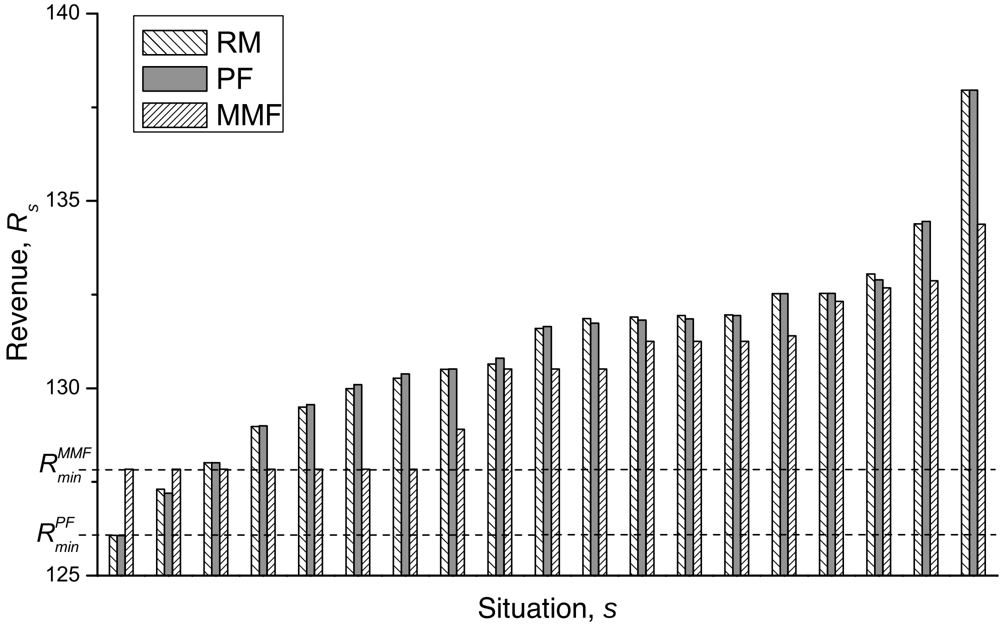


Figure 5.9: Sorted revenues for RM-PF, PF-PF and MMF-PF models for N_{12}, $\xi_e \equiv 1$.

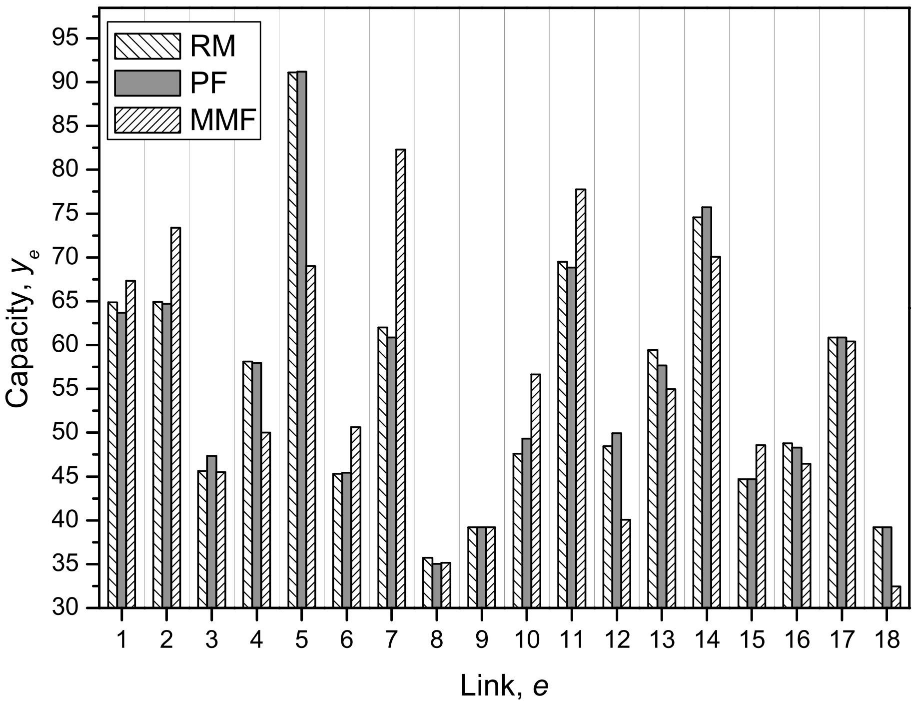


Figure 5.10: Link capacities for RM-PF, PF-PF and MMF-PF models for N_{12}, $\xi_e \equiv 1$.

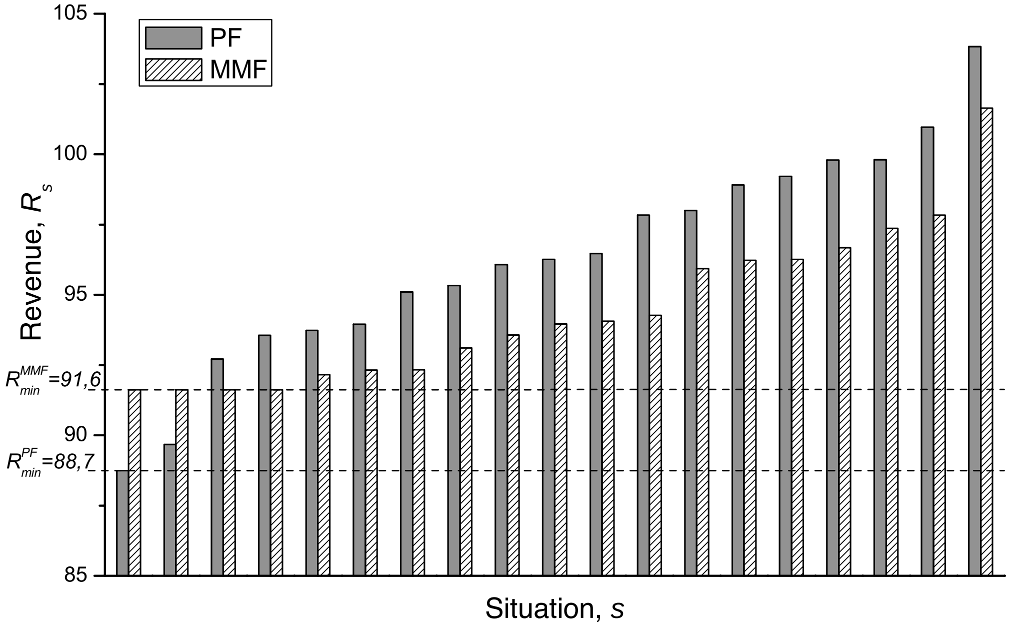


Figure 5.11: Sorted revenues for PF-PF and MMF-PF models for N_{12} with real ξ_e.

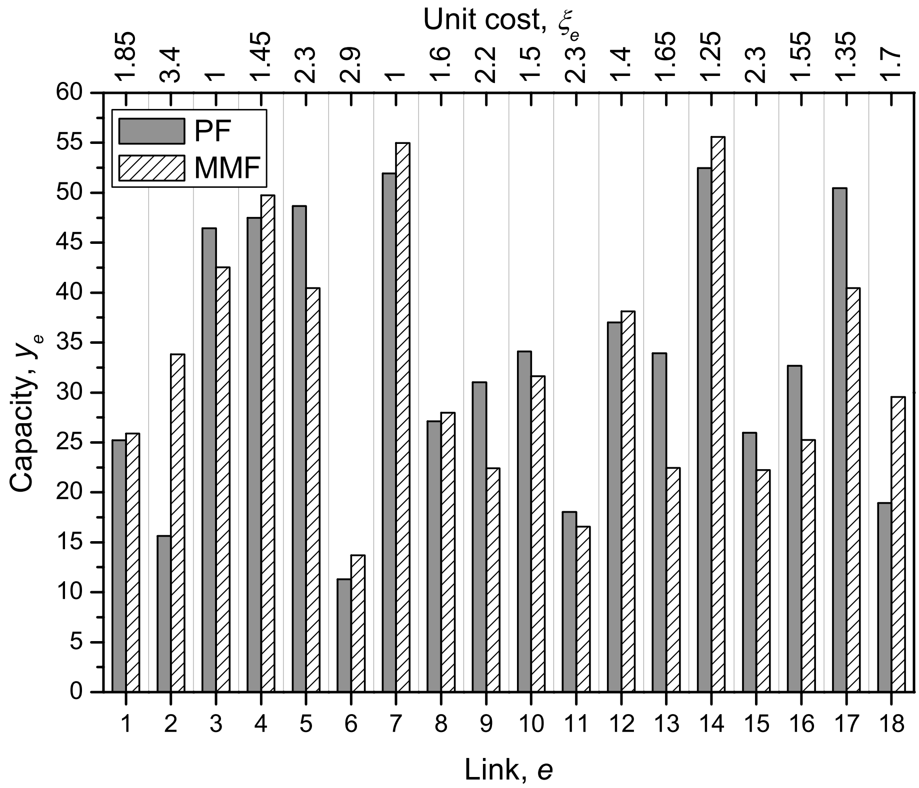


Figure 5.12: Link capacities for PF-PF and MMF-PF models for N_{12} with real ξ_e.

	R_s		y_e	
	PF-PF	MMF-PF	PF-PF	MMF-PF
Min	88.738	**91.617**	11.301	13.700
Max	**103.833**	101.641	52.462	55.601
Range	15.095	**10.024**	41.160	41.901
Var	14.422	**7.480**	178.157	155.029
Mean	**96.312**	94.429	33.797	32.966
P25	**93.729**	92.152	25.205	22.433
Median	**96.259**	93.957	33.311	30.603
P75	99.217	96.258	47.491	40.460
Sum	**1829.921**	1794.160	608.355	593.386

Table 5.9: Revenue and capacity statistics for N_{12} with real ξ_e.

	R_s		y_e	
	PF-PF	MMF-PF	PF-PF	MMF-PF
Min	24.808	**30.859**	0	0
Max	**50.717**	47.209	28.055	31.606
Range	25.909	**16.350**	28.055	31.606
Var	49.347	**34.367**	37.885	37.166
Mean	**42.556**	39.306	7.446	7.459
P25	**37.460**	33.100	2.164	2.164
Median	**45.605**	39.815	5.797	6.083
P75	**48.411**	44.071	12.232	11.333
Sum	**1489.473**	1375.708	536.144	537.031

Table 5.10: Revenue and capacity statistics for N_{41} with real ξ_e.

Reference code	Programming environment	Solver
C++	Microsoft Visual C++ 6.0; Callable CPLEX libraries	CPLEX 7.5.0
AMPL	AMPL ver. 20010215	CPLEX 7.5.0
Matlab[a]	MATLAB 6.1	MATLAB Optimization toolbox

[a] for the basic algorithm

Table 5.11: Software implementations of the algorithms.

1GHz CPU, RAM of 256 MB, a Quantum Atlas10K2-TY092L SCSI HDD and the Windows 2000 Pro SP2 OS. The different software implementations are given in Table 5.11.

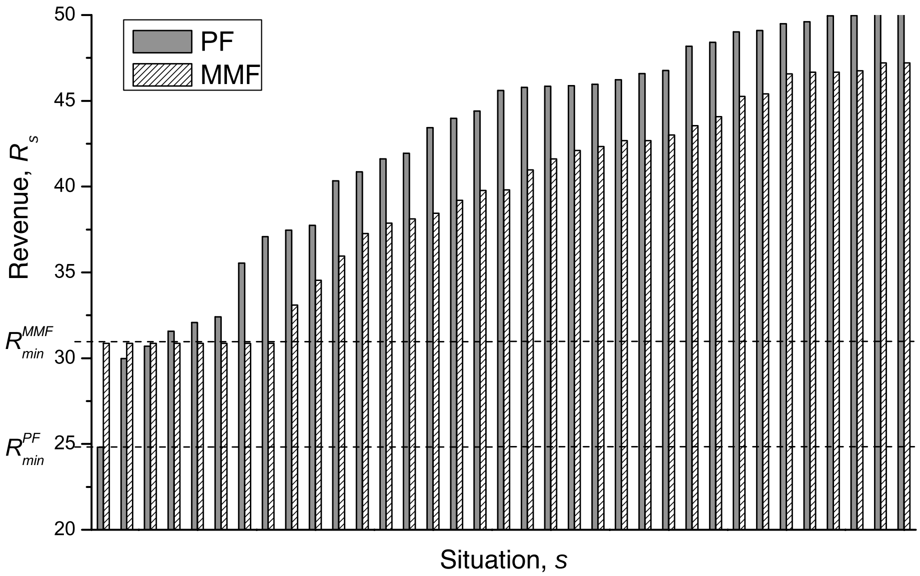


Figure 5.13: Sorted revenues for PF-PF and MMF-PF models for N_{41} with real ξ_e.

5.3.7.6 Computational efficiency

It is of interest to illustrate the computational efficiency of the algorithms and their implementations in terms of execution time. The corresponding data is found in Table 5.12. Tests for N_{41} have been performed using only AMPL implementations. Therefore, some entries for N_{41} in Table 5.12 are left blank. The choice to use the AMPL implementations for these tests is based on the observation that these implementations exhibited the smallest difference between execution times of the basic and modified algorithms for the other networks considered. Therefore, since the computation times for large networks such as N_{41} are quite long for all the considered implementations, we have chosen to use the AMPL implementations to illustrate the differences in time performance of the two proposed algorithms.

5.3.7.7 Remarks

As seen from Table 5.12, for the network examples considered the modified algorithm performs, as expected, much better than the basic one. One can also notice that performance advantage of the modified algorithm increases with the size of the network design problem used.

The revenues, R_s, for different situations after the final iteration (algorithm completion) may differ only by a small number, although making actual flow

Reference code	Execution times (seconds)				
	Implementations of basic algorithm			Implementations of modified algorithm	
	C++	AMPL	Matlab	C++	AMPL
N_3	< 1	2	4	< 1	< 1
N_5	< 1	3	12	< 1	2
N_{12}	3859	390	6.4e4	179	325
N_{41}		24945			6960

Table 5.12: Execution times.

differences significant, since such a revenue difference is logarithmically related to the aggregated flow difference.

5.3.8 Overcoming numerical difficulties

Since representation of the real numbers in computers is finite, the results of computations are rounded-off, and the strict equality tests, performed during execution of the algorithm (e.g. for selection of situations to different sets), may fail. Note that this concerns only the basic algorithm described in Subsection 5.3.5.1, since it implies checking of strict equality between two variables (R_s and t_s) in establishing set possession (Y, Z_1 or Z_0) for each situation (definition of situation sets in (5.65) and (5.68)). Values of these variables can be internally rounded-off during arithmetical operations, causing the strict equality checks to fail. For this reason it is convenient to introduce a perturbation tolerance, ε, representing the maximum value by which R_s and t_s can differ to still be interpreted as equal. The value of ε has to be selected not too large so its influence on the solution results is minimized, yet large enough to cover the possible fluctuations in the last positions of the number representation. Accordingly, the following numerical modifications of the basic algorithm are introduced.

- The definition of sets (5.65) is replaced by
  $Y := \left\{\ s \in Z_1 : -\varepsilon \leq R_s - t_s \leq \varepsilon\ \right\}$ and
  $Z_1 := \left\{\ s \in Z_1 : R_s - t_s > \varepsilon\ \right\}$.

- The redefinition of sets (5.68) is replaced by
  $Z_0 := Z_0 \cup \left\{\ s' : -\varepsilon \leq R_{s'} - t_{s'} \leq \varepsilon\ \right\}$ and
  $Z_1 := Z_1 \cup \left\{\ s' : R_{s'} - t_{s'} > \varepsilon\ \right\}$.

5.4 Path generation for the single-layer resilient network

In Section 5.2 we have discussed how path generation can be applied for resolving problem NDT. Although there is an explicit solution to problem NDT, we carried the detailed derivation of the PG algorithm, mainly for illustrative purposes. We will now take advantage of this and will use the same approach for developing PG algorithm for resolving linearized and relaxed problem RDT. Recall, that path generation technique is useful only when the set of all allowable paths is somehow restricted, i.e., by maximum hop count, thus making impossible to state the problem in node-link formulation. Assuming this is the case, and recalling that problem RDT has to be solved with the help of a special algorithm (e.g., the one presented in Section 5.3.5.4), we will develop a PG algorithm for generating new paths, which will replace the task solved in Step 1 of the algorithm in Section 5.3.5.4. Furthermore, we will extend the original problem RDT by including lower and upper bounds on total demand volumes. Since the path generation technique can be applied only to linear problems, we will consider a relaxation of the problem assuming $x_{dps}, y_e \in \mathbb{R}^+$ and t, $f_{ds} \in \mathbb{R}$. Let us first restate the optimization task of Step 1. We will denote it by LRRDT.

LRRDT:

$$\max \quad t \tag{5.95}$$

s.t.:

$$[\pi \geq 0] \quad \sum_e \xi_e y_e \leq B, \tag{5.96}$$

$$[\sigma_{ds} \geq 0] \quad h_{ds} \leq \sum_p x_{dp}, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{5.97}$$

$$[\tau_{ds} \geq 0] \quad \sum_p x_{dp} \leq H_{ds}, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{5.98}$$

$$[\gamma_{es} \geq 0] \quad \sum_d \sum_p \delta_{edp} x_{dps} \leq \alpha_{es} y_e, \quad \forall e \in \mathcal{E}, s \in \mathcal{S} \tag{5.99}$$

$$[\beta_{dsi} \geq 0] \quad f_{ds} \leq a_i \sum_p x_{dps} + b_i, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S}, i \in \mathcal{I} \tag{5.100}$$

$$[\lambda_s \geq 0] \quad t_s - \sum_d w_{ds} f_{ds} \leq 0, \qquad \forall s \in Z_0 \tag{5.101}$$

$$[\lambda_s \geq 0] \quad t - \sum_d w_{ds} f_{ds} \leq 0, \qquad \forall s \in Z_1 \tag{5.102}$$

$$t, f_{ds} \in \mathbb{R}; x_{dps} \in \mathbb{R}^+, \qquad \forall d \in \mathcal{D}, p \in \mathcal{P}_d, \forall s \in \mathcal{S} \tag{5.103}$$

$$y_e \in \mathbb{R}^+, \qquad \forall e \in \mathcal{E}. \tag{5.104}$$

Dual variables associated with constraints (5.96)-(5.102) are indicated in brackets in front of each constraint. Note that dual variables λ_s are associated both with constraint (5.101) and (5.102). This is a natural choice since constraints are essentially the same, and indexing of situations among them does not overlap, i.e., $Z_0 \cap Z_1 = \emptyset$. For convenience we convert the objective into minimization. The Lagrangian function for the problem is:

$$\begin{aligned} L(t, \boldsymbol{f}, \boldsymbol{x}, \boldsymbol{y}\ ;\ \pi, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\gamma}) &= \sum_d \sum_s \Big(\sum_i \beta_{dsi} - w_{ds}\lambda_s \Big) f_{ds} \\ &+ \sum_d \sum_p \sum_s \Big(\tau_{ds} - \sigma_{ds} - \sum_i a_i \beta_{dsi} + \sum_e \delta_{edp}\gamma_{es} \Big) x_{dps} \\ &+ \sum_e \Big(\pi\xi_e - \sum_s \alpha_{es}\gamma_{es} \Big) y_e + \Big(\sum_{s \in Z_1} \lambda_s - 1 \Big) t \\ &+ \sum_d \sum_s \Big(h_{ds}\sigma_{ds} - H_{ds}\tau_{ds} - \sum_i b_i \beta_{dsi} \Big) + \sum_{s \in Z_0} t_s \lambda_s - \pi B \end{aligned} \tag{5.105}$$

and the dual function of LRRDT becomes:

$$\begin{aligned} W(\pi, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\gamma}) &= \min_{\boldsymbol{x}, \boldsymbol{y} \geq \boldsymbol{0}\ ,\ t \gtreqless 0, \boldsymbol{f} \gtreqless \boldsymbol{0}} L(t, \boldsymbol{f}, \boldsymbol{x}, \boldsymbol{y}\ ;\ \pi, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\gamma}) \\ &= \sum_d \sum_s (h_{ds}\sigma_{ds} - H_{ds}\tau_{ds} - \sum_i b_i \beta_{dsi}) + \sum_{s \in Z_0} t_s \lambda_s - \pi B. \end{aligned} \tag{5.106}$$

Constraining the values of dual variables only to those leading to the bounded dual function W, the dual problem is stated as follows:
DLRRDT:

$$\max \quad W(\pi, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\gamma}) \tag{5.107}$$

$$\text{s.t.:} \quad \sum_{s \in Z_1} \lambda_s = 1, \tag{5.108}$$

$$\sum_i \beta_{dsi} = w_{ds}\lambda_s, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{5.109}$$

(5.110)

$$\sigma_{ds} - \tau_{ds} + \sum_i a_i \beta_{dsi} \leq \sum_e \delta_{edp}\gamma_{es}, \qquad \forall d \in \mathcal{D}, p \in \mathcal{P}_d, \quad \forall s \in \mathcal{S} \tag{5.111}$$

$$\sum_s \alpha_{es}\gamma_{es} \leq \xi_e \pi, \qquad \forall e \in \mathcal{E} \tag{5.112}$$

$$\pi, \lambda_s, \gamma_{es} \in \mathbb{R}^+ \qquad \forall e \in \mathcal{E}, s \in \mathcal{S} \tag{5.113}$$

$$\beta_{dsi}, \sigma_{ds}, \tau_{ds} \in \mathbb{R}^+, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S}, \quad \forall i \in \mathcal{I} \tag{5.114}$$

Let t^*, $\boldsymbol{f}^*$,$\boldsymbol{x}^*$,$\boldsymbol{y}^*$ be any primal optimal solution of LRRDT. Similarly, let π^*,$\boldsymbol{\beta}^*$, $\boldsymbol{\lambda}^*$,$\boldsymbol{\sigma}^*$,$\boldsymbol{\tau}^*$, γ^* be any dual optimal solution. The primal and dual optimal solutions form a saddle point of the Lagrangian. (The dual optimal multipliers π^*,$\boldsymbol{\beta}^*$,$\boldsymbol{\lambda}^*$,$\boldsymbol{\sigma}^*$,$\boldsymbol{\tau}^*$,$\boldsymbol{\gamma}^*$ can be obtained from an LP solver.) It is observed, that for any saddle point of the Lagrangian (5.26), the following optimality conditions hold, implied by the complementary slackness property, for the paths that carry flow in the optimal solution:

$$\sum_{s \in Z_1} \lambda_s^* = 1, \tag{5.115}$$

$$\sum_i \beta_{dsi}^* = w_{ds}\lambda_s^*, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{5.116}$$

$$\sigma_{ds}^* - \tau_{ds}^* + \sum_i a_i \beta_{dsi}^* = \mathcal{L}_{ds}, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{5.117}$$

$$\sum_s \alpha_{es}\gamma_{es}^* = \xi_e \pi^*, \qquad \forall e \in \mathcal{E} \tag{5.118}$$

In an optimal solution, each variable γ_{es}^* represent a cost of link e in situation s. Let $\mathscr{P}_d$ be the given list of candidate paths for demand d and let $\mathcal{L}_{ds}$ in equation (5.117) be a length of the shortest path for demand d and situation s in this list with respect to metrics $\boldsymbol{\gamma}^* = (\gamma_{es} : e \in \mathcal{E}, s \in \mathcal{S})$:

$$\mathcal{L}_{ds} = \min_{p \in \mathcal{P}_d} \left(\sum_e \delta_{edp}\gamma_{es}^* \right), \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{5.119}$$

where $\mathcal{P}_d$ contains indices of paths in set $\mathscr{P}_d$. Define a set $\mathcal{S}^e$ containing all situations in which link e is fully available, i.e.,

$$\mathcal{S}^e = \{s \in \mathcal{S} : \alpha_{es} = 1\} \qquad \forall e \in \mathcal{E}. \tag{5.120}$$

Then we can rewrite equation (5.118) as follows:

$$\sum_{s \in \mathcal{S}^e} \gamma^*_{es} = \xi_e \pi^*, \qquad \forall e \in \mathcal{E}. \tag{5.121}$$

One can observe from (5.121) that in general, all the situations in which some link is available share a part of a real link cost ξ_e, scaled by π^*.

Another important observation following from (5.115)-(5.118) is that we can have a meaningful interpretation of γ^*_{es} and $\mathcal{L}_{ds}$ lengths only for the blocking situations. Otherwise, if we consider some non blocking situation, for which $\lambda^*_s = 0$, we can get that $\mathcal{L}_{ds} = 0$, and thus we cannot find any shorter path. We will use this fact in the PG algorithm, presented below.

Note that two properties we have identified when discussing PG for problem LRNDT hold for this problem as well. However, we now consider a flow for some demand in a particular situation. Thus, if the flow is between the bounds that must be obeyed for the demand in the given situation, then it holds that $\sigma^*_{ds} = \tau^*_{ds} = 0$. In general, only one of the variables can be non zero, which is the case when flow is equal to one of the bounds. Thus, the following equality always holds:

$$\sigma^*_{ds} \tau^*_{ds} = 0, \qquad \forall d \in \mathcal{D}, \ \forall s \in \mathcal{S}. \tag{5.122}$$

The second observation concerns dual variables β^*_{dsi}. For some demand d and situation s, usually one, and at most two β^*_{dsi} variables are non zero. As it was discussed in Section 5.2, if for some demand $\hat{d}$ and situation $\hat{s}$ only one approximation slope $i(\hat{d})$ is active for some optimal $X^*_{\hat{d}\hat{s}}$, then from equation (5.116) we get:

$$\beta^*_{\hat{d}\hat{s}i(\hat{d})} = w_{\hat{d}\hat{s}} \lambda^*_{\hat{s}}. \tag{5.123}$$

Furthermore, assuming that the flow is between bounds, i.e., $h_{\hat{d}\hat{s}} < X^*_{\hat{d}\hat{s}} < H_{\hat{d}\hat{s}}$, we can rewrite equation (5.117) as follows:

$$a_{i(\hat{d})} w_{\hat{d}\hat{s}} \lambda^*_{\hat{s}} = \mathcal{L}_{\hat{d}\hat{s}}. \tag{5.124}$$

When two approximation pieces are active, in general the following holds

$$\beta^*_{\hat{d}\hat{s}i(\hat{d})} + \beta^*_{\hat{d}\hat{s}i(\hat{d})+1} = w_{\hat{d}\hat{s}} \lambda^*_{\hat{s}}. \tag{5.125}$$

We will now state a proposition underlying PG algorithm for the problem. Suppose that all allowable paths for a given network instance of problem LRRDT realizing demand d belong to a set $\widehat{\mathscr{P}}_d$. Let $\widehat{\mathscr{P}}_{ds} \subseteq \widehat{\mathscr{P}}_d$ be a subset of paths for demand d that are fully available in (survive) the failure situation s. For each demand and situation, let $\wp_{ds} \in \widehat{\mathscr{P}}_{ds}$ be one of the shortest paths among all allowable paths surviving failure situation s, and let ℓ_{ds} be its cost with respect to the link cost metrics $\boldsymbol{\gamma}^*$, i.e.,

$$\ell_{ds} = \|\wp_{ds}\|_{\boldsymbol{\gamma}^*} = \min_{p \in \widehat{\mathcal{P}}_{ds}} \left(\sum_e \delta_{edp} \gamma^*_{es} \right) \qquad \forall d \in \mathcal{D}, \forall s \in \mathcal{S} \tag{5.126}$$

where set $\widehat{\mathcal{P}}_{ds}$ contains indices of paths in set $\widehat{\mathscr{P}}_{ds}$. We consider two versions of problem LRRDT:

- *full* problem LRRDT-FULL with the set of candidate paths for each demand d equal to $\widehat{\mathscr{P}}_d$
- *limited* problem LRRDT-LIMITED with the set of candidate paths for each demand d equal to $\mathscr{P}_d$, where $\mathscr{P}_d \subseteq \widehat{\mathscr{P}}_d$ for all d, and $\mathscr{P}_d \subsetneq \widehat{\mathscr{P}}_d$ for at least one demand d

Proposition 5.5. *Let $(t^*, \boldsymbol{f}^*, \boldsymbol{x}^*, \boldsymbol{y}^* \; ; \pi^*, \boldsymbol{\beta}^*, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*, \gamma^*)$ be a saddle point of the Lagrangian (5.105) for an instance of the limited problem LRRDT-LIMITED with a given set of candidate paths $\mathscr{P}_d$ for each demand* d. *Let $\wp_{ds} \in \widehat{\mathscr{P}}_{ds} \setminus \mathscr{P}_d$ be a shortest path for demand* d *in situation s and let ℓ_{ds} denote its length (5.126). Then:*

- *If $\mathcal{L}_{ds} = \ell_{ds}$ for all $d \in \mathcal{D}$, $s \in \mathcal{S}$ then any optimal solution to LRRDT-LIMITED is also optimal for LRRDT-FULL (the solution is optimal in a wider sense). Notice that this is the case when problem LRRDT-FULL has no paths shorter than LRRDT-LIMITED.*
- *Otherwise, if $\mathcal{L}_{ds} > \ell_{ds}$ for at least one demand* d *and situation s, then adding path $\wp_{ds}$ into $\mathscr{P}_d$ can possibly improve the current optimal solution.*

The proof for this proposition is analogous to proof of Proposition 5.1, just assuming that we perturb the problem LRRDT-LIMITED for some demand in only one situation, i.e., we route flow ε for some demand $\hat{d}$ on a shortest path (not in a list of candidate paths for the demand) only in some situation $\hat{s}$. Further reasoning and derivations are analogous to those in the proof of Proposition 5.1.

We will now present an algorithm for resolving problem LRRNDT. It is based on the improved algorithm for resolving problem RDT (see Section 5.3.5.4), but extends the algorithm by the path generation performed for all demands in blocking situations. Thus the employed PG strategy is add-all for blocking situations.

5.4.1 Algorithm for resolving LRRDT using PG

This iterative PG algorithm is based on the algorithm presented in Section 5.3.5.4. The main difference is that Step 2 includes a special (extended) test, based on the values of dual variables, checking the optimality of solution according to Proposition 5.5.

Resolution algorithm for LRRDT using PG.

Step 0 (Initialization):
Put $n := 0$, $Z_0 := \emptyset$, $Z_{PG} := \emptyset$, $Z_1 := \{1, 2, \ldots, S\}$, $t_s := 0$ for all s. Form the initial instance of the LRRDT problem, using small sets of candidate paths.

Step 1:
Solve LRRDT problem (If Step 1 is entered from Step 3, then use the optimal solution from the previous iteration as a staring point for solving the new instance of the problem).

Let $(t^*,\ \boldsymbol{f}^*,\ \boldsymbol{x}^*,\ \boldsymbol{y}^*\ ;\ \pi^*, \boldsymbol{\beta}^*,\ \boldsymbol{\lambda}^*,\ \boldsymbol{\sigma}^*,\ \boldsymbol{\tau}^*,\ \gamma^*)$ be a saddle point of the Lagrangian for the problem. Get values of dual variables from an LP solver.

Step 2:
Run the following test :

- Put $Z := \{s \in Z_1 : \lambda^*_s > 0\}$ and $t_s := t^*$ for each $s \in Z$. Put $Z_1 := Z_1 \setminus Z$.
- For each $s' \in Z$ and each demand d run a shortest path algorithm, using $\boldsymbol{\gamma}^*$ as links' costs, to find a shortest path $\wp_{ds'}$ and its cost $\ell_{ds'}$. If there exists such a path $\wp_{ds'}$ that $\ell_{ds'} < \mathcal{L}_{ds'}$ and the same path as $\wp_{ds'}$ has not been included yet to $\mathscr{P}_d$ for another situation $s' \in Z$, then include it to $\mathscr{P}_d$. If the path was already included while considering other s', skip it, but in Step 3 consider the current situation $s^{'}$, as if the new path was included for it.

Step 3:
Put $n := n + 1$, $Z_{PG} := \{s \in Z\ :\ \text{new paths } \wp_{ds}\ \text{ for s have been added}\}$,

$Z_1 := Z_1 \cup Z_{PG}$, $Z_0 := Z_0 \cup (Z \setminus Z_{PG})$ and form a new instance of the problem with the extended lists of available paths. **Go to Step 1**.

The entities used in the algorithm are analogous to those presented in section 5.3.5, except for the sets Z and Z_{PG}. Set Z is used to store blocking situations in each iteration, for which it is not known yet if shorter paths for some demands exist. Set Z_{PG}, in each iteration, is used to store situations for which a new path was added to a routing lists of at least one demand.

The algorithm is essentially the same as the improved algorithm for RDT presented before. However, in the algorithm with PG blocking situations are handled differently. We have mentioned in the previous section while studying properties of the dual problem that we can use effectively dual link costs only for the blocking situations. Therefore, in each blocking situation we look for a shortest path for each demand with respect to dual costs γ^*, and if a path with length $\ell_{ds} < \mathcal{L}_{ds}$ is found we include it into the candidate path list. Then, each such blocking situation, for which a new path was added to a routing list of at least one demand, is moved back to a set of non-blocking situations and the problem is resolved.

The path generation procedure is polynomial both for the single and multiple link failure cases, and decomposes into finding a shortest path for each demand in each situation.

CHAPTER 6

Fair bandwidth sharing in resilient two-layer networks

6.1 Introduction

So far we have discussed models for fair allocation in resilient single layer networks and presented two algorithms for resolving the associated network design problems. We will now extend the models still further and consider the networks with multiple layers of resources. As discussed in the introduction, as for now the architecture of backbone networks evolves towards two-layer IP over DWDM model. Therefore mathematical models and resolution algorithms for the design problems to be presented in this section are for two-layer networks. However, both the models and the algorithms are quite general and can be easily extended to account for more resource layers.

Thus, let us consider a resilient two-layer (IP over DWDM) core network model. It is assumed that demand volumes between source-destination (S-D) pairs are imposed on the packet layer by mixed elastic and non-elastic IP traffic streams. The elastic traffic streams can consume any assigned bandwidth within certain bounds. Flows (bandwidth allocated to different demands' paths) in both upper (packet) layer and lower (optical) layer are potentially reconfigurable in a coordinated way in case of failures. As in the previously presented single layer models for problem RDT, we will assume that bandwidth allocation (among the flows realizing demands) follows the PF rule in each considered (predefined) failure situation. Similarly, we will assume that revenues for the individual situations are forced to obey the MMF principle, thus, again, arriving at the multi-criteria optimization case. Flows assigned to demands' paths will be subject to maximization under a given budget constraint with respect to lower/upper bounds for each of demand volumes. Since an uncapacitated network design problems are considered, optimal link capacities are found as well.

Nodes and links of the lower layer are subject to failures. Since we are dealing with a multi-layer network, there are several possibilities for performing the flow recovery after a failure. We will present three models, differing by the layer where the recovery takes place: for flow reconfiguration only in the lower layer, only in the upper layer, and in both layers simultaneously. We will start by introducing the network model for the normal state only and will then proceed to discussion of the models assuring resiliency.

6.2 Two-layer network design for the normal state

Consider the problem of designing the two-layer network (e.g., IP/MPLS over DWDM), given a budget B for installing lower layer links and capacity unit costs on these links. A mix of non-elastic and elastic (with specified bounds on bandwidth) demands is imposed on the upper layer. It is assumed that bandwidth between the elastic demands is shared according to the PF principle. For demands in both layers lists with candidate paths are predefined. Our purpose is to find capacities of links and flow allocation in both layers, while the revenue function (6.1) must be maximized. In this model we are neglecting all the possible failures and are interested only in designing the network for the normal state. The design problem can be stated mathematically as follows.

MNOM:

$$\max \quad R = \sum_d w_d \log(X_d) \tag{6.1}$$

$$\text{s.t.:} \quad \sum_g \xi_g u_g \ \le \ B \ , \tag{6.2}$$

$$\sum_d \sum_p \delta_{edp} x_{dp} \ \le y_e \ , \qquad \forall e \in \mathcal{E} \tag{6.3}$$

$$\sum_p x_{dp} \ = X_d \ , \qquad \forall d \in \mathcal{D} \tag{6.4}$$

$$h_d \le X_d \le H_d \ , \qquad \forall d \in \mathcal{D} \tag{6.5}$$

$$\sum_k z_{ek} \ \ge \ y_e \ , \qquad \forall e \in \mathcal{E} \tag{6.6}$$

$$\sum_e \sum_k \varphi_{gek} z_{ek} \ \le \ u_g \ , \quad \forall g \in \mathcal{G} \tag{6.7}$$

$$x_{dp}, \ z_{ek} \in \mathbb{Z}^+ \qquad \forall d \in \mathcal{D}, \ p \in \mathcal{P}_d, \ e \in \mathcal{E}, \ k \in \mathcal{K}_e \tag{6.8}$$

$$y_e, \ u_g \in \mathbb{Z}^+ \qquad \forall e \in \mathcal{E}, \ g \in \mathcal{G}. \tag{6.9}$$

We will denote this problem by MNOM. Constraints (6.3), (6.4) and (6.5) are the same as constraints (5.4), (5.3) and (5.8) for single layer problem NDT (see

Section 5.1.1). They govern the flow allocation in the upper layer. Constraint (6.4) and variables X_d are auxiliary and are included in the formulation just for the clarity of presentation. The formulation becomes more compact if the left-hand side of (6.4) is directly used in the objective function and constraint (6.5). Similarly, constraints (6.3) and (6.6) can be merged, eliminating variables y_e and decreasing the number of constraints in the problem formulation.

The objective function (6.1) is also the same as for problem NDT, and implies PF allocation among flows in the upper layer. An allocation of flows in the upper layer implies upper layer link capacities y_e, which are then taken as capacity requirements for the lower layer. Constraint (6.6) assures that sums of the flows z_{ek} in the lower layer is sufficient for realizing the capacity requirements y_e. Constraints (6.7), similarly to (6.3), force the sums of all the flows of the lower layer (z_{ek}), that are routed on the paths traversing link g, not to exceed the available capacity (u_g) of link g. Budget constraint (6.2) assures that the cost of lower layer links does not exceed the budget B.

As for the single layer problems discussed in the previous chapters, the link capacities y_e and u_g should be assumed modular. This implies modifying constraints (6.3) and (6.7) in the following way:

$$\sum_d \sum_d \delta_{edp} x_{dp} \;\leq\; y_e M \;, \qquad \forall e \in \mathcal{E} \;, \tag{6.10}$$

$$\sum_e \sum_k \varphi_{gek} z_{ek} \;\leq\; u_g N \;, \qquad \forall g \in \mathcal{G} \;, \tag{6.11}$$

where:

$y_e \in \mathbb{Z}^+ \;, \; \forall e \in \mathcal{E}$	number of capacity modules installed on link e;
$u_g \in \mathbb{Z}^+ \;, \; \forall g \in \mathcal{G}$	number of capacity modules installed on link g;
M	size of the link capacity module in the upper layer;
N	size (in M-size units) of the link capacity module in the lower layer.

Note, that assuming modular link capacities changes the meaning not only of variables y_e, u_g, but also of flows z_{ek}, which are now calculated in M-size units, as variables y_e.

Resolving the problem MNOM will give the flow allocations (implying the selection of paths for flows) and link capacities in both layers. For example, consider the multi-layer routing scenario for normal network case depicted in Figure 6.1 (a somewhat similar scenario was also discussed in Section 3.2.2). Assume that the path chosen in the upper layer for flow realizing demand $a - d$ is $a - e - d$,

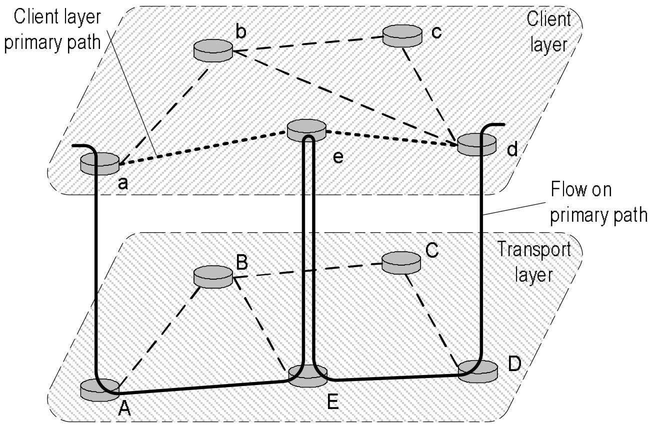


Figure 6.1: Routing in a two-layer network.

and all the demand volume is carried by this flow. The routing of all demands in the upper layer implies (total) capacity requirements on the upper layer links. These capacity requirements have to be realized by the flows in the lower layer. Say that the capacity of the upper layer link $a - e$ is realized by the flow $A - E$ in the lower layer, and similarly the capacity of $e - d$—by the flow $E - D$. Thus the actual flow for demand $a - d$ in this network is $a - A - E - e - E - D - d$. Similarly, all the flows in the lower layer imply the lower layer link loads that must not exceed the lower layer capacities.

The model presented above does not take failures into account. We will now extend the presented two-layer network model to account for failures occurring in the lower layer, as illustrated in Figure 6.2. Since links in the upper layer are virtual (their capacities are realized by the flows in the lower layer), failures of the upper layer links are not considered explicitly. Below we present three network design models implying flow recovery in case of failures. The models differ by which layer do the recovery actions take place in.

6.3 Problem RLL: flow Reconfiguration in Lower Layer

6.3.1 Recovery mechanism

The following model assumes that in a case of failure, recovery actions are taken only in the lower layer. The recovery from failures in the network consist in rerouting flows in the lower layer realizing the capacity of the upper layer links.

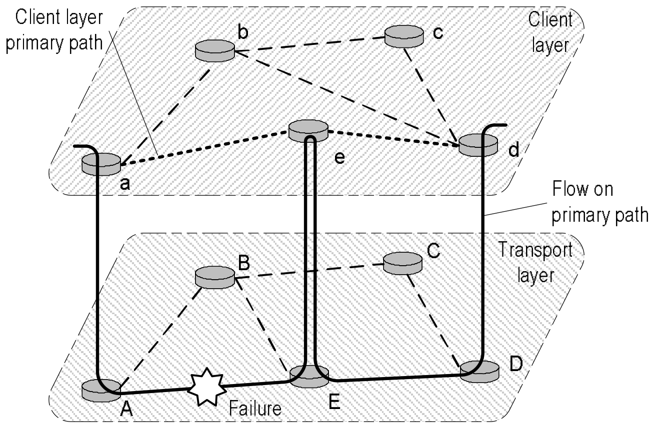


Figure 6.2: Failure in the lower layer.

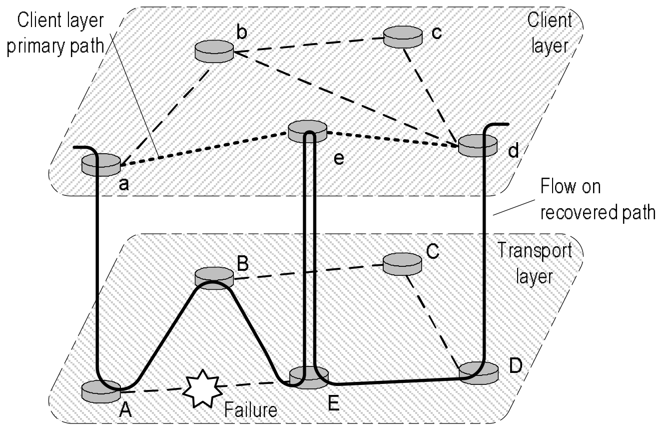


Figure 6.3: Recovery in the lower layer only.

Thus, recovery from the failure illustrated in Figure 6.2 will, for example, result in rerouting flow $A-E$ of the lower layer on a new path $A-B-E$. The actual flow for demand $a-d$ will in this state follow the path $a-A-B-E-e-E-D-d$. This recovery scenario is illustrated in Figure 6.3. Flows in the upper layer are in all the operation states routed on the same paths, while paths for flows in the lower layer are situation-dependent. We assume the unrestricted reconfiguration mechanism for recovery in the lower layer, which, essentially, implies, that when a failure occurs, all lower layer flows are disconnected and reconnected according to a pre-planned scheme for this particular failure case, using network resources that survive the failure. The reconfiguration actions (i.e., selection of paths and flows on them) for each of the network operation states are pre-planed at the network design stage and result from the optimization model presented below.

6.3.2 Mathematical formulation

The two-layer network design problem with recovery in lower layer only is defined as follows. We will refer to this problem as RLL.

RLL:

$$\max \quad R = \sum_d w_d \log(X_d) \tag{6.12}$$

$$\text{s.t.:} \quad \sum_g \xi_g u_g \ \le \ B \ , \tag{6.13}$$

$$\sum_d \sum_p \delta_{edp} x_{dp} \ \le y_e \ , \qquad \forall e \in \mathcal{E} \tag{6.14}$$

$$\sum_p x_{dp} \ = X_d \ , \qquad \forall d \in \mathcal{D} \tag{6.15}$$

$$h_d \le X_d \le H_d \ , \qquad \forall d \in \mathcal{D} \tag{6.16}$$

$$\sum_k z_{eks} \ \ge \ y_e \ , \qquad \forall e \in \mathcal{E}, \ \forall s \in \mathcal{S} \tag{6.17}$$

$$\sum_e \sum_k \varphi_{gek} z_{eks} \ \le \ \alpha_{gs} u_g \ , \quad \forall g \in \mathcal{G}, \ \forall s \in \mathcal{S} \tag{6.18}$$

$$x_{dp} \in \mathbb{Z}^+ \ , \qquad \forall d \in \mathcal{D}, \ p \in \mathcal{P}_d \tag{6.19}$$

$$z_{eks} \in \mathbb{Z}^+ \ , \qquad \forall e \in \mathcal{E}, \ \forall k \in \mathcal{K}_e, \ \forall s \in \mathcal{S} \tag{6.20}$$

$$y_e, \ u_g \in \mathbb{Z}^+ \ , \qquad \forall e \in \mathcal{E}, \ \forall g \in \mathcal{G}. \tag{6.21}$$

Constraints (6.17) assure that sums of the flows of the lower layer (z_{eks}) are sufficient to implement the capacity requirements y_e in all the predefined failure situations. Thus, for any upper layer link e, the capacity y_e is available in any of the predefined failure situations. This is assured by performing flow reconfiguration in the lower layer. Constraints (6.18), similarly to (6.14), force the sums of all the flows of the lower layer (z_{eks}), that are routed on the paths traversing link g, not to exceed the available (remaining) capacity ($\alpha_{gs} u_g$) of link g in situation s. Constraints (6.13)-(6.16) and the objective function (6.12) are the same as for problem MNOM, and have already been discussed before. Like for problem MNOM, constraint (6.15) and variables X_d are auxiliary and are included in the formulation just for the clarity of presentation. The formulation can be made more compact by substituting the left-hand side of (6.15) into the objective function and constraint (6.16), as well as merging constraints (6.14) and (6.17).

Capacities of links in both layers can be assumed modular. The modularity requirement is implied replacing constraint (6.14) by (6.10), and constraint (6.18) by the following:

$$\sum_e \sum_k \varphi_{gek} z_{eks} \le \alpha_{gs} u_g N , \qquad \forall g \in \mathcal{G} , \forall s \in \mathcal{S}. \tag{6.22}$$

Recall that introduction of modular link capacities changes the meaning of variables y_e, u_g and z_{eks}. Flows z_{eks} are then calculated in M-size units, like capacities y_e.

6.4 Problem RUL: flow Reconfiguration in Upper Layer

6.4.1 Recovery mechanism

As opposed to problem RLL, in this case all the recovery actions take place only in the upper layer. When a failure affects lower layer link it is the upper layer which takes all the responsibility of recovering from the failure, while the lower layer takes no action. Thus, flows in the upper layer are rerouted on alternative paths, not affected by the failure. For example, for the failure situation illustrated in Figure 6.2, the possible recovery scenario is shown in Figure 6.4. In this example the upper layer flow for demand $a - d$ is rerouted on a path $a - b - d$. Since the capacity of link $a - b$ is realized by flow $A - B$ in the lower layer, and the capacity of link $b - d$ is realized by the flow $B - E - D$, the actual flow in the network becomes $a - A - B - b - B - E - D - d$. The changed routing in the upper layer implies also different load on the lower layer links and the actual flow in the network. However, the lower layer flows realizing the capacities of the upper layer links remain the same in all the states of operation. These flows are dimensioned to account for the highest bandwidth requirement for realizing the (varying) upper layer capacities in any of the operation states. The unrestricted reconfiguration mechanism for recovery in the upper layer is assumed, implying, that when a failure in the lower layer occurs, all upper layer flows are disconnected and reconnected according to a pre-planned scheme for this particular failure case, using network resources that survive the failure. The selection of the recovery paths and flows on them are decided at the network design stage by resolving the problem presented below.

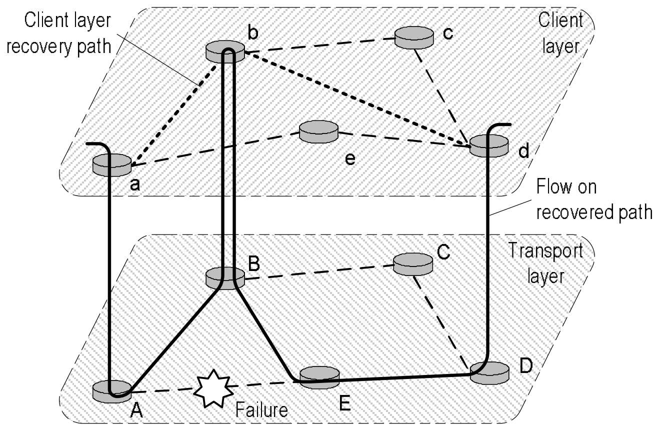


Figure 6.4: Recovery in the upper layer only.

6.4.2 Mathematical formulation

This formulation assumes flow reconfiguration only in the upper layer (we will refer to the problem presented here as RUL). RUL uses lexicographical maximization. Therefore we will define problem RUL similarly to problem RDT (see Section 5.3.3). Suppose $\bar{X}$ is the set of all (feasible) vectors $\boldsymbol{X} = \{X_{ds} : d = 1, 2, \ldots, D, s = 1, 2, \ldots, S\}$ defined by the following constraints:

$$\sum_g \xi_g u_g \ \le\ B\ , \tag{6.23}$$

$$\sum_d \sum_p \delta_{edp} x_{dps} \ \le\ y_{es}\ , \qquad \forall e \in \mathcal{E}\ , \forall s \in \mathcal{S} \tag{6.24}$$

$$\sum_p x_{dps} \ =\ X_{ds}\ , \qquad \forall d \in \mathcal{D}\ , \forall s \in \mathcal{S} \tag{6.25}$$

$$h_{ds} \le X_{ds} \ \le\ H_{ds}\ , \qquad \forall d \in \mathcal{D}\ , \forall s \in \mathcal{S} \tag{6.26}$$

$$\sum_k \theta_{eks} z_{ek} \ \ge\ y_{es}\ , \qquad \forall e \in \mathcal{E}\ , \forall s \in \mathcal{S} \tag{6.27}$$

$$\sum_e \sum_k \varphi_{gek} z_{ek} \ \le\ u_g\ , \qquad \forall g \in \mathcal{G} \tag{6.28}$$

$$x_{dps} \in \mathbb{Z}^+\ , \qquad \forall d \in \mathcal{D},\ p \in \mathcal{P}_d,\ \forall s \in \mathcal{S} \tag{6.29}$$

$$z_{ek} \in \mathbb{Z}^+\ , \qquad \forall e \in \mathcal{E},\ \forall k \in \mathcal{K}_e \tag{6.30}$$

$$y_{es},\ u_g \in \mathbb{Z}^+\ , \qquad \forall e \in \mathcal{E},\ \forall s \in \mathcal{S},\ \forall g \in \mathcal{G}. \tag{6.31}$$

For each $\boldsymbol{X} \in \bar{X}$ let $\boldsymbol{R}(\boldsymbol{X}) = (R_1(\boldsymbol{X}), R_2(\boldsymbol{X}), \ldots, R_S(\boldsymbol{X})$ denote the vector of revenues,

$$R_s(\boldsymbol{X}) \;=\; \sum_d w_{ds} \log(X_{ds}) \;\;,\;\; \forall s \in \mathcal{S}, \tag{6.32}$$

sorted in non-decreasing order. The reader can recognize that function (6.32) implying PF allocation among aggregated demand flows in each state is the same as the corresponding function (5.56) for problem RDT-MMF-PF (see Section 5.3.3). Then the problem RUL is formulated as follows:

$$\textbf{RUL:} \qquad \text{lex max } \{\boldsymbol{R}(\boldsymbol{X}) : \boldsymbol{X} \in \bar{X}\}. \tag{6.33}$$

The lexicographical maximization objective function for the problem, which is the same as for problem RDT, assures that the problem RUL results in lexicographically maximal (unique) solution vector of revenues; this implies the MMF allocation of revenues among situations. In each failure situation the total flows are allocated in a proportionally fair way among the demands due to (6.32). These aggregated flows for each demand in each situation are given in (6.25) and are forced to attain values within the assumed bounds by constraints (6.26). Note that constraint (6.25) and variables X_{ds} are auxiliary and can be excluded from the problem by directly using the left-hand side of constraint (6.25) in (6.32) and (6.26). Constraints (6.24) force the sums of all the upper layer flows (x_{dps}) that are routed on paths traversing link e, to be equal to the allocated capacity for link e in the situation s. Constraints (6.27) assure that the total available flows of the lower layer ($\theta_{eks} z_{ek}$) on the remaining working paths are sufficient to implement y_{es} in each of the failure situations. Recall, that θ_{eks} is an availability coefficient of lower layer path k realizing capacity of upper layer link e in situation s. Constraints (6.24) and (6.27) can be combined in order to make the formulation more compact. Constraints (6.28), similarly to (6.24), force the sums of all the flows of the lower layer (z_{ek}), that are routed on paths traversing link g, not to exceed the capacity allocated for link g.

The problem formulation can be modified to take modular link capacities into account by replacing constraint (6.28) with (6.11), and replacing constraint (6.24) with the following:

$$\sum_d \sum_d \delta_{edp} x_{dps} \;\le\; y_{es} M \;, \qquad \forall e \in \mathcal{E} \;, \forall s \in \mathcal{S}. \tag{6.34}$$

As it was discussed before, the introduction of modular link capacities imply that the meaning of variables y_{es}, u_g and z_{ek} changes. In this case, flows z_{ek} are calculated in M-size units, like capacities y_{es}.

6.5 Problem RBL: flow Reconfiguration in Both Layers

6.5.1 Recovery mechanism

The last recovery option considered here is the coordinated recovery in both layers. When a failure occurs in the lower layer, flows are rerouted (in a coordinated way) in both the lower and the upper layer in order to circumvent the failure. The possible recovery scenario for the failure situation illustrated in Figure 6.2 may be as shown in Figure 6.5. In this case the actual recovered flow for demand $a-d$ becomes $a-B-b-B-C-D-d$. Another, probably more obvious, example of the coordinated recover in both layers, could be for the case when two simultaneous failures affect the nominal path. A possible outcome of the coordinated recovery actions could then be that flow is recovered from one of the failures by rerouting in one layer and from the second failure–by rerouting in the other layer. In general, flows may be rerouted on alternative paths in both layers in order to circumvent the failure. We assume the unrestricted reconfiguration mechanism for recovery in both layers. Thus in the worst case when a failure occurs all lower and upper layer flows are disconnected and reconnected according to a pre-planned scheme for this particular failure case, using network resources that survive the failure. The selection of the recovery paths and flows on them are decided at the network design stage by resolving the problem presented below.

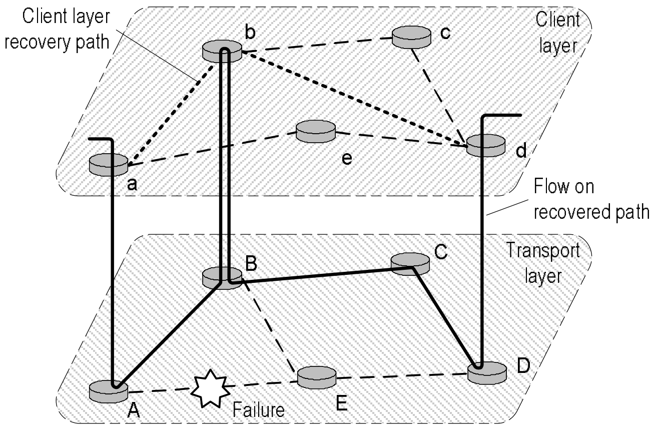


Figure 6.5: Coordinated recovery in both layers.

6.5.2 Mathematical formulation

The following final formulation assumes flow reconfiguration in the upper and lower layers simultaneously (we will refer to this network design problem as RBL). It is the most flexible flow reconfiguration option, but also the most complicated one. Like RUL, it uses lexicographical maximization, therefore we will define it similarly to RUL. The two problems differ only by the few constraints defining the feasibility region $\bar{X}$. For the problem RBL, $\bar{X}$ is defined by the following constraints:

$$\sum_g \xi_g u_g \;\le\; B\,, \tag{6.35}$$

$$\sum_d \sum_p \delta_{edp} x_{dps} \;\le\; y_{es}\,, \qquad \forall e \in \mathcal{E}\,, \forall s \in \mathcal{S} \tag{6.36}$$

$$\sum_p x_{dps} \;=\; X_{ds}\,, \qquad \forall d \in \mathcal{D}\,, \forall s \in \mathcal{S} \tag{6.37}$$

$$h_{ds} \le X_{ds} \;\le\; H_{ds}\,, \qquad \forall d \in \mathcal{D}\,, \forall s \in \mathcal{S} \tag{6.38}$$

$$\sum_k z_{eks} \;\ge\; y_{es}\,, \qquad \forall e \in \mathcal{E}\,, \forall s \in \mathcal{S} \tag{6.39}$$

$$\sum_e \sum_k \varphi_{gek} z_{eks} \;\le\; \alpha_{gs} u_g\,, \qquad \forall g \in \mathcal{G}\,, \forall s \in \mathcal{S} \tag{6.40}$$

$$x_{dps} \in \mathbb{Z}^+\,, \qquad \forall d \in \mathcal{D},\; p \in \mathcal{P}_d,\; \forall s \in \mathcal{S} \tag{6.41}$$

$$z_{eks} \in \mathbb{Z}^+\,, \qquad \forall e \in \mathcal{E},\; \forall k \in \mathcal{K}_e,\; \forall s \in \mathcal{S} \tag{6.42}$$

$$y_{es},\; u_g \in \mathbb{Z}^+\,, \qquad \forall e \in \mathcal{E},\; \forall s \in \mathcal{S},\; \forall g \in \mathcal{G}. \tag{6.43}$$

For each $\boldsymbol{X} \in \bar{X}$ let $\boldsymbol{R}(\boldsymbol{X}) = (R_1(\boldsymbol{X}), R_2(\boldsymbol{X}), \ldots, R_S(\boldsymbol{X})$ denote the vector of revenues (6.32) sorted in non-decreasing order. Then the problem RBL is formulated as follows:

$$\textbf{RBL:} \qquad \text{lex max } \{\boldsymbol{R}(\boldsymbol{X}) : \boldsymbol{X} \in \bar{X}\}. \tag{6.44}$$

Most of the constraints are analogous to those of problem RUL, except (6.39)-(6.40). Constraints (6.39) assure that sums of the lower layer flows (z_{eks}) are sufficient to implement capacities y_{es} in each failure situation. Constraints (6.40) force the sums of the lower layer flows (z_{eks}), that are routed on paths traversing link g, not to exceed the available capacity of link g in situation s ($\alpha_{gs} u_g$). As was noted for the other two-layer problems, the formulation can be made more compact by excluding the auxiliary variables X_{ds} as well as constraints (6.37), and using the left-hand side of (6.37) directly in (6.32) and (6.38). Also, constraints (6.36) and (6.39) can be merged.

Modular link capacities can be introduced by replacing constraint (6.36) by (6.34) and (6.40) by (6.22). As it was discussed before, the introduction of modular link capacities imply that the meaning of variables y_{es}, u_g and z_{eks} changes. In this case, flows z_{eks} are calculated in M-size units, like capacities y_{es}.

6.6 Algorithms for resolving the two-layer problems

All the three two-layer problems presented above are non-linear due to the logarithmic function in (6.32). They can be linearized by applying the piece-wise linear approximation of the logarithmic function discussed in section 4.3.4. After the approximation of the logarithmic function, problem RLL can be treated as a LP problem, and as such can be directly solved by a LP solver, such as CPLEX [49].

Similarly to RDT, problems RUL and RBL are not mathematical programming problem (since they use lexicographical order maximization) and must be resolved in a special way, e.g., by the algorithms analogous to those presented in Section 5.3.5. Below we present an algorithm for resolving the linearized and relaxed problems RUL and RBL, i.e., with variables $x_{dps}, z_{ek}, z_{eks}, y_{es}, u_g \in \mathbb{R}^+$. The algorithm is identical with the improved algorithm for resolving problem RDT (presented in Section 5.3.5.4), with only difference being that the set of feasible solutions for the optimization problem solved in Step 1 is in this case defined by the constraints of problems RUL and RBL, respectively. For clarity below we restate the full algorithm, with adjusted constraint set in the problem of Step 1.

6.6.1 Algorithm for resolving RUL and RBL

Algorithm for resolving RUL and RBL

Step 0 (Initialization):
Put $n := 0$, $Z_0 := \emptyset$, $Z_1 := \{1, 2, \ldots, S\}$, $t_s := 0$ for all s.

Step 1:
Solve the following convex programme:

maximize t
subject to (6.23), (6.24)-(6.26);
(6.27)-(6.28) for RUL or (6.39)-(6.40) for RBL; and

$$R_s = \sum_d w_{ds} \log(X_{ds}) = t_s \ , \ s \in Z_0 \quad (6.45)$$

$$R_s = \sum_d w_{ds} \log(X_{ds}) \geq t \ \ , \ s \in Z_1. \quad (6.46)$$

Let t^* be the optimal solution of the above problem and $\lambda_s^*, s \in Z_1$ be the optimal dual variables corresponding to constraints (6.46).

Step 2:
Put $n := n+1$ and $t_s := t^0$ for each $s \in Z_1$. Put $Z_0 := Z_0 \cup \{s \in Z_1 : \lambda_s^0 > 0\}$ and $Z_1 := \{s \in Z_1 : \lambda_s^0 = 0\}$.

If $Z_1 = \emptyset$ then STOP (the vector $\boldsymbol{R} = (R_1, R_2, \ldots, R_S) = (t_1, t_2, \ldots, t_S)$, sorted in non-decreasing order, is the solution of the problem); else go to Step 1.

6.7 Numerical experiments for the two-layer models

A number of experiments have been performed with two sets of network models. The first set of experiments considers networks with symmetrical demands, i.e., when lower bounds h_d and upper bounds H_d on demand volume are the same for all demands. The second set of experiments considers network instances with unsymmetrical demands. For the second set of experiments two-layer network instances have been produced from real network instances available from SNDlib library [118].

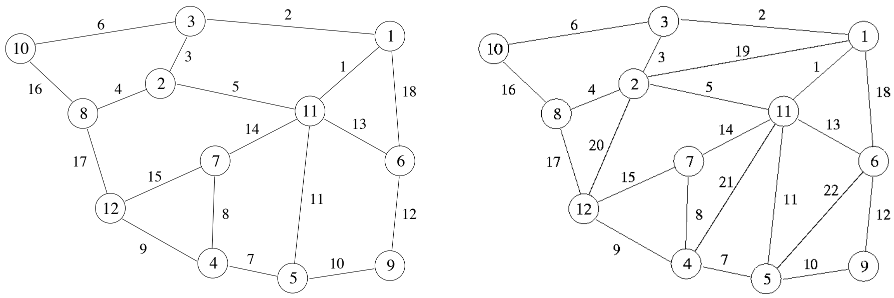
6.7.1 Example networks with symmetrical demands

Two numerical examples for two different size networks – mid-size (N_{12}) and large (N_{41}) – are presented in this section. The models are summarized in Table 6.1 and the network topologies of both layers are shown in Figures 6.6-6.7. Topologies of the lower layer networks (indicated as L1 in Table 6.1) for both N_{12} and N_{41} are the same as for the corresponding networks used in the numerical experiments for problem RDT (see Section 5.3.7). Thus the costs for the lower layer links for N_{12} and N_{41} are also the same as those considered for problem RDT and are given in Tables 5.5 and 5.7, respectively. Note that in the tables links are indexed by e, since the networks were considered in the single layer scenario. In the case of the two-layer networks presented here the link index in Tables 5.5 and 5.7 should be substituted by g and the link cost by ξ_g. Topologies of the upper layers were produced by extending the topologies of the lower layers.

Failure situations have been generated according to the following rule: in situation $s = 1$ (called the nominal situation) all links are fully available. In each of

ref. code	layer	# nodes	# links	# paths per demand	# demands	# failure situations
N_{12}	L_2	12	22	6-14	66	-
	L_1	12	18	2-3	22	19
N_{41}	L_2	21	37	6	209	-
	L_1	41	72	3	37	22

Table 6.1: Networks with symmetrical demands used for experiments.



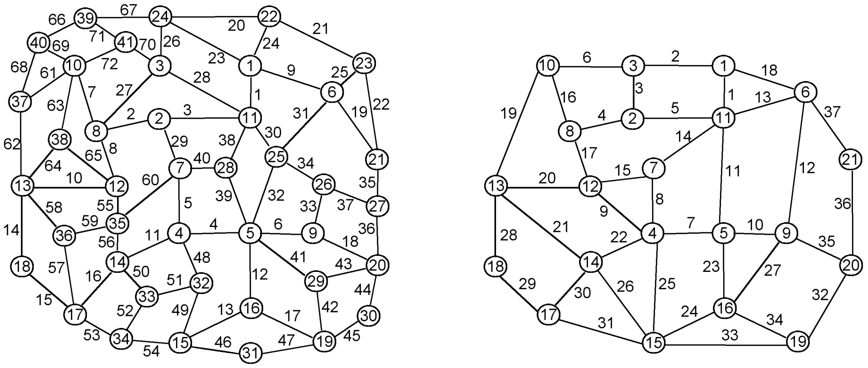
(a) Lower layer (b) Upper layer

Figure 6.6: Topology of layers for network N_{12}.

the remaining situations two randomly selected links are assumed to fail entirely, so that their link availability coefficients α_{gs} become equal to 0 (coefficients for the remaining links are equal to 1). It has been assured that the situations are unique, and that they do not result in disjoint graphs. The pairs of links that fail in each situation are given in Tables 6.3 and 6.4. The experiments have been performed with $S = 19$ situations for the network N_{12} and $S = 22$ situations for the network N_{41}. For all experiments all revenue coefficients w_{ds} have been set to 1 and budget B to 10^6.

6.7.2 Example networks with unsymmetrical demands

Two-layer networks used for this series of experiments were produced from real single-layer network instances available from SNDlib library [118]. Four networks have been used: *france, newyork, di-yuan* and *cost266*. The network models



(a) Lower layer (b) Upper layer

Figure 6.7: Topology of layers for network N_{41}.

g	1	2	3	4	5	6	7	8	9
ξ_g	1.85	3.4	1	1.45	2.3	2.9	1	1.6	2.2
g	10	11	12	13	14	15	16	17	18
ξ_g	1.5	2.3	1.4	1.65	1.25	2.3	1.55	1.35	1.7

Table 6.2: Link marginal costs for network N_{12}.

s	1	2	3	4	5	6	7	8	9	10
g	-	5,16	8,12	6,15	3,8	11,15	6,11	3,5	2,17	4,14
s	11	12	13	14	15	16	17	18	19	
g	11,13	2,16	7.10	2,7	17,18	1,17	7,18	6,13	1,9	

Table 6.3: Links that fail in each situation for network N_{12}.

s	1	2	3	4	5	6	7	8	9	10	11
g	-	8,15	9,11	17,55	19,5	33,3	23,4	61,31	12,63	38,2	17,8
s	12	13	14	15	16	17	18	19	20	21	22
g	13,16	19,6	48,53	40,5	31,51	49,11	60,16	30,39	14,2	8,70	10,1

Table 6.4: Links that fail in each situation for network N_{41}.

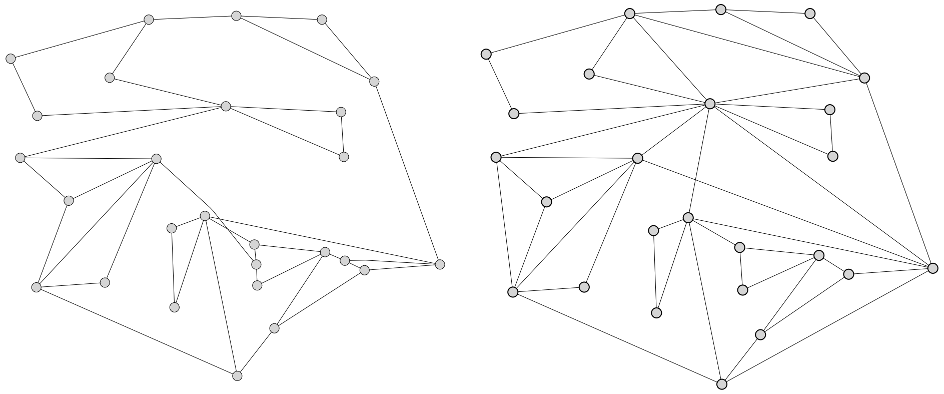
are summarized in Table 6.5 and the network topologies of both layers are shown in Figures 6.8-6.11. Statistical measures of the link costs ξ_g and demand volumes

h_d for all the networks are summarized in Table 6.6. The upper bound on demand volumes was in all cases assumed to be infinity and the budget B was set to 10^6. The upper layer networks for the two-layer instances are directly taken from the SNDLib library. Relation between upper and lower layer networks of any instance is that all upper layer nodes have to exist in the lower layer as well. Lower layer network can have more nodes, but often has less links than the corresponding upper layer network. The networks for lower layer L_1 were in all cases produced from the corresponding upper layer networks by the following modifications: a randomly generated number of links between 30 and 45% of links in L_2 topology were considered for removal. The particular links to remove were also chosen randomly, however, assuring that graph does not become disconnected, and nodes to which the links were connected still maintain degree of at least three after the removal. For some networks the actual number of removed links was less than the considered one due to the constraints described above. In the second step of the network transformation a number of new nodes were added, breaking one randomly chosen link each. The number of nodes to add was randomly chosen, but was upper bounded by 30% of the existing upper layer nodes. The degree for the new nodes was randomly chosen to be between 3 and 5. Each new node was first connected to end nodes of the link which was broken by placing the node, and then, between 1 and 3 (depending on the node degree) randomly chosen nodes in the network. In the final step of network transformation the capacity and cost of the links in the resulting network were scaled by a randomly chosen (for each link) factor which lies between 0.5 and 1.5. In this way a lower layer network was produced from the upper layer network.

In order to carry on the experiments with models developed in this section, demands were assumed to be undirected. Candidate path lists for the demands in networks of both layers were generated, containing a chosen number of shortest paths for each demand. In the experiments with the four networks single link failures were assumed. Thus, all the links are available in the nominal situation ($s = 0$), and in the subsequent situations a single link corresponding to a situation number is failed.

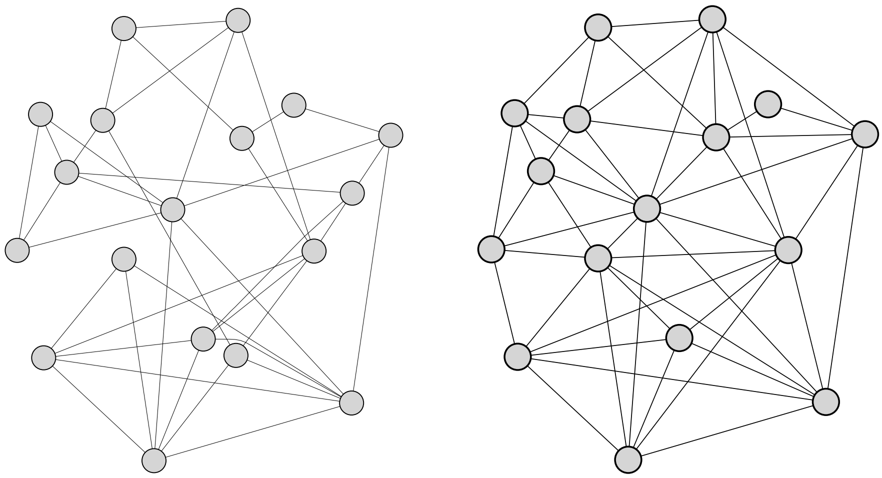
6.7.3 Numerical results

Resulting revenues of the three reconfiguration options have been compared in the unbounded (when X_d or X_{ds} could take any value between 0 and $+\infty$) and bounded (when X_d or X_{ds} could be assigned any values from the intervals $h_d \leq X_d \leq H_d$ or $h_{ds} \leq X_{ds} \leq H_{ds}$, respectively) cases. Two sets of network examples were considered- with symmetrical and unsymmetrical demand



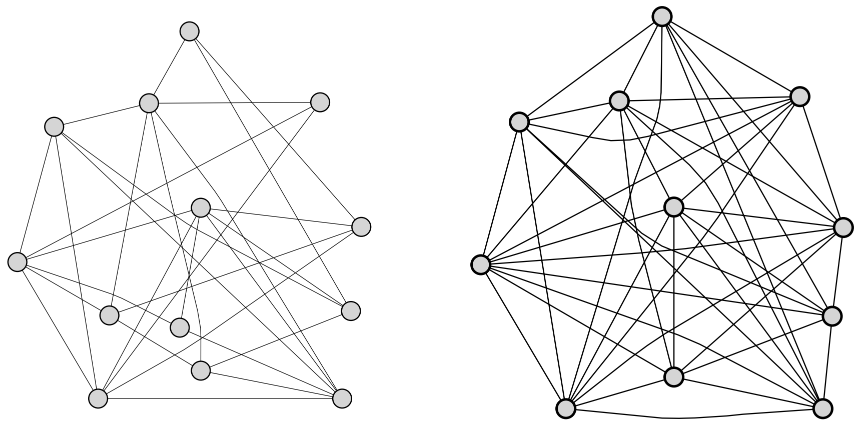
(a) Lower layer (b) Upper layer

Figure 6.8: Topology of layers for network *france*.



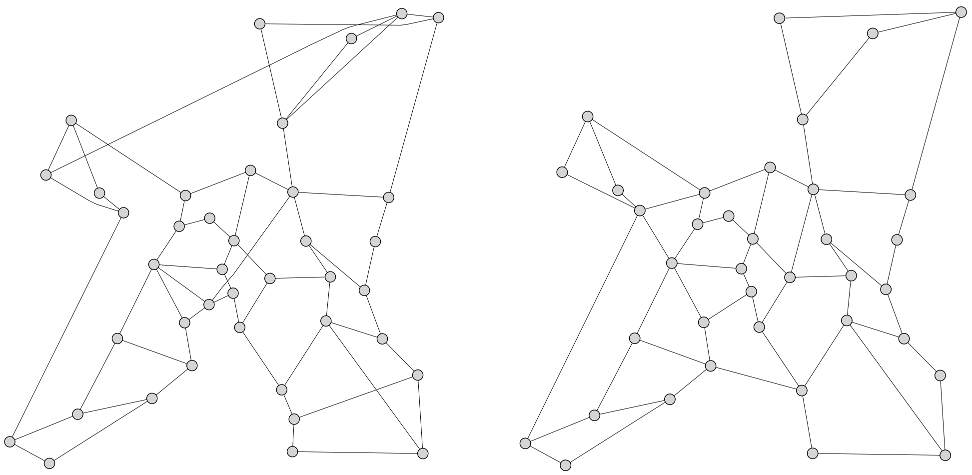
(a) Lower layer (b) Upper layer

Figure 6.9: Topology of layers for network *newyork*.



(a) Lower layer (b) Upper layer

Figure 6.10: Topology of layers for network *di-yuan*.



(a) Lower layer (b) Upper layer

Figure 6.11: Topology of layers for network *cost266*.

ref. code	layer	# nodes	# links	# paths per demand	# demands	# failure situations
france	L_2	25	45	2-10	300	-
	L_1	27	40	2-10	45	41
newyork	L_2	16	49	10	120	-
	L_1	18	39	10	49	40
di-yuan	L_2	11	42	10	22	-
	L_1	13	30	10	42	31
cost266	L_2	37	57	10	666	-
	L_1	40	61	10	57	62

Table 6.5: Networks with unsymmetrical demands used for experiments.

	Networks							
	france		*newyork*		*di-yuan*		*cost266*	
	ξ_g	h_d	ξ_g	h_d	ξ_g	h_d	ξ_g	h_d
Min	0.205	48	2.12	2	6.4	1	0.13	19
Max	5.7	1808	36.446	42	61.016	5	1.22	5626
Range	5.495	1760	34.325	40	54.616	4	1.09	5607
Var	1.502	$9.676e4$	51.483	44.588	199.138	1.682	0.058	$2.836e5$
Mean	1.178	332.767	12.207	7.5	27.745	2.409	0.439	510.209
P25	0.372	116	7.824	4	19.964	1	0.294	157
Median	0.869	213	10.83	5	24.065	2	0.373	358
P75	1.38	437	14.546	10	36.545	3	0.517	677
Sum	47.131	$9.983e4$	476.088	900	832.358	53	26.782	$3.397e5$

Table 6.6: Link costs and demand volumes for the unsymmetrical networks.

volumes (lower bounds). Imposing upper bound H_{ds} limits the highest value for the flows X_{ds} at a certain value, although, if allowed by the budget, it would be possible to increase it even more. In this way the resulting vector of revenues is lexicographically smaller than in the unbounded case. Imposing lower bound (LB) is of more interest, because it in general may imply a different flow allocation scheme. Therefore only the results for the unbounded case and the case with lower bounds ($LB = 1000$ for N_{12} and $LB = 10$ for N_{41}) are presented for the networks with symmetrical demands. For the networks with unsymmetrical demand volumes the LBs are implied. The upper bound in all experiments was set to $+\infty$.

First we will discuss the results for the networks with the symmetrical de-

mand volumes. Statistical characterization of the resulting revenues and lower layer capacities for the three reconfiguration options in the unbounded case is given in Tables 6.7 and 6.8. Each of the tables presents main statistical measures for revenues (R_s) and resulting link capacities (u_g) for the three reconfiguration options. The preferred values are bolded in the tables. Revenue in problem RLL is not situation-dependent, so for all situations it is the same and equal to 6007.04 (in the unbounded case) for network N_{12}, and 2072.19 for network N_{41}. Thus, all the entries in revenue column of the tables for option RLL are set to the same value, except of the entries for range and variance that are naturally equal to zero. To allow for comparison with RUL and RBL options, the sum of revenues for RLL option is calculated multiplying the RLL revenue by the number of situations. The same assumptions regarding presentation of results for problem RLL hold also for the other tables given below. Revenues of RUL and RBL are situation-dependent. Figures 6.12(a) and 6.12(b) illustrate lexicographically ordered revenue vectors for the three reconfiguration options in the unbounded case for the networks N_{12} and N_{41} respectively. Because of the lexicographical ordering the numberings of situations may not coincide for different reconfiguration options. Therefore the situations are not numbered in the figures. As it can be seen from the figures and tables, revenue vectors for RUL and RBL are almost the same for network N_{12}, while for N_{41} RBL is clearly better. It should be noted, that for network N_{12} the results for RUL and RBL look exactly the same (see Table 6.7) when rounded up to the three places after the decimal point, as is the case for results presented in the tables. The (negligible) differences between revenue vectors for RUL and RBL can be observed only when examining the twelfth position after the decimal point. Then it can be seen that the vector for RBL is marginally lexicographically bigger than for RUL:
$\boldsymbol{R}^{RBL} = \{12132.37887293740\mathbf{8}, 12132.378872937410,$
$12132.378872937414, \ldots, 15132.09314945\mathbf{4997}\} >$
$\boldsymbol{R}^{RUL} = \{12132.37887293740\mathbf{6}, 12132.378872937408,$
$12132.378872937410, \ldots, 15132.09314945\mathbf{5350}\}.$
It can be seen, that only because the smallest revenue values (in the lexicographical order) attained for RBL are higher than the ones for RUL, it makes RBL marginally better. But the maximal revenue achieved in the RUL case is higher, even for N_{41} network (see Table 6.8). This similarity of RUL and RBL for N_{12} can be explained by the very similar network topologies of the upper and lower layers. Network N_{41} with different layer topologies shows obvious superiority of RBL. Examining results in the Table 6.8 one can see that for network N_{41} all the best values (except of the maximal revenue value) were attained by RBL option.

Results for the bounded case are presented in Tables 6.9 and 6.10 and Figures 6.13(a) and 6.13(b). As it can be seen from the tables and figures for the bounded case ($LB = 1000$ for N_{12} and $LB = 10$ for N_{41}) the overall picture is the same as in the unbounded case. The lexicographically ordered revenue vector for problem RBL is again greater than the one for RUL. In this case, the difference is non-negligible for both N_{41} and N_{12} as it can be seen from the revenue values (for N_{12}) below:
$\boldsymbol{R}^{RBL} = \{9078.90, 9078.90, 9078.90, \ldots, 11592.75\} >$
$\boldsymbol{R}^{RUL} = \{9069.50, 9069.50, 9069.50, \ldots, 11581.00\}$.
In this case revenue values for RBL are significantly higher than for RUL, because the former has more reconfiguration capabilities (in both layers simultaneously) which are especially useful under the tight LB constraints. It can also be seen from the figures, that revenue values for the bounded case are, as expected, smaller than in the unbounded case.

Much higher values of revenues are achieved with RUL and RBL in comparison with RLL, because the first two algorithms involve lexicographical maximization of the revenues, while RLL has to maintain the same revenue in all situations. Higher revenue values also mean higher values of aggregated flows (X_{ds}), which are beneficial for elastic traffic networks. Besides, even a small difference in total logarithmic flows ($log(X_{ds})$) makes much bigger difference between the "plain" total flows X_{ds}. Figures showing total (non-logarithmic) values X_d for each failure situation have a similar character to that for revenues. It can be seen from the results that, in the considered cases, both RUL and RBL are almost equally good, as compared to RLL, although RBL performs better, especially for the networks with different topology of the layers (e.g. N_{41}) or with tight lower bounds. In almost all the cases for the networks with symmetrical demand volumes highest mean and lowest variance of revenue values were achieved by option RBL. Thus, for the networks considered, RBL not only results in lexicographically largest revenue vector, it also assures on average highest revenues in all situations and lowest variation of the revenues between situations.

Results for the networks with unsymmetrical demand volumes are presented in Tables 6.11-6.14. Similarly as for the networks with symmetrical demand volumes, each of the tables presents main statistical measures for revenues (R_s) and resulting link capacities (u_g) for the three reconfiguration options. The preferred values are bolded in the tables. The results are not presented by figures, but only in terms of statistical measures, because of the large number of considered failure situations, which makes it difficult to present the results graphically. Examining data in the tables we can see that results are similar to those discussed above for

	R_s			u_g		
	RLL	RUL	RBL	RLL	RUL	RBL
Min	$0.6e4$	$1.213e4$	$1.213e4$	145.666	68.063	68.063
Max	$0.6e4$	$1.513e4$	$1.513e4$	$1.609e5$	$1.662e5$	$1.662e5$
Range	0	2999.714	2999.714	$1.608e5$	$1.662e5$	$1.662e5$
Var	0	$2.347e6$	$2.347e6$	$4.706e9$	$5.841e9$	$5.841e9$
Mean	$0.6e4$	$1.355e4$	$1.355e4$	$3.607e4$	$4.622e4$	$4.622e4$
P25	$0.6e4$	$1.213e4$	$1.213e4$	318.47	142.486	142.486
Median	$0.6e4$	$1.214e4$	$1.214e4$	517.78	194.612	194.612
P75	$0.6e4$	$1.512e4$	$1.512e4$	723.45	$1.657e5$	$1.657e5$
Sum	$1.141e5$	$2.574e5$	$2.574e5$	$6.493e5$	$8.32e5$	$8.32e5$

Table 6.7: Revenue and capacity statistics for N_{12} network in the unbounded case.

	R_s			u_g		
	RLL	RUL	RBL	RLL	RUL	RBL
Min	2072.186	2141.233	**2144.83**	0	0	0
Max	2072.186	**2174.476**	2155.831	$8.756e4$	$8.602e4$	$8.648e4$
Range	0	33.242	**11.001**	$8.756e4$	$8.602e4$	$8.648e4$
Var	0	55.617	**13.927**	$1.061e8$	$1.023e8$	$1.035e8$
Mean	2072.186	2144.062	**2147.483**	1382.685	1385.349	1385.141
P25	2072.186	2141.233	**2144.83**	99.501	25.366	35.462
Median	2072.186	2141.233	**2144.832**	179.777	159.35	161.245
P75	2072.186	2141.665	**2151.462**	229.789	261.063	237.007
Sum	$4.559e4$	$4.717e4$	$\mathbf{4.724e4}$	$9.955e4$	$9.975e4$	$9.973e4$

Table 6.8: Revenue and capacity statistics for N_{41} network in the unbounded case.

	R_s			u_g		
	RLL	RUL	RBL	RLL	RUL	RBL
Min	4409.304	9069.499	**9078.899**	6500	7000	6000
Max	$0.441e4$	$1.158e4$	$\mathbf{1.159e4}$	$0.906e5$	$1.229e5$	$1.227e5$
Range	0	**2511.5**	2513.85	$0.841e5$	$1.159e5$	$1.167e5$
Var	0	$8.954e5$	$\mathbf{8.874e5}$	$0.877e9$	$1.911e9$	$1.91e9$
Mean	4409.304	9971.788	**9977.88**	$3.341e4$	$3.931e4$	$3.935e4$
P25	4409.304	9069.499	**9078.899**	$1.3e4$	$1.1e4$	$1.1e4$
Median	4409.304	**9855.499**	9834.899	$2.3e4$	$1.433e4$	$1.45e4$
P75	$0.441e4$	$\mathbf{1.094e4}$	$1.093e4$	$3.2e4$	$8.319e4$	$8.266e4$
Sum	$0.838e5$	$1.895e5$	$\mathbf{1.896e5}$	$6.014e5$	$7.076e5$	$7.083e5$

Table 6.9: Revenue and capacity statistics for N_{12} networks with $LB = 1000$.

	R_s			u_g		
	RLL	RUL	RBL	RLL	RUL	RBL
Min	2044.99	2123.746	**2131.857**	0	0	0
Max	2044.99	2160.291	**2164.76**	$8.431e4$	$8.308e4$	$8.381e4$
Range	0	36.545	**32.904**	$8.431e4$	$8.308e4$	$8.381e4$
Var	0	77.769	**55.042**	$9.825e7$	$9.538e7$	$9.709e7$
Mean	2044.99	2128.044	**2135.928**	1384.537	1387.393	1387.292
P25	2044.99	2123.746	**2131.857**	129.638	30	53.309
Median	2044.99	2123.746	**2131.857**	225.1	214.844	197.309
P75	2044.99	2128.918	**2139.192**	280.246	304.036	292.464
Sum	$4.499e4$	$4.682e4$	**$4.699e4$**	$9.969e4$	$9.989e4$	$9.989e4$

Table 6.10: Revenue and capacity statistics for N_{41} networks with $LB = 10$.

the network with symmetrical demand volumes. Among the three reconfiguration options the lexicographically largest vector of revenues was attained by RBL for all the networks. The difference between minimum revenue values, and the difference between maximum revenue values among RUL and RBL options were very small for *newyork* and *di-yuan* networks, while for the other two networks the differences were quite large. Even the highest revenue value for network *newyork* has been achieved by RUL option. The differences in revenue values between RUL and RBL for different networks depend greatly on the ratio between the available network capacities (implied by budget and link costs) and the total network load, implied by demand volumes to be realized. Among the considered four networks, much higher total demand volume had to be realized for networks *france* and *cost266* (see last row in Table 6.6) than for networks *newyork* and *di-yuan* for the same budget. Thus, the design problem instances for the first two networks were more tight than for the last two networks. Since the optimization model for RML is more constrained than for RBL, revenues achieved for RML were much smaller than for RBL in the case of *france* and *cost266* networks than for the other two networks.

As in the case of N_{12} and N_{41} networks, the revenue value achieved by RLL option for all the four networks was much lower than minimum revenue values for RUL and RBL. For three out of four networks, i.e., *france, di-yuan* and *cost266*, lowest revenue range and variance values were attained by RUL option. This differs from the results for the symmetrical networks, where the lowest variance was achieved for option RBL. We assume that small values for range and variance are preferred since this implies that revenues in different situations become similar, increasing fairness among situations. For all the network examples, mean, P25,

	R_s			u_g		
	RLL	RUL	RBL	RLL	RUL	RBL
Min	3774.25	9300.597	**13188.435**	2708	1232	1713.721
Max	3774.25	11492.469	**16524.305**	$0.529e5$	$1.209e5$	$1.852e5$
Range	0	**2191.872**	3335.87	$0.502e5$	$1.197e5$	$1.835e5$
Var	0	$\mathbf{0.705e6}$	$1.381e6$	$0.298e9$	$0.881e9$	$2.078e9$
Mean	3774.25	10694.534	**15547.865**	20615.08	26304.521	30486.808
P25	3774.25	9786.189	**14656.607**	7976	6598.667	4891.76
Median	3774.25	11200.221	**16294.319**	11843.5	10381	9635.86
P75	3774.25	11327.784	**16349.858**	36840.5	46217.933	45081
Sum	$1.547e5$	$4.385e5$	$\mathbf{6.374e5}$	$0.825e6$	$1.052e6$	$1.219e6$

Table 6.11: Revenue and capacity statistics for *france* network.

median and P75 of revenues were highest for RBL option, implying that in general higher revenues were achieved for this option, as compared to RLL and RUL. Sum of revenues over all situations was highest for RBL for all the four networks.

Analyzing the resulting link capacities for all the considered network examples (both, with symmetrical and unsymmetrical demand volumes) it is interesting to note that capacity assignments with lowest range and variance resulted from RLL option. The 25$^{\text{th}}$ percentile and the median value for capacities were also highest for RLL. For all networks with unsymmetrical demand volumes, except network *france*, even the 75$^{\text{th}}$ percentile for capacities was highest when using RLL. This means that when using RLL variation between the assigned link capacities was smaller, and at least half of the links were assigned higher capacities, compared to assignments resulting from RML and RBL cases. However, mean capacities for RML and RBL were in all cases higher than for RLL. Higher mean and variation between the link capacities for both, RML and RBL, as compared to RLL, is mainly because some small subset of links were assigned much higher capacities than other links in the same network. It is hard to say, which of the capacity allocations is preferred, since this greatly depends on network operation strategy of an operator, forecasted demands, etc.

6.8 Summary of the results

A numerical case study with a number of network examples (with symmetrical and unsymmetrical demand volumes) show that in all cases RBL achieves lexicographically largest vector of revenues. Both RBL and RUL achieved much higher revenues than RLL, since RBL and RUL employ lexicographical maxi-

	R_s			u_g		
	RLL	RUL	RBL	RLL	RUL	RBL
Min	2035.863	3362.34	**3365.949**	0	0	0
Max	2035.863	**4378.349**	4377.478	$2.325e4$	$5.638e4$	$5.642e4$
Range	0	1016.009	**1011.529**	$2.325e4$	$5.638e4$	$5.642e4$
Var	0	$9.352e4$	**$9.349e4$**	$0.711e8$	$2.98e8$	$2.985e8$
Mean	2035.863	4267.887	**4271.436**	3730.015	5864.915	5868.585
P25	2035.863	4358.998	**4365.894**	116.591	41.644	24.808
Median	2035.863	4367.726	**4372.752**	206.957	114.611	104.473
P75	2035.863	4376.421	**4377.478**	303.75	191.702	180.729
Sum	$0.814e5$	$1.707e5$	**$1.709e5$**	$1.455e5$	$2.287e5$	$2.289e5$

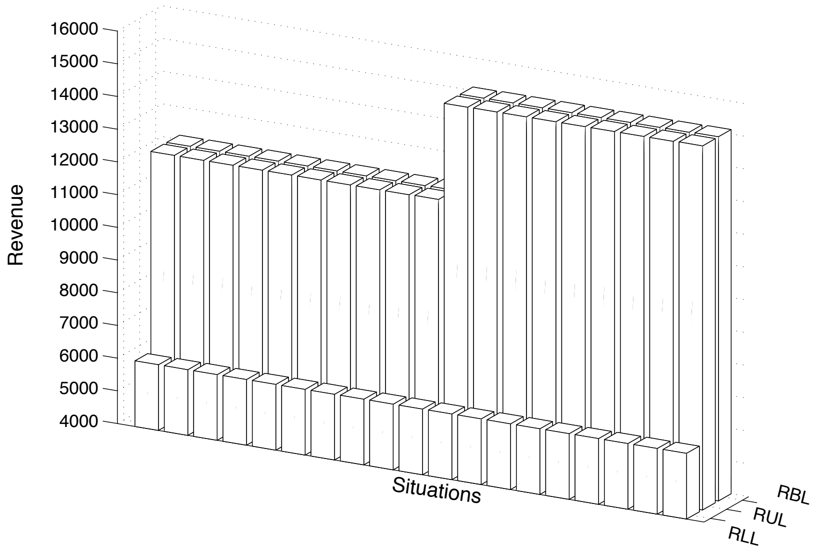
Table 6.12: Revenue and capacity statistics for *newyork* network.

	R_s			u_g		
	RLL	RUL	RBL	RLL	RUL	RBL
Min	525.76	1089.983	**1090.332**	0	0	0
Max	525.76	1603.929	**1604.968**	4510.483	$2.85e4$	$2.853e4$
Range	0	**513.946**	514.636	4510.483	$2.85e4$	$2.853e4$
Var	0	**$2.362e4$**	$2.375e4$	$0.28e7$	$7.5e7$	$7.519e7$
Mean	525.76	1551.735	**1553.489**	1663.353	2869.605	2871.545
P25	525.76	1597.456	**1600.384**	0	0	0
Median	525.76	1601.719	**1603.256**	1871.099	25.308	25.815
P75	525.76	1603.929	**1604.968**	2317.26	77.632	76.437
Sum	$1.63e4$	$4.81e4$	**$4.816e4$**	$4.99e4$	$8.609e4$	$8.615e4$

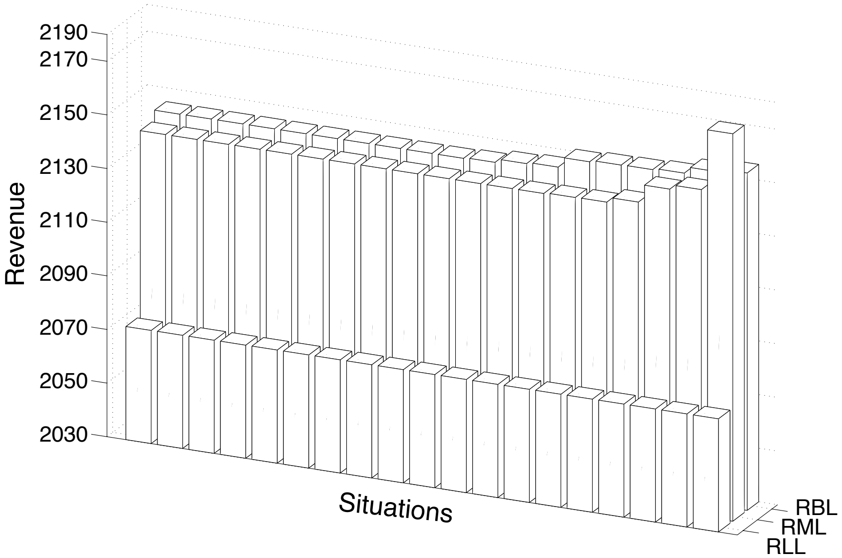
Table 6.13: Revenue and capacity statistics for *di-yuan* network.

	R_s			u_g		
	RLL	RUL	RBL	RLL	RUL	RBL
Min	$1.241e4$	$2.237e4$	**$3.228e4$**	4602.333	0	0
Max	$1.241e4$	$2.752e4$	**$3.905e4$**	$0.851e5$	$2.542e5$	$3.601e5$
Range	0	**5145.693**	6764.038	$0.805e5$	$2.542e5$	$3.601e5$
Var	0	**$1.708e6$**	$4.184e6$	$0.545e9$	$1.733e9$	$5.097e9$
Mean	$1.241e4$	$2.587e4$	**$3.698e4$**	$4.229e4$	$4.573e4$	$5.246e4$
P25	$1.241e4$	$2.513e4$	**$3.6e4$**	$2.265e4$	$2.162e4$	$2.265e4$
Median	$1.241e4$	$2.603e4$	**$3.768e4$**	$4.245e4$	$4.169e4$	$3.096e4$
P75	$1.241e4$	$2.706e4$	**$3.852e4$**	$6.33e4$	$4.793e4$	$5.123e4$
Sum	$0.77e6$	$1.604e6$	**$2.293e6$**	$2.58e6$	$2.79e6$	$3.2e6$

Table 6.14: Revenue and capacity statistics for *cost266* network.

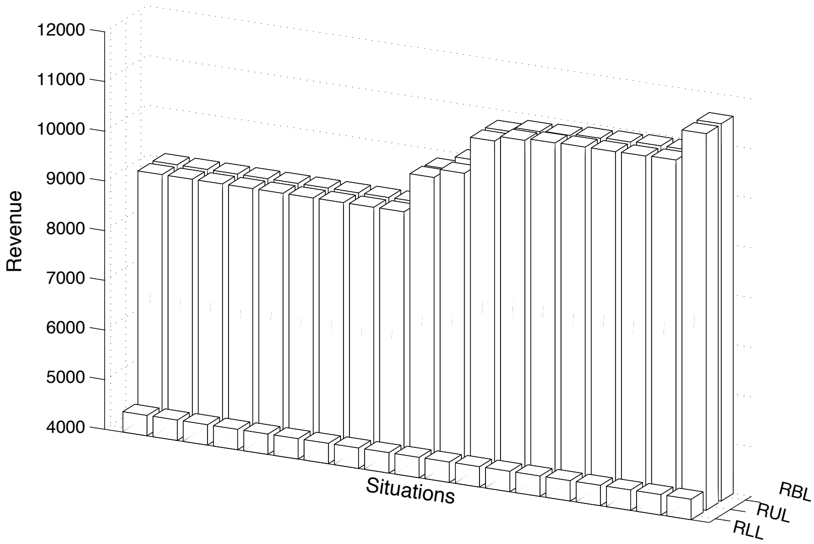


(a) N_{12}

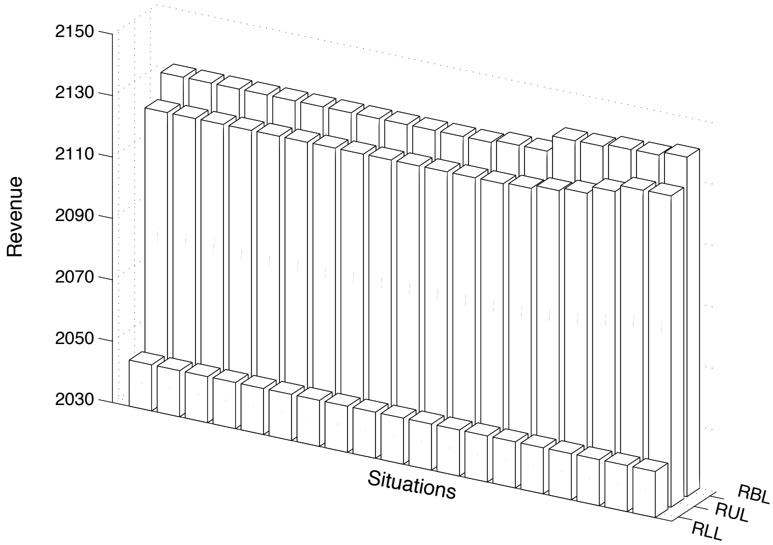


(b) N_{41}

Figure 6.12: Revenue values for RLL, RUL and RBL in the unbounded case.



(a) $N_{12}, LB = 1000$



(b) $N_{41}, LB = 10$

Figure 6.13: Revenue values for RLL, RUL and RBL in the bounded case.

mization. The difference between revenues achieved by RBL and RUL depends greatly on the ratio between the available network capacities (implied by budget and link costs) and the total network load, implied by demand volumes to be realized. The advantage of RBL becomes obvious for network instances for which design problems get more tight, i.e., when a surplus of budget after realizing minimum required demand volumes is not big. In all the cases RBL achieves (at least marginally) better solution than RUL, as any feasible solution of RUL is also a feasible solution of RBL.

For networks with symmetrical demand volumes RBL is clearly superior when topologies of the two layers are not similar, while RBL and RUL are almost equally good for the networks with similar topologies of layers (e.g., N_{12}). Similarly, RBL performs better than RUL when significant lower bounds are imposed. For all networks RBL performs much better when network load implied by the total demand volume is relatively high compared to the available network capacities, which depend on budget and link costs. It is interesting to note that lowest variance among the revenues was attained by RBL for networks with symmetrical demands and by RUL for networks with unsymmetrical demands. Lowest variance among link capacities was observed in the results for RLL option, while capacity mean values were highest for RBL and RUL (depending on the network instance).

These observations favor the RUL option for the networks with similar topologies of the layers, or when the network load (implied by total demand volume to be realized) is small compared to the available capacities (implied by budget and link costs). RUL option is preferred as it is based on a considerably simpler reconfiguration mechanism than RBL. For the networks with different topologies of the layers, or high load, RBL is significantly better than RUL.

In all the three models, full reconfiguration has been assumed in the case of failures, which is hard to implement, especially in the lower layer. However, the study shows what results could be achieved in that case and weather it is worth to use some kind of (coordinated) two-layer reconfiguration. The results can be used as an upper bound for evaluating more realistic (and thus restrictive) reconfiguration strategies, as, e.g., path protection and link protection.

CHAPTER 7

Path generation for two-layer network design problems

In this chapter we will discuss how path generation can be applied to the four two-layer network design problems introduced above. We develop the path generation algorithms for the linearized and relaxed problems. Thus we assume that the logarithmic functions in the objective function are replaced by their piece-wise linear approximation, and all variables are continuous and non-negative. The methods and algorithms presented for the four problems can be extended for other multi-layer network design problems.

7.1 Path generation for MNOM

In this section we will develop a path generation algorithm for a linearized and relaxed version of the MNOM problem given by (6.1)-(6.9), i.e., when the logarithms in objective function (6.1) are approximated by a piece-wise linear function, and variables x_{dp}, z_{ek}, y_e, u_g are continuous and non-negative. As for the single-layer problems discussed before, we note that this two-layer problem can also be formulated in the node-link formulation if all the possible paths are allowed in a problem. The need for using PG arises when a set of allowable paths is limited by some restrictions. It is this realistic assumption that gives us motivation for developing the PG algorithm for problem MNOM, as well as for other two-layer problems.

We will follow closely the steps of the PG algorithm development for problem NDT (see Section 5.2). Thus, we will first form the Lagrangian function for the MNOM, examine the dual variables and state the dual problem. Then the optimality conditions for PG can be specified using the dual variables.

Before proceeding to the dual problem we rewrite MNOM in a more compact way, as we did for problem NDT. Our aim is to eliminate unnecessary variables and to transform the constraints into a form more convenient for calculations.

First of all we rewrite constraint (6.5) as two separate constraints and substitute the expression for X_d from (6.4), arriving at the following two constraints:

$$h_d \leq \sum_p x_{dp}, \qquad \forall d \in \mathcal{D} \tag{7.1}$$

$$\sum_p x_{dp} \leq H_d, \qquad \forall d \in \mathcal{D}. \tag{7.2}$$

Secondly, we merge constraints (6.3) and (6.6) eliminating (auxiliary) variables y_e and arriving at the following constraint:

$$\sum_d \sum_p \delta_{edp} x_{dp} \leq \sum_k z_{ek}, \qquad \forall e \in \mathcal{E}. \tag{7.3}$$

Constraints (6.2) and (6.7) are also merged by substituting variable u_g in constraint (6.2) by the left-hand side of constraint (6.7):

$$\sum_g \sum_e \sum_k \varphi_{gek} z_{ek} \xi_g \leq B. \tag{7.4}$$

Furthermore, we linearize objective function (6.1) as described in Section 4.3.4 and will develop our resolution algorithm with embedded path generation for the linearized version of the MNOM problem.

The linearized relaxed MNOM (LRMNOM) problem is stated as follows:

LRMNOM:

$$\max \qquad R = \sum_d w_d f_d \tag{7.5}$$

s.t.:

$$[\pi \geq 0] \qquad \sum_g \sum_e \sum_k \varphi_{gek} z_{ek} \xi_g \leq B, \tag{7.6}$$

$$[\sigma_d \geq 0] \qquad h_d \leq \sum_p x_{dp}, \qquad \forall d \in \mathcal{D} \tag{7.7}$$

$$[\tau_d \geq 0] \qquad \sum_p x_{dp} \leq H_d, \qquad \forall d \in \mathcal{D} \tag{7.8}$$

$$[\gamma_e \geq 0] \qquad \sum_d \sum_p \delta_{edp} x_{dp} \leq \sum_k z_{ek}, \qquad \forall e \in \mathcal{E} \tag{7.9}$$

$$[\beta_{di} \geq 0] \qquad f_d \leq a_i \sum_p x_{dp} + b_i, \qquad \forall d \in \mathcal{D}, \quad \forall i \in \mathcal{I} \tag{7.10}$$

$$
\begin{aligned}
&f_d \in \mathbb{R}, && \forall d \in \mathcal{D} && && (7.11)\\
&x_{dp} \in \mathbb{R}^+, && \forall d \in \mathcal{D}, && \forall p \in \mathcal{P}_d && (7.12)\\
&z_{ek} \in \mathbb{R}^+, && \forall e \in \mathcal{E}, && \forall k \in \mathcal{K}_e && (7.13)
\end{aligned}
$$

Dual variables associated with constraints (7.6)-(7.10) are indicated in brackets in front of each constraint. As in the case of NDT, for convenience we transform objective (5.17) into minimization. The Lagrangian function for problem LRMNOM is:

$$
\begin{aligned}
L(\boldsymbol{f},\boldsymbol{x}\,;\,\pi,\boldsymbol{\beta},\boldsymbol{\sigma},\boldsymbol{\tau}) = &\sum_d \Big(\sum_i \beta_{di} - w_d\Big) f_d + \\
+ &\sum_d \sum_p \Big(\tau_d - \sigma_d - \sum_i a_i \beta_{di} + \sum_e \delta_{edp}\gamma_e\Big) x_{dp} \\
+ &\sum_e \sum_k \Big(\pi \sum_g \varphi_{gek}\xi_g - \gamma_e\Big) z_{ek} \\
+ &\sum_d \Big(h_d\sigma_d - H_d\tau_d - \sum_i b_i\beta_{di}\Big) - B\pi,
\end{aligned}
\tag{7.14}
$$

and the dual function of LRMNOM becomes:

$$
\begin{aligned}
W(\pi,\boldsymbol{\beta},\boldsymbol{\sigma},\boldsymbol{\tau},\boldsymbol{\gamma}) &= \min_{\boldsymbol{x},\boldsymbol{z}\geq 0\ ,\ \boldsymbol{f}\gtreqless 0} L(\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}\,;\,\pi,\boldsymbol{\beta},\boldsymbol{\sigma},\boldsymbol{\tau},\boldsymbol{\gamma}) \\
&= \sum_d \Big(h_d\sigma_d - H_d\tau_d - \sum_i b_i\beta_{di}\Big) - B\pi.
\end{aligned}
\tag{7.15}
$$

Constraining the values of dual variables only to those leading to the bounded dual function W, the dual problem to LRMNOM is stated as follows:
DLRMNOM:

$$
\begin{aligned}
\max\quad & W(\pi,\boldsymbol{\beta},\boldsymbol{\sigma},\boldsymbol{\tau},\boldsymbol{\gamma}) && && \\
\text{s.t.:}\quad & \sum_i \beta_{di} = w_d\,, && \forall d \in \mathcal{D} && (7.16)\\
& \sum_i a_i\beta_{di} + \sigma_d - \tau_d \leq \sum_e \delta_{edp}\gamma_e\,, && \forall d \in \mathcal{D}\,, \forall p \in \mathcal{P}_d && (7.17)\\
& \gamma_e \leq \pi \sum_g \varphi_{gek}\xi_g\,, && \forall e \in \mathcal{E}\,, \forall k \in \mathcal{K}_e && (7.18)\\
& \pi, \beta_{di}, \sigma_d, \tau_d, \gamma_e \in \mathbb{R}^+\,, && \forall d \in \mathcal{D},\ \forall i \in \mathcal{I}, && \\
& && \forall e \in \mathcal{E}. && (7.19)
\end{aligned}
$$

Let $\boldsymbol{x}^*$,$\boldsymbol{z}^*$,$\boldsymbol{f}^*$ be any primal optimal solution of LRMNOM. Similarly, let π^*, $\boldsymbol{\beta}^*$, $\boldsymbol{\sigma}^*$, $\boldsymbol{\tau}^*$, $\boldsymbol{\gamma}^*$ be any dual optimal solution. The primal and dual optimal solutions form a saddle point of the Lagrangian. (The dual optimal multipliers $\pi^*, \boldsymbol{\beta}^*, \boldsymbol{\sigma}^*, \boldsymbol{\tau}^*, \boldsymbol{\gamma}^*$ can be obtained directly from LP solvers.) The complementary slackness property states that in particular the following conditions must be satisfied for the primal and dual solutions to be optimal:

$$(\sum_e \delta_{edp}\gamma_e^* - \sum_i a_i\beta_{di}^* - \sigma_d^* + \tau_d^*)x_{dp}^* \;=\; 0\,, \quad \forall d \in \mathcal{D}, \forall p \in \mathcal{P}_d \tag{7.20}$$

$$(\pi^* \sum_g \varphi_{gek}\xi_g - \gamma_e^*)z_{ek}^* \;=\; 0\,, \quad \forall e \in \mathcal{E}, \forall k \in \mathcal{K}_e. \tag{7.21}$$

Thus, if the value in the parentheses is not equal to zero for at least one optimal dual solution then the corresponding primal variable must be zero. For example, if for some path p the value in parentheses of (7.20) is not zero, then the corresponding flow variable x_{dp}^* must be zero, implying that in any optimal primal solution the path is not carrying any flow. Therefore, for any saddle point of the Lagrangian (7.14), the following optimality conditions hold for the paths that carry non-negative flow in the optimal solution:

$$\sum_i \beta_{di}^* \;=\; w_d\,, \quad \forall d \in \mathcal{D} \tag{7.22}$$

$$\sum_i a_i\beta_{di}^* + \sigma_d^* - \tau_d^* \;=\; \min_{p\in\mathcal{P}_d}\left(\sum_e \delta_{edp}\gamma_e^*\right), \quad \forall d \in \mathcal{D} \tag{7.23}$$

$$\gamma_e^* \;=\; \pi^*\kappa_e\,, \quad \forall e \in \mathcal{E}\,, \tag{7.24}$$

where set $\mathcal{P}_d$ contains indices of paths in set $\mathscr{P}_d$, and κ_e is the length/cost of the shortest/cheapest path in the current candidate path list $\mathscr{K}_e$ for realizing the capacity of link e, and is calculated as follows:

$$\kappa_e = \min_{k\in\mathcal{K}_e}\left(\sum_g \varphi_{gek}\xi_g\right), \quad \forall e \in \mathcal{E}. \tag{7.25}$$

The set $\mathcal{K}_e$ contains indices of paths in set $\mathscr{K}_e$. Substituting the right-hand side of (7.24) to (7.23), we arrive at:

$$\sum_i a_i\beta_{di}^* + \sigma_d^* - \tau_d^* \;=\; \pi^*\mathcal{L}_d\,, \quad \forall d \in \mathcal{D}, \tag{7.26}$$

where $\mathcal{L}_d$ is the length/cost of the shortest/cheapest path in the current list of candidate paths $\mathscr{P}_d$ for realizing the demand d, and is calculated as follows:

$$\mathcal{L}_d = \min_{p \in \mathcal{P}_d} \left(\sum_e \delta_{edp} \kappa_e \right), \qquad \forall d \in \mathcal{D}. \tag{7.27}$$

We can recognize that (7.26) has the same form as (5.36), the corresponding expression for problem LRNDT. The only difference between the two is that for problem LRMNOM the value of L_d in (7.26) is calculated according to metrics $\boldsymbol{\kappa}$, whereas for problem LRNDT it is calculated with respect to real link costs $\boldsymbol{\xi}$. We also note that the two properties we have identified for problem LRNDT hold also for dual variables of problem LRMNOM. Thus, if for some demand $\hat{d}$ the inequality $h_{\hat{d}} < X_{\hat{d}} < H_{\hat{d}}$ holds and only one piece $i(\hat{d})$ of the piece-wise linear approximation is active, constraints (7.22)- (7.26) can be combined into the following expression, which is analogous to (5.38):

$$w_{\hat{d}} a_{i(\hat{d})} = \pi^* \mathcal{L}_{\hat{d}}. \tag{7.28}$$

We will now state the proposition underlying the path generation algorithm. Suppose that for a given network instance of problem LRMNOM all allowable paths in the upper layer form a set $\widehat{\mathscr{P}_d}$, and all allowable paths in the lower layer form set $\widehat{\mathscr{K}_e}$. For each upper layer link e let $\wp_e \in \widehat{\mathscr{K}_e}$ be one of the shortest paths among all allowable paths in the lower layer, and let $\bar{\ell}_e$ be its cost with respect to $\boldsymbol{\xi} = (\xi_g : g \in \mathcal{G})$, i.e.,

$$\bar{\ell}_e = \|\wp_e\|_{\boldsymbol{\xi}} = \min_{k \in \widehat{\mathcal{K}}_e} \left(\sum_g \varphi_{gek} \xi_g \right), \qquad \forall e \in \mathcal{E}, \tag{7.29}$$

where set $\widehat{\mathcal{K}}_e$ contains indices of the paths in set $\widehat{\mathscr{K}_e}$. Similarly, for each demand d, let $\wp_d \in \widehat{\mathscr{P}_d}$ be one of the shortest paths among all allowable paths in the upper layer, realizing bandwidth of demand d, and let ℓ_d be its cost with respect to the cost metrics $\boldsymbol{\kappa} = (\kappa_e : e \in \mathcal{E})$, i.e.,

$$\ell_d = \|\wp_d\|_{\boldsymbol{\kappa}} = \min_{p \in \widehat{\mathcal{P}}_d} \left(\sum_e \delta_{edp} \kappa_e \right), \qquad \forall d \in \mathcal{D}, \tag{7.30}$$

where set $\widehat{\mathcal{P}}_d$ contains indices of paths in set $\widehat{\mathscr{P}_d}$. We consider two versions of problem LRMNOM:

- *full* problem LRMNOM-FULL with the set of candidate paths for each demand d equal to $\widehat{\mathscr{P}_d}$ and the set of candidate paths for each link e equal to $\widehat{\mathscr{K}_e}$

- *limited* problem LRMNOM-LIMITED with the set of candidate paths for each demand d equal to $\mathscr{P}_d$, where $\mathscr{P}_d \subseteq \widehat{\mathscr{P}_d}$ for all d, the set of candidate paths for each link e equal to $\mathscr{K}_e$, where $\mathscr{K}_e \subseteq \widehat{\mathscr{K}_e}$ for all e, and $\mathscr{P}_d \subsetneq \widehat{\mathscr{P}_d}$ for at least one demand d, and/or $\mathscr{K}_e \subsetneq \widehat{\mathscr{K}_e}$ for at least one link e.

Proposition 7.1. *Let ($\boldsymbol{f}^*$, $\boldsymbol{x}^*$, $\boldsymbol{z}^*$; π^*,$\boldsymbol{\beta}^*$, $\boldsymbol{\sigma}^*$, $\boldsymbol{\tau}^*$, γ^*) be a saddle point of the Lagrangian for an instance of the limited problem LRMNOM-LIMITED with given sets of candidate paths $\mathscr{P}_d$ for all $d \in \mathcal{D}$ and $\mathscr{K}_e$ for all $e \in \mathcal{E}$, in the upper and lower layers, respectively. Let $\wp_e \in \widehat{\mathscr{K}_e} \setminus \mathscr{K}_e$ be a shortest path in the lower layer realizing capacity of link e and let $\bar{\ell}_e$ denote its length (7.29). Similarly, let $\wp_d \in \widehat{\mathscr{P}_d} \setminus \mathscr{P}_d$ be a shortest path in the upper layer realizing demand d and let ℓ_d denote its length (7.30). Then:*

- *If $\kappa_e = \bar{\ell}_e$ for all $e \in \mathcal{E}$ and $\mathcal{L}_d = \ell_d$ for all $d \in \mathcal{D}$ then any optimal solution to LRMNOM-LIMITED is also optimal for LRMNOM-FULL, i.e., the solution is optimal in a wider sense. Notice that this is the case when problem LRMNOM-FULL has no paths shorter than LRMNOM-LIMITED.*

- *Otherwise, the current optimal solution may be possibly improved in any of the following three cases:*

Case 1: Suppose that for all $e \in \mathcal{E}$ the length of the shortest path in the candidate path list $\mathscr{K}_e$ is equal to $\bar{\ell}_e$, i.e., $\kappa_e = \bar{\ell}_e$. Suppose also that there exists a demand d and a path $\wp_d \in \widehat{\mathscr{P}_d} \setminus \mathscr{P}_d$ with length $\ell_d < \mathcal{L}_d$. Then including path $\wp_d$ into list $\mathscr{P}_d$ for at least one such demand d can improve the current optimal solution.

Case 2: Suppose that there exists a link e and a path $\wp_e \in \widehat{\mathscr{K}_e} \setminus \mathscr{K}_e$ with length $\bar{\ell}_e < \kappa_e$. Suppose also that there exists a demand d, for which, if $\wp_e$ is included into $\mathscr{K}_e$ implying the new cost vector $\boldsymbol{\kappa}'$, there exists a path $\wp_d \in \widehat{\mathscr{P}_d} \setminus \mathscr{P}_d$ with length $\ell_d < \mathcal{L}_d$, which is a shortest path for the demand with respect to costs $\boldsymbol{\kappa}'$. Then including path $\wp_e$ into list $\mathscr{K}_e$ and including path $\wp_d$ into list $\mathscr{P}_d$, for at least one such link e and demand d, can improve the current optimal solution.

Case 3: Suppose there exists a link e and a path $\wp_e \in \widehat{\mathscr{K}_e} \setminus \mathscr{K}_e$ with length $\bar{\ell}_e < \kappa_e$. Suppose also that, when $\wp_e$ is included into $\mathscr{K}_e$ implying *the new cost vector $\boldsymbol{\kappa}'$, the candidate path list $\mathscr{P}_d$ for every demand d contains at least one shortest (with respect to costs $\boldsymbol{\kappa}'$) path with length $\mathcal{L}_d = \ell_d$. Then including path $\wp_e$ into list $\mathscr{K}_e$, for at least one such link e, can improve the current optimal solution.*

The proof of the above proposition is analogous to proof of Proposition 5.1 in Section 5.2, and therefore we will omit it, and just comment on similarities and differences between the two problems.

The dual objective function (5.27) for problem LRNDT (see Section 5.2) is identical with (7.15). Likewise, the dual optimality conditions (7.22) and (7.26) for problem LRMNOM are identical with the conditions (5.34)-(5.36) for LRNDT, except of the fact that for LRNDT the length of the path for demand d, $\mathcal{L}_d$, is calculated in (5.33) with respect to real costs $\boldsymbol{\xi}$, while for LRMNOM it is calculated in (7.27) with respect to costs $\boldsymbol{\kappa}$. Essentially, the lower layer structure in problem LRMNOM is hidden behind link costs $\boldsymbol{\kappa}$, thus making problems LRNDT and LRMNOM equivalent from the viewpoint of a demand d. Thus, the approach taken in proving the proposition for LRNDT applies also for problem LRMNOM.

However, the proof covers only Case 1 of Proposition 7.1, i.e., when the lower layer costs $\boldsymbol{\kappa}$ do not change. Still, the reasoning in Proposition 5.1 can be extended also to Cases 2 and 3, observing that if for some link $\hat{e}$ belonging to a path for demand $\hat{d}$ there exists a shorter path in the lower layer, implying that the cost $\kappa_{\hat{e}}$ is decreased, this is equivalent to having found a shorter path for demand $\hat{d}$.

Assuming that only one piece of the approximation is active for the total flow of some demand $\hat{d}$, and the demand volume is between the bounds the maximum revenue gain which can be attained by moving a flow of size ε of some demand $\hat{d}$ to a shorter path not belonging to the set of candidate paths, is given by (5.51), i.e.,

$$(R^{\varepsilon})^* - R^* = \varepsilon w_{\hat{d}} a_{i(\hat{d})} \frac{\mathcal{L}_{\hat{d}} - \ell_{\hat{d}}}{\mathcal{L}_{\hat{d}}}. \tag{7.31}$$

where π^* is calculated from equation (7.26), and $\mathcal{L}_{\hat{d}}$ is calculated from (7.27). We have noted in Section 5.2 that expression (7.31) gives only a maximum possible (not always attainable) revenue gain, and that the gain for a particular demand depends on the approximation of the demands flow, and weather the flow is equal to any of the bounds h_d or H_d. The same comments apply also for problem LRMNOM.

The path generation procedure for problem LRMNOM has only one iteration, and consists in two steps:

- For each upper layer link e find a shortest path in the lower layer (with respect to costs $\boldsymbol{\xi}$) and add it to the list of candidate paths $\mathscr{K}_e$.
- For each demand d find a shortest path in the upper layer (with respect to costs $\boldsymbol{\kappa}$) and add it to the list of candidate paths $\mathscr{P}_d$.

After executing the two steps the algorithm terminates. Thus the stopping criterion is implied. The procedure is applicable when a problem instance at hand does not initially contain the shortest path for each demand in the candidate path lists.

7.2 Path generation for RLL

Analogously to MNOM, we first merge constraints (6.14) and (6.17) eliminating (auxiliary) variables y_e and arriving at the following constraint:

$$\sum_d \sum_p \delta_{edp} x_{dp} \ \le \ \sum_k z_{eks} \,, \qquad \forall e \in \mathcal{E}, \forall s \in \mathcal{S}. \tag{7.32}$$

We also rewrite constraint (6.16) as two separate constraints (7.1) and (7.2), and linearize objective function (6.12) in a way we did it for problem MNOM, arriving at the linearized relaxed formulation for problem RLL (LRRLL):

LRRLL:

$$\max \qquad R = \sum_d w_d f_d \tag{7.33}$$

s.t.:

$$[\pi \ge 0] \qquad \sum_g \xi_g u_g \ \le B, \tag{7.34}$$

$$[\sigma_d \ge 0] \qquad h_d \ \le \ \sum_p x_{dp}, \qquad \forall d \in \mathcal{D} \tag{7.35}$$

$$[\tau_d \ge 0] \qquad \sum_p x_{dp} \ \le \ H_d, \qquad \forall d \in \mathcal{D} \tag{7.36}$$

$$[\gamma_{es} \ge 0] \qquad \sum_d \sum_p \delta_{edp} x_{dp} \ \le \ \sum_k z_{eks}, \quad \forall e \in \mathcal{E},\ \forall s \in \mathcal{S} \tag{7.37}$$

$$[\mu_{gs} \ge 0] \qquad \sum_e \sum_k \varphi_{gek} z_{eks} \ \le \ \alpha_{gs} u_g, \quad \forall g \in \mathcal{G},\ \forall s \in \mathcal{S} \tag{7.38}$$

$$
\begin{array}{llll}
[\beta_{di} \geq 0] & f_d \;\leq\; a_i \sum_p x_{dp} + b_i, & \forall d \in \mathcal{D},\ \forall i \in \mathcal{I} & (7.39) \\
& f_d \in \mathbb{R},\ x_{dp} \in \mathbb{R}^+, & \forall d \in \mathcal{D},\ \forall p \in \mathcal{P}_d & (7.40) \\
& u_{gs} \in \mathbb{R}^+, & \forall g \in \mathcal{G},\ \forall s \in \mathcal{S} & (7.41) \\
& z_{eks} \in \mathbb{R}^+, & \forall e \in \mathcal{E},\ \forall k \in \mathcal{K}_e, & \\
& & \forall s \in \mathcal{S}. & (7.42)
\end{array}
$$

Dual variables associated with constraints (7.34)-(7.39) are indicated in brackets in front of each constraint. As before, for convenience we transform objective (7.33) into minimization and form the Lagrangian function:

$$
\begin{aligned}
L(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}\ ;\ \pi, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{\mu}) &= \sum_d \Big(\sum_i \beta_{di} - w_d \Big) f_d \\
&+ \sum_d \sum_p \Big(\tau_d - \sigma_d - \sum_i a_i \beta_{di} + \sum_e \delta_{edp} \sum_s \gamma_{es} \Big) x_{dp} \\
&+ \sum_e \sum_k \sum_s \Big(\sum_g \varphi_{gek} \mu_{gs} - \gamma_{es} \Big) z_{eks} \\
&+ \sum_g \Big(\xi_g \pi - \sum_s \alpha_{gs} \mu_{gs} \Big) u_g \\
&+ \sum_d \Big(h_d \sigma_d - H_d \tau_d - \sum_i b_i \beta_{di} \Big) - \pi B,
\end{aligned}
\tag{7.43}
$$

and the dual function of LRRLL is the same as for problem LRMNOM:

$$
\begin{aligned}
W(\pi, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{\mu}) &= \min_{\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u} \geq \boldsymbol{0}\ ,\ \boldsymbol{f} \gtreqless 0} L(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}\ ;\ \pi, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{\mu}) \\
&= \sum_d (h_d \sigma_d - H_d \tau_d - \sum_i b_i \beta_{di}) - \pi B.
\end{aligned}
\tag{7.44}
$$

Constraining the values of dual variables only to those leading to the bounded dual function W, the dual problem to LRRLL is as follows:

DLRRLL:

$$
\begin{array}{llll}
\max & W(\pi, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{\mu}) & & (7.45) \\
\text{s.t.:} & \sum_i \beta_{di} = w_d, & \forall d \in \mathcal{D} & (7.46) \\
& \sigma_d - \tau_d + \sum_i a_i \beta_{di} \leq \sum_e \delta_{edp} \sum_s \gamma_{es}, & \forall d \in \mathcal{D}, p \in \mathcal{P}_d & (7.47)
\end{array}
$$

$$\gamma_{es} \leq \sum_g \varphi_{gek}\mu_{gs}, \qquad \forall e \in \mathcal{E}, k \in \mathcal{K}_e,$$

$$\forall s \in \mathcal{S} \tag{7.48}$$

$$\sum_s \alpha_{gs}\mu_{gs} \leq \xi_g \pi, \qquad \forall g \in \mathcal{G} \tag{7.49}$$

$$\pi \geq 0; \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{\mu} \geq \mathbf{0}. \tag{7.50}$$

Let $\boldsymbol{f}^*$, $\boldsymbol{x}^*$, $\boldsymbol{z}^*$, $\boldsymbol{u}^*$ be any primal optimal solution of LRRLL. Similarly, let π^*, $\boldsymbol{\beta}^*$, $\boldsymbol{\sigma}^*$, $\boldsymbol{\tau}^*$, $\boldsymbol{\gamma}^*$, $\boldsymbol{\mu}^*$ be any dual optimal solution. The primal and dual optimal solutions form a saddle point of the Lagrangian (7.43). (The dual optimal multipliers together with optimal primal variables can be obtained from LP solvers.) It can be observed that for any saddle point of the Lagrangian the following optimality conditions hold implied by the complementary slackness property:

$$\sum_i \beta^*_{di} = w_d, \qquad \forall d \in \mathcal{D} \tag{7.51}$$

$$\sum_i a_i\beta^*_{di} + \sigma^*_d - \tau^*_d = \mathcal{L}_d, \qquad \forall d \in \mathcal{D} \tag{7.52}$$

$$\gamma^*_{es} = \kappa_{es}, \qquad \forall e \in \mathcal{E}, s \in \mathcal{S} \tag{7.53}$$

$$\sum_s \alpha_{gs}\mu^*_{gs} = \xi_g \pi^* \ , \qquad \forall g \in \mathcal{G}. \tag{7.54}$$

At optimum, each dual variable μ^*_{gs} can be interpreted as a (dual) cost of link g in situation s. Let $\mathscr{K}_e$ be the given list of candidate paths in the lower layer realizing capacity of link e and let κ_{es} in equation (7.53) be a length of the shortest path for link e in situation s in this list with respect to metric $\boldsymbol{\mu}^* = (\mu^*_{gs} : g \in \mathcal{G}, s \in \mathcal{S})$:

$$\kappa_{es} = \min_{k \in \mathcal{K}_e} \left(\sum_g \varphi_{gek}\mu^*_{gs} \right), \qquad \forall e \in \mathcal{E}, \forall s \in \mathcal{S}. \tag{7.55}$$

Set $\mathcal{K}_e$ contains indices of paths in set $\mathscr{K}_e$. Similarly, let $\mathscr{P}_d$ be the given list of candidate paths in the upper layer realizing demand d and let $\mathcal{L}_d$ in equation (7.52) be a length of the shortest path for demand d in this list with respect to costs $\boldsymbol{\kappa} = (\kappa_{es} : e \in \mathcal{E}, s \in \mathcal{S})$:

$$\mathcal{L}_d = \min_{p \in \mathcal{P}_d} \left(\sum_e \delta_{edp} \sum_s \gamma^*_{es} \right) = \min_{p \in \mathcal{P}_d} \left(\sum_e \delta_{edp} \sum_s \kappa_{es} \right), \forall d \in \mathcal{D}. \tag{7.56}$$

Set $\mathcal{P}_d$ contains indices of paths in set $\mathscr{P}_d$. The last sum in (7.56) shows that cost of some link e in every situation s contributes to the final cost of the link, which is then used for finding the shortest path for a demand. Let us define this total cost of a link by L_e, i.e.,

$$L_e = \sum_s \kappa_{es}, \qquad \forall e \in \mathcal{E}, \tag{7.57}$$

and rewrite (7.56) as:

$$\mathcal{L}_d = \min_{p \in \mathcal{P}_d} \left(\sum_e \delta_{edp} L_e \right), \qquad \forall d \in \mathcal{D}. \tag{7.58}$$

We also observe from (7.54), that real cost ξ_g (scaled by π^*) of each lower layer link is, in general, shared among the situations in which the link survives. Thus we define a set $\mathcal{S}^g$ containing all situations in which link g is fully available, i.e.,

$$\mathcal{S}^g = \{s \in \mathcal{S} : \alpha_{gs} = 1\} \qquad \forall g \in \mathcal{G} \tag{7.59}$$

and rewrite equation (7.54) as follows:

$$\sum_{s \in \mathcal{S}^g} \mu^*_{gs} = \xi_e \pi^*, \qquad \forall g \in \mathcal{G}. \tag{7.60}$$

We note that the two properties (see the discussion for problem MNOM in Section 7.1) regarding dual variables $\boldsymbol{\beta}$,$\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ hold for this problem as well.

We will now state the proposition underlying the PG algorithm for problem LRRLL. Suppose that for a given network instance of problem LRRLL all allowable paths (in the lower layer) realizing each link e belong to set $\widehat{\mathscr{K}_e}$ and all allowable paths (in the upper layer) for each demand d belong to set $\widehat{\mathscr{P}_d}$. Let $\widehat{\mathscr{K}_{es}} \subseteq \widehat{\mathscr{K}_e}$ be a subset of lower layer paths for link e that are fully available (not failing) in situation s. For each upper layer link e and situation s, let $\wp_{es} \in \widehat{\mathscr{K}_{es}}$ be a one of the shortest paths in the lower layer among all allowable paths surviving failure situation s, and let $\bar{\ell}_{es}$ be its cost with respect to the cost metrics $\boldsymbol{\mu}^*$ be , i.e.,

$$\bar{\ell}_{es} = \|\wp_{es}\|_{\boldsymbol{\mu}^*} = \min_{k \in \widehat{\mathcal{K}}_{es}} \left(\sum_g \varphi_{gek} \mu^*_{gs} \right), \qquad \forall e \in \mathcal{E}, \forall s \in \mathcal{S}, \tag{7.61}$$

where set $\widehat{\mathcal{K}}_{es}$ contains indices of paths in set $\widehat{\mathscr{K}_{es}}$. For each demand d, let $\wp_d \in \widehat{\mathscr{P}_d}$ be a shortest path in the upper layer among all allowable paths realizing bandwidth of demand d and let ℓ_d be its cost with respect to the cost metrics $\boldsymbol{L} = (L_e : e \in \mathcal{E})$, i.e.,

$$\ell_d = \|\wp_d\|_{\boldsymbol{L}} = \min_{p \in \widehat{\mathcal{P}}_d} \left(\sum_e \delta_{edp} L_e \right), \qquad \forall d \in \mathcal{D}, \tag{7.62}$$

where set $\widehat{\mathcal{P}}_d$ contains indices of paths in set $\widehat{\mathscr{P}_d}$. We consider two versions of problem LRRLL:

- *full* problem LRRLL-FULL with the set of candidate paths for each demand d equal to $\widehat{\mathscr{P}_d}$ and the set of candidate paths for each link e equal to $\widehat{\mathscr{K}_e}$
- *limited* problem LRRLL-LIMITED with the set of candidate paths for each demand d equal to $\mathscr{P}_d$, where $\mathscr{P}_d \subseteq \widehat{\mathscr{P}_d}$ for all d, the set of candidate paths for each link e equal to $\mathscr{K}_e$, where $\mathscr{K}_e \subseteq \widehat{\mathscr{K}_e}$ for all e, and $\mathscr{P}_d \subsetneq \widehat{\mathscr{P}_d}$ for at least one demand d, and/or $\mathscr{K}_e \subsetneq \widehat{\mathscr{K}_e}$ for at least one link e.

Proposition 7.2. *Let ($\boldsymbol{f}^*$, $\boldsymbol{x}^*$, $\boldsymbol{z}^*$, $\boldsymbol{u}^*$; π^*,$\boldsymbol{\beta}^*$, $\boldsymbol{\sigma}^*$, $\boldsymbol{\tau}^*$, γ^*, $\boldsymbol{\mu}^*$) be a saddle point of the Lagrangian for an instance of the limited problem LRRLL-LIMITED with given sets of candidate paths $\mathscr{P}_d$ for all $d \in \mathcal{D}$ and $\mathscr{K}_e$ for all $e \in \mathcal{E}$, in the upper and lower layers, respectively. Let $\wp_{es} \in \widehat{\mathscr{K}_{es}} \setminus \mathscr{K}_e$ be a shortest path in the lower layer realizing capacity of link e in situation s and let $\bar{\ell}_{es}$ denote its length (7.61). Similarly, let $\wp_d \in \widehat{\mathscr{P}_d} \setminus \mathscr{P}_d$ be a shortest path in the upper layer realizing demand d and let ℓ_d denote its length (7.62). Then:*

- *If $\kappa_{es} = \bar{\ell}_{es}$ for all $e \in \mathcal{E}$, $s \in \mathcal{S}$ and $\mathcal{L}_d = \ell_d$ for all $d \in \mathcal{D}$ then any optimal solution to LRRLL-LIMITED is also optimal for LRRLL-FULL, i.e., the solution is optimal in a wider sense. Notice that this is the case when problem LRRLL-FULL has no paths shorter than LRRLL-LIMITED.*
- *Otherwise, the current optimal solution may be possibly improved in any of the following three cases:*

Case 1: Suppose that for all $e \in \mathcal{E}$ and $s \in \mathcal{S}$ the length of the shortest path in the candidate path list $\mathscr{K}_e$ is equal to $\bar{\ell}_{es}$, i.e., $\kappa_{es} = \bar{\ell}_{es}$. Suppose also that there exists a demand d and a path $\wp_d \in \widehat{\mathscr{P}_d} \setminus \mathscr{P}_d$ with length $\ell_d < \mathcal{L}_d$. Then including path $\wp_d$ into list $\mathscr{P}_d$ for at least one such demand d can improve the current optimal solution.

Case 2: *Suppose that there exists a link* e*, situation* s *and a path* $\wp_{es} \in \widehat{\mathscr{K}_{es}} \setminus \mathscr{K}_{\mathrm{e}}$ *with length* $\bar{\ell}_{es} < \kappa_{es}$*. Suppose also that there exists a demand* d*, for which, if* $\wp_{es}$ *is included into* $\mathscr{K}_{\mathrm{e}}$ *implying the new cost vector* $\boldsymbol{\kappa}'$*, there exists a path* $\wp_{\mathrm{d}} \in \widehat{\mathscr{P}_{\mathrm{d}}} \setminus \mathscr{P}_{\mathrm{d}}$ *with length* $\ell_{\mathrm{d}} < \mathcal{L}_{\mathrm{d}}$*, which is a shortest path for the demand with respect to costs* $\boldsymbol{\kappa}'$*. Then including path* $\wp_{es}$ *into list* $\mathscr{K}_{\mathrm{e}}$ *for at least on such link* e *and situation* s*, and including path* $\wp_{\mathrm{d}}$ *into list* $\mathscr{P}_{\mathrm{d}}$*, for at least one such demand* d*, can improve the current optimal solution.*

Case 3: *Suppose there exists a link* e*, situation* s *and a path* $\wp_{es} \in \widehat{\mathscr{K}_{es}} \setminus \mathscr{K}_{\mathrm{e}}$ *with length* $\bar{\ell}_{es} < \kappa_{es}$*. Suppose also that, when* $\wp_{es}$ *is included into* $\mathscr{K}_{\mathrm{e}}$ *implying the new cost vector* $\boldsymbol{\kappa}'$*, the candidate path list* $\mathscr{P}_{\mathrm{d}}$ *for every demand* d *contains at least one shortest (with respect to costs* $\boldsymbol{\kappa}'$*) path with length* $\mathcal{L}_{\mathrm{d}} = \ell_{\mathrm{d}}$*. Then including path* $\wp_{es}$ *into list* $\mathscr{K}_{\mathrm{e}}$*, for at least one such link* e*, can improve the current optimal solution.*

The proof of the above proposition is analogous to proof of Proposition 5.1 in Section 5.2, and therefore it is omitted here.

7.2.1 Algorithm for resolving LRLL using PG

Problem LRRLL can be resolved with the following iterative algorithm, using PG. The test for the shortest paths, according to Proposition 7.2, is performed in Step 2. Any suitable shortest path algorithm (e.g., Dijkstra's algorithm) can be used for finding the shortest paths.

Resolution algorithm for LRRLL using PG.

Step 0: Form the initial instance of the LRRLL problem, using small sets of candidate paths (e.g., single paths).

Step 1: Resolve problem LRRLL (if Step 1 is entered from Step 2, then use the current optimal solution from the previous iteration as a staring point for solving the new instance of the problem). Get the values of dual variables from the LP solver. Let ($\boldsymbol{f}^*$, $\boldsymbol{x}^*$, $\boldsymbol{z}^*$, $\boldsymbol{u}^*$; π^*,$\boldsymbol{\beta}^*$, $\boldsymbol{\sigma}^*$, $\boldsymbol{\tau}^*$, γ^*, $\boldsymbol{\mu}^*$) be a saddle point of the Lagrangian for the problem.

Step 2: Do the following test:

- For each link e and situation s run a shortest path algorithm, using $\boldsymbol{\mu}^*$ as L1 links' costs, to find a shortest path $\wp_{es}$ and its cost $\bar{\ell}_{es}$. If there exists path $\wp_{es}$ such that $\bar{\ell}_{es} < \kappa_{es}$ then include it to the list $\mathscr{K}_e$.

- For each demand d run a shortest path algorithm, using L_e (7.57) as L2 links' costs, to find a shortest path $\wp_d$ and its cost ℓ_d. If there exists path $\wp_d$ such that $\ell_d < \mathcal{L}_d$, then include it to the list $\mathscr{P}_d$.

If for all d, $\mathcal{L}_d = \ell_d$ then **STOP**. The current solution to the problem is optimal in a wider sense. Otherwise, form a new instance of the problem with the extended lists of candidate paths. **Go to Step 1**.

7.3 Path generation for RUL

Recall that problem RUL uses lexicographical maximization and therefore requires a special resolution algorithm. Such an algorithm was presented in Section 6.6.1 and will be the basis for our PG-based resolution algorithm presented here.

Similarly to problem RLL, we will first restate few constraints and then define the linearized version of the convex problem solved in Step 1 of the resolution algorithm (presented in Section 6.6.1) for problem RUL.

Constraint (6.26) is divided into two separate constraints and the expression for X_{ds} is substituted from (6.25), arriving at:

$$h_{ds} \le \sum_p x_{dps}\,, \qquad \forall d \in \mathcal{D},\ \forall s \in \mathcal{S} \tag{7.63}$$

$$\sum_p x_{dps} \le H_{ds}\,, \qquad \forall d \in \mathcal{D},\ \forall s \in \mathcal{S}. \tag{7.64}$$

Constraints (5.54) and (6.27) are merged into the following constraint:

$$\sum_d \sum_p \delta_{edp} x_{dps} \le \sum_k \theta_{eks} z_{ek}\,, \qquad \forall e \in \mathcal{E},\ \forall s \in \mathcal{S}. \tag{7.65}$$

Problem RUL is linearized by applying the piece-wise linear approximation of the logarithmic function (6.32) as described in Section 4.3.4. Then constraints (6.45) and (6.46) from the convex program in Step 1 of the RUL resolution algorithm may be rewritten as:

$$t_s - \sum_d w_{ds} f_{ds} \le 0\,, \qquad \forall s \in Z_0 \tag{7.66}$$

$$t - \sum_d w_{ds} f_{ds} \le 0\,, \qquad \forall s \in Z_1 \tag{7.67}$$

$$t \in \mathbb{R}. \tag{7.68}$$

Taking all the above-mentioned modifications into account the convex program to be solved in Step 1 of the RUL resolution algorithm (see Section 6.6.1) is stated as follows:

LRRUL:

$$\max \quad t \tag{7.69}$$

s.t.:

$$[\pi \geq 0] \quad \sum_g \sum_e \sum_k \varphi_{gek} z_{ek} \xi_g \leq B, \tag{7.70}$$

$$[\sigma_{ds} \geq 0] \quad h_{ds} \leq \sum_p x_{dps}, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{7.71}$$

$$[\tau_{ds} \geq 0] \quad \sum_p x_{dps} \leq H_{ds}, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{7.72}$$

$$[\gamma_{es} \geq 0] \quad \sum_d \sum_p \delta_{edp} x_{dps} \leq \sum_k \theta_{eks} z_{ek}, \quad \forall e \in \mathcal{E}, s \in \mathcal{S} \tag{7.73}$$

$$[\beta_{dsi} \geq 0] \quad f_{ds} \leq a_i \sum_p x_{dps} + b_i, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S}, \; \forall i \in \mathcal{I} \tag{7.74}$$

$$[\lambda_s \geq 0] \quad t_s - \sum_d w_{ds} f_{ds} \leq 0, \qquad \forall s \in Z_0 \tag{7.75}$$

$$[\lambda_s \geq 0] \quad t - \sum_d w_{ds} f_{ds} \leq 0, \qquad \forall s \in Z_1 \tag{7.76}$$

$$t, f_{ds} \in \mathbb{R}, \; x_{dps} \in \mathbb{R}^+, \qquad \forall d \in \mathcal{D}, p \in \mathcal{P}_d, \; \forall s \in \mathcal{S} \tag{7.77}$$

$$z_{eks} \in \mathbb{R}^+, \qquad \forall e \in \mathcal{E}, k \in \mathcal{K}_e, \; \forall s \in \mathcal{S}. \tag{7.78}$$

Dual variables associated with each of constraints (7.70)-(7.76) are indicated in brackets in front of each constraint. Observe, that dual variables $\boldsymbol{\lambda}$ are defined for both constraints (7.75) and (7.76), since $Z_0 \cap Z_1 = \emptyset$. As before, for convenience we transform objective (7.33) into minimization. The Lagrangian function for problem LRRUL is:

$$
\begin{aligned}
L(t, \boldsymbol{f}, \boldsymbol{x}, \boldsymbol{z}\,;\, \pi, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \gamma) = & \Big(\sum_{s\in Z_1} \lambda_s - 1\Big)t \\
&+ \sum_d \sum_s \Big(\sum_i \beta_{dsi} - w_{ds}\lambda_s\Big) f_{ds} \\
&+ \sum_d \sum_p \sum_s \Big(\tau_{ds} - \sigma_{ds} - \sum_i a_i \beta_{dsi} + \sum_e \delta_{edp}\gamma_{es}\Big) x_{dps} \\
&+ \sum_e \sum_k \Big(\pi \sum_g \varphi_{gek}\xi_g - \sum_s \theta_{eks}\gamma_{es}\Big) z_{ek} \\
&+ \sum_{s\in Z_0} \lambda_s t_s + \sum_d \sum_s \Big(h_{ds}\sigma_{ds} - H_{ds}\tau_{ds} - \sum_i b_i \beta_{dsi}\Big) - \pi B.
\end{aligned} \tag{7.79}
$$

Let t^*,$\boldsymbol{f}^*$,$\boldsymbol{x}^*$,$\boldsymbol{z}^*$ be any primal optimal solution of LRRUL. Similarly, let π^*,$\boldsymbol{\beta}^*$, $\boldsymbol{\lambda}^*$, $\boldsymbol{\sigma}^*$, $\boldsymbol{\tau}^*$, $\boldsymbol{\gamma}^*$ be any dual optimal solution. The primal and dual optimal solutions form a saddle point of the Lagrangian (7.79). (The optimal dual multipliers and optimal primal variables can be obtained from LP solvers.) It can be observed that for any saddle point of the Lagrangian the following optimality conditions hold implied by the complementary slackness property:

$$\sum_{s\in Z_1} \lambda^*_s = 1 \tag{7.80}$$

$$\sum_i \beta^*_{dsi} = w_{ds}\lambda^*_s\,, \qquad \forall d \in \mathcal{D}, \forall s \in \mathcal{S} \tag{7.81}$$

$$\sigma^*_{ds} - \tau^*_{ds} + \sum_i a_i \beta^*_{dsi} = \mathcal{L}_{ds}\,, \qquad \forall d \in \mathcal{D}, \forall s \in \mathcal{S} \tag{7.82}$$

$$\sum_s \theta_{eks}\gamma^*_{es} \le \pi^* \sum_g \varphi_{gek}\xi_g\,, \qquad \forall e \in \mathcal{E}, \forall k \in \mathcal{K}_e \tag{7.83}$$

At optimum, each dual variable γ^*_{es} represent a cost of link e in situation s. It can be observed from (7.83) that, in general, for each link e and path k, sum of costs γ^*_{es} over situations in which the path is available (not failing) share the real cost of the path (calculated according to metrics $\boldsymbol{\xi} = (\xi_g : g \in \mathcal{G})$) scaled by π^*. Let $\mathscr{K}_e$ be the given list of candidate paths in the lower layer realizing capacity of link e and let κ_e be a length of the shortest path for link e in this list with respect to metric $\boldsymbol{\xi}$:

$$\kappa_e = \min_{k\in\mathcal{K}_e} \Big(\sum_k \varphi_{gek}\xi_g\Big), \qquad \forall e \in \mathcal{E}. \tag{7.84}$$

The set $\mathcal{K}_e$ contains indices of paths in set $\mathscr{K}_e$. Similarly, let $\mathscr{P}_d$ be the given list of candidate paths in the upper layer realizing demand d and let $\mathcal{L}_{ds}$ in equation (7.82) be a length of the shortest path for demand d in situation s in this list with respect to costs $\boldsymbol{\gamma}^* = (\gamma^*_{es} : e \in \mathcal{E}, s \in \mathcal{S})$:

$$\mathcal{L}_{ds} = \min_{p \in \mathcal{P}_d} \left(\sum_e \delta_{edp} \gamma^*_{es} \right), \qquad \forall d \in \mathcal{D}, \forall s \in \mathcal{S}. \tag{7.85}$$

The set $\mathcal{P}_d$ contains indices of paths in set $\mathscr{P}_d$. We note that the two properties (see the discussion for problem MNOM in Section 7.1) regarding dual variables $\boldsymbol{\beta}$,$\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ hold for this problem as well.

We will now state the proposition underlying the PG algorithm for problem LRRUL. Suppose that for a given network instance of problem LRRUL all allowable paths (in the lower layer) realizing each link e belong to a set $\widehat{\mathscr{K}_e}$ and all allowable paths (in the upper layer) for each demand belong to set $\widehat{\mathscr{P}_d}$. Let $\widehat{\mathscr{P}_{ds}} \subseteq \widehat{\mathscr{P}_d}$ be a subset of paths for demand d that are fully available in situation s. For each upper layer link e let $\wp_e \in \widehat{\mathscr{K}_e}$ be one of the shortest paths in the lower layer among all allowable paths surviving failure situation s, and let $\bar{\ell}_e$ be its cost with respect to the cost metrics $\boldsymbol{\xi}$, i.e.,

$$\bar{\ell}_e = \|\wp_e\|_{\boldsymbol{\xi}} = \min_{k \in \widehat{\mathcal{K}}_e} \left(\sum_g \varphi_{gek} \xi_g \right), \qquad \forall e \in \mathcal{E}, \tag{7.86}$$

where set $\widehat{\mathcal{K}}_e$ contains indices of paths in set $\widehat{\mathscr{K}_e}$. For each demand d and situation s, let $\wp_{ds} \in \widehat{\mathscr{P}_{ds}}$ be one of the shortest paths in the upper layer among all allowable paths surviving failure situation s, and let ℓ_{ds} be its cost with respect to the cost metrics $\boldsymbol{\gamma}^*$, i.e.,

$$\ell_{ds} = \|\wp_{ds}\|_{\boldsymbol{\gamma}^*} = \min_{p \in \widehat{\mathcal{P}}_{ds}} \left(\sum_e \delta_{edp} \gamma^*_{es} \right), \qquad \forall d \in \mathcal{D}, \forall s \in \mathcal{S}, \tag{7.87}$$

where set $\widehat{\mathcal{P}}_{ds}$ contains indices of paths in set $\widehat{\mathscr{P}_{ds}}$. We consider two versions of problem LRRUL:

- *full* problem LRRUL-FULL with the set of candidate paths for each demand d equal to $\widehat{\mathscr{P}_d}$ and the set of candidate paths for each link e equal to $\widehat{\mathscr{K}_e}$

- *limited* problem LRRUL-LIMITED with the set of candidate paths for each demand d equal to $\mathscr{P}_d$, where $\mathscr{P}_d \subseteq \widehat{\mathscr{P}_d}$ for all d, the set of candidate paths for each link e equal to $\mathscr{K}_e$, where $\mathscr{K}_e \subseteq \widehat{\mathscr{K}_e}$ for all e, and $\mathscr{P}_d \subsetneq \widehat{\mathscr{P}_d}$ for at least one demand d, and/or $\mathscr{K}_e \subsetneq \widehat{\mathscr{K}_e}$ for at least one link e.

Proposition 7.3. *Let ($\boldsymbol{f}^*$, $\boldsymbol{x}^*$, $\boldsymbol{z}^*$; π^*,$\boldsymbol{\beta}^*$, $\boldsymbol{\sigma}^*$, $\boldsymbol{\tau}^*$, γ^*) be a saddle point of the Lagrangian for an instance of the limited problem LRRUL-LIMITED with given sets of candidate paths $\mathscr{P}_d$ for all $d \in \mathcal{D}$ and $\mathscr{K}_e$ for all $e \in \mathcal{E}$, in the upper and lower layers, respectively. Let $\wp_e \in \widehat{\mathscr{K}_e} \setminus \mathscr{K}_e$ be a shortest path in the lower layer realizing capacity of link e and let $\bar{\ell}_e$ denote its length (7.86). Similarly, let $\wp_{ds} \in \widehat{\mathscr{P}_{ds}} \setminus \mathscr{P}_d$ be a shortest path in the upper layer realizing demand d in situation s and let ℓ_{ds} denote its length (7.87). Then:*

- *If $\kappa_e = \bar{\ell}_e$ for all $e \in \mathcal{E}$ and $\mathcal{L}_{ds} = \ell_{ds}$ for all $d \in \mathcal{D}$ and $s \in \mathcal{S}$ then any optimal solution to LRRUL-LIMITED is also optimal for LRRUL-FULL, i.e., the solution is optimal in a wider sense. Notice that this is the case when problem LRRUL-FULL has no paths shorter than LRRUL-LIMITED.*

- *Otherwise, the current optimal solution may be possibly improved in any of the following three cases:*

Case 1: *Suppose that for all $e \in \mathcal{E}$ the length of the shortest path in the candidate path list $\mathscr{K}_e$ is equal to $\bar{\ell}_e$, i.e., $\kappa_e = \bar{\ell}_e$. Suppose also that there exists a demand d, situation s and a path $\wp_{ds} \in \widehat{\mathscr{P}_{ds}} \setminus \mathscr{P}_d$ with length $\ell_{ds} < \mathcal{L}_{ds}$. Then including path $\wp_{ds}$ into list $\mathscr{P}_d$ for at least one such demand d and situation s can improve the current optimal solution.*

Case 2: *Suppose that there exists a link e and a path $\wp_e \in \widehat{\mathscr{K}_e} \setminus \mathscr{K}_e$ with length $\bar{\ell}_e < \kappa_e$. Suppose also that there exists a demand d and situation s, for which, if $\wp_e$ is included into $\mathscr{K}_e$ implying the new cost vector $\boldsymbol{\gamma}^{*\prime}$, there exists a path $\wp_{ds} \in \widehat{\mathscr{P}_{ds}} \setminus \mathscr{P}_d$ with length $\ell_{ds} < \mathcal{L}_{ds}$, which is a shortest path for the demand and situation with respect to costs $\boldsymbol{\gamma}^{*\prime}$. Then including path $\wp_e$ into list $\mathscr{K}_e$ for at least on such link e, and including path $\wp_{ds}$ into list $\mathscr{P}_d$, for at least one such demand d and situation s, can improve the current optimal solution.*

Case 3: *Suppose there exists a link e and a path $\wp_e \in \widehat{\mathscr{K}_e} \setminus \mathscr{K}_e$ with length $\bar{\ell}_e < \kappa_e$. Suppose also that, when $\wp_e$ is included into $\mathscr{K}_e$ implying the new cost vector $\boldsymbol{\gamma}^{*\prime}$, for every demand d and situation s the candidate path list $\mathscr{P}_d$ contains at least one shortest (with respect to costs $\boldsymbol{\gamma}^{*\prime}$) path with length $\mathcal{L}_{ds} = \ell_{ds}$. Then including path $\wp_e$ into list $\mathscr{K}_e$, for at least one such link e, can improve the current optimal solution.*

The proof of the above proposition is analogous to proof of Proposition 5.1 in Section 5.2, and therefore it is omitted here. Algorithm to resolve LRUL using PG is based on this observation and is given in the next section.

7.3.1 Algorithm for resolving LRUL using PG

This iterative algorithm is based on the algorithm presented in Section 6.6.1. The main difference is that Step 2 includes a special (extended) test, based on the values of dual variables, checking the optimality of solution according to Proposition 7.3.

Resolution algorithm for LRRUL using PG.

Step 0:
Put $n := 0$, $Z_0 := \emptyset$, $Z_{PG} := \emptyset$, $Z_1 := \{1, 2, \ldots, S\}$, $t_s := 0$ for all s. Form the initial instance of the LRRUL problem, using small sets of candidate paths (e.g., single paths).

Step 1:
Solve LRRUL problem (If Step 1 is entered from Step 2, then use the optimal solution from the previous iteration as a staring point for solving the new instance of the problem).

Let $t^*, \boldsymbol{f}^*, \boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*$ be the optimal solution of the current instance of the problem. Get values of dual variables from a LP solver. Let (t^*, $\boldsymbol{f}^*$, $\boldsymbol{x}^*$, $\boldsymbol{z}^*$; π^*,$\boldsymbol{\beta}^*$, $\boldsymbol{\lambda}^*$, $\boldsymbol{\sigma}^*$, $\boldsymbol{\tau}^*$, γ^*) be a saddle point of the Lagrangian for the problem.

Step 2:
Run the following test :

- For each link e run a shortest path algorithm, using $\boldsymbol{\xi}$ as L1 links' costs, to find a shortest path $\wp_e$ and its cost $\bar{\ell}_e$. If there exists a path $\wp_e$ such that $\bar{\ell}_e < \kappa_e$, then include it to $\mathscr{K}_e$, form a new instance of LRRUL problem and **Go to Step 1**. Otherwise, proceed with the following:

- Put $Z := \{s \in Z_1 : \lambda_s^* > 0\}$ and $t_s := t^*$ for each $s \in Z$. Put $Z_1 := Z_1 \setminus Z$.

- For each $s' \in Z$ and each demand d run a shortest path algorithm, using $\boldsymbol{\gamma^*}$ as L2 links' costs, to find a shortest path $\wp_{ds'}$ and its cost $\ell_{ds'}$. If there exists a path $\wp_{ds'}$ such that $\ell_{ds'} < \boldsymbol{\mathcal{L}}_{ds'}$ and the same path $\wp_{ds'}$ has not been included yet to $\mathscr{P}_d$ for another situation $s' \in Z$, then include it to $\mathscr{P}_d$. If the path was already included while considering other s', skip it, but in Step 3 consider the current situation, as if the new path was included for it.

Step 3:
Put $n := n + 1$, $Z_{PG} := \{s \in Z$: new paths $\wp_{ds}$ for s have been added$\}$, $Z_1 := Z_1 \cup Z_{PG}$, $Z_0 := Z_0 \cup (Z \setminus Z_{PG})$ and form a new instance of the problem with the extended lists of available paths. **Go to Step 1**.

The entities used in the algorithm are analogous to those presented in Sections 5.3.5 and 5.4.1.

7.4 Path generation for RBL

Proceeding the same way like for problems LRRLL and LRRUL, constraint (6.38) is rewritten as two constraints (7.63)-(7.64), logarithmic function is linearized and constraints (6.36) and (6.39) are merged into the following single constraint:

$$\sum_d \sum_p \delta_{edp} x_{dps} \le \sum_k z_{eks}\,, \qquad \forall e \in \mathcal{E},\ \forall s \in \mathcal{S}. \tag{7.88}$$

Also, constraints required by the algorithm for resolving problem RBL (see Section 5.3.5.4) are added. The linearized, relaxed and extended problem LRRBL is then defined as follows:

LRRBL:

$$\max \qquad t \tag{7.89}$$

s.t.:

$$[\pi \ge 0] \qquad \sum_g \xi_g u_g \le B, \tag{7.90}$$

$$[\sigma_{ds} \ge 0] \qquad h_{ds} \le \sum_p x_{dps}, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{7.91}$$

$$[\tau_{ds} \ge 0] \qquad \sum_p x_{dps} \le H_{ds}, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{7.92}$$

$$[\gamma_{es} \ge 0] \qquad \sum_d \sum_p \delta_{edp} x_{dps} \le \sum_k z_{eks}, \qquad \forall e \in \mathcal{E}, s \in \mathcal{S} \tag{7.93}$$

$$[\mu_{gs} \ge 0] \qquad \sum_e \sum_k \varphi_{gek} z_{eks} \le \alpha_{gs} u_g, \qquad \forall g \in \mathcal{G},\ \forall s \in \mathcal{S} \tag{7.94}$$

$$[\beta_{dsi} \geq 0] \qquad f_{ds} \leq a_i \sum_p x_{dps} + b_i, \qquad \forall d \in \mathcal{D}, s \in \mathcal{S}, \forall i \in \mathcal{I} \tag{7.95}$$

$$[\lambda_s \geq 0] \qquad t_s - \sum_d w_{ds} f_{ds} \leq 0, \qquad \forall s \in Z_0 \tag{7.96}$$

$$[\lambda_s \geq 0] \qquad t - \sum_d w_{ds} f_{ds} \leq 0, \qquad \forall s \in Z_1 \tag{7.97}$$

$$t, f_{ds} \in \mathbb{R},\ x_{dps} \in \mathbb{R}^+, \qquad \forall d \in \mathcal{D}, p \in \mathcal{P}_d, \forall s \in \mathcal{S} \tag{7.98}$$

$$z_{eks} \in \mathbb{R}^+, \qquad \forall e \in \mathcal{E}, k \in \mathcal{K}_e, \forall s \in \mathcal{S}, \tag{7.99}$$

$$u_g \in \mathbb{R}^+, \qquad \forall g \in \mathcal{G}. \tag{7.100}$$

In the same way as for the other problems presented, dual variables associated with each of constraints (7.90)-(7.97) are indicated in brackets in front of each constraint. Observe, that dual variables $\boldsymbol{\lambda}$ are defined for both constraints (7.96) and (7.97), since $Z_0 \cap Z_1 = \emptyset$. As before, for convenience we transform objective (7.33) into minimization. The Lagrangian function for problem LRRBL is:

$$\begin{aligned}
L(t, \boldsymbol{f}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}\ ;\ \pi, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{\mu}) &= \Big(\sum_{s \in Z_1} \lambda_s - 1\Big) t \\
&+ \sum_d \sum_s \Big(\sum_i \beta_{dsi} - w_{ds}\lambda_s\Big) f_{ds} \\
&+ \sum_d \sum_p \sum_s \Big(\tau_{ds} - \sigma_{ds} - \sum_i a_i \beta_{dsi} + \sum_e \delta_{edp}\gamma_{es}\Big) x_{dps} \\
&+ \sum_e \sum_k \sum_s \Big(\sum_g \varphi_{gek}\mu_{gs} - \gamma_{es}\Big) z_{eks} \\
&+ \sum_g \Big(\xi_g \pi - \sum_s \alpha_{gs}\mu_{gs}\Big) u_g \\
&+ \sum_{s \in Z_0} \lambda_s t_s + \sum_d \sum_s \Big(h_{ds}\sigma_{ds} - H_{ds}\tau_{ds} - \sum_i b_i \beta_{dsi}\Big) - \pi B.
\end{aligned} \tag{7.101}$$

Let t^*,$\boldsymbol{f}^*$,$\boldsymbol{x}^*$,$\boldsymbol{z}^*$,$\boldsymbol{u}^*$ be any primal optimal solution of LRRBL. Similarly, let π^*, $\boldsymbol{\beta}^*$, $\boldsymbol{\lambda}^*$, $\boldsymbol{\sigma}^*$, $\boldsymbol{\tau}^*$, $\boldsymbol{\gamma}^*$, $\boldsymbol{\mu}^*$ be any dual optimal solution. The primal and dual optimal solutions form the saddle point of the Lagrangian (7.101). (The dual optimal multipliers together with optimal primal variables are available from an LP solvers.) It can be observed that for any saddle point of the Lagrangian the following optimality conditions hold implied by the complementary slackness property:

$$\sum_{s \in Z_1} \lambda_s^* = 1 \tag{7.102}$$

$$\sum_i \beta_{dsi}^* = w_{ds} \lambda_s^* , \qquad \forall d \in \mathcal{D}, \forall s \in \mathcal{S} \tag{7.103}$$

$$\sigma_{ds}^* - \tau_{ds}^* + \sum_i a_i \beta_{dsi}^* = \mathcal{L}_{ds} , \qquad \forall d \in \mathcal{D}, \forall s \in \mathcal{S} \tag{7.104}$$

$$\gamma_{es}^* = \kappa_{es} , \qquad \forall e \in \mathcal{E}, \forall s \in \mathcal{S} \tag{7.105}$$

$$\sum_s \alpha_{gs} \mu_{gs}^* = \xi_g \pi^* , \qquad \forall g \in \mathcal{G}. \tag{7.106}$$

At optimum, each dual variable μ_{gs}^* can be interpreted as a (dual) cost of link g in situation s.

Let $\mathscr{K}_e$ be the given list of candidate paths in the lower layer realizing capacity of link e and let κ_{es} (in equation 7.105) be a length of the shortest path for link e and situation s in this list with respect to metric $\boldsymbol{\mu}^* = (\mu_{es}^* : e \in \mathcal{E}, s \in \mathcal{S})$:

$$\kappa_{es} = \min_{k \in \mathcal{K}_e} \left(\sum_g \varphi_{gek} \mu_{gs}^* \right), \qquad \forall e \in \mathcal{E}, \forall s \in \mathcal{S}. \tag{7.107}$$

The set $\mathcal{K}_e$ contains indices of paths in set $\mathscr{K}_e$. Similarly, let $\mathscr{P}_d$ be the given list of candidate paths in the upper layer realizing demand d and let $\mathcal{L}_{ds}$ in equation (7.104) be a length of the shortest path for demand d in situation s in this list with respect to costs $\boldsymbol{\kappa} = (\kappa_{es} : e \in \mathcal{E}, s \in \mathcal{S})$:

$$\mathcal{L}_{ds} = \min_{p \in \mathcal{P}_d} \left(\sum_e \delta_{edp} \gamma_{es}^* \right) = \min_{p \in \mathcal{P}_d} \left(\sum_e \delta_{edp} \kappa_{es} \right), \qquad \forall d \in \mathcal{D}. \tag{7.108}$$

The set $\mathcal{P}_d$ contains indices of paths in set $\mathscr{P}_d$. Similarly to problem LRRLL, it can be observed from (7.106), that real cost ξ_g (scaled by π^*) of each lower layer link is, in general, shared among the situations in which the link survives. Thus we define a set $\mathcal{S}^g$ containing all situations in which link g is fully available, i.e.,

$$\mathcal{S}^g = \{ s \in \mathcal{S} : \alpha_{gs} = 1 \} \qquad \forall g \in \mathcal{G}. \tag{7.109}$$

Then we can rewrite equation (7.106) as follows:

$$\sum_{s\in\mathcal{S}^g} \mu^*_{gs} = \xi_e \pi^*, \qquad \forall g \in \mathcal{G}. \tag{7.110}$$

We note that the two properties (see the discussion for problem MNOM in Section 7.1) regarding dual variables $\boldsymbol{\beta}$,$\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ hold for this problem as well, like for the other two-layer problems we have discussed before.

We will now state the proposition underlying the PG algorithm for problem LRRBL. Suppose that for a given network instance of problem LRRBL all allowable paths (in the lower layer) realizing each link belong to set $\widehat{\mathscr{K}_e}$ and all allowable paths (in the upper layer) for each demand belong to set $\widehat{\mathscr{P}_d}$. Let $\widehat{\mathscr{K}_{es}} \subseteq \widehat{\mathscr{K}_e}$ be a set of lower layer paths for link e that are fully available in situation s. Similarly, let $\widehat{\mathscr{P}_{ds}} \subseteq \widehat{\mathscr{P}_d}$ be a set of upper layer paths for demand d that are fully available in situation s. For each upper layer link e and situation s let $\wp_{es} \in \widehat{\mathscr{K}_{es}}$ be a shortest path in the lower layer among all allowable paths surviving failure situation s, and let $\bar{\ell}_{es}$ be its cost with respect to the cost metrics $\boldsymbol{\mu}^*$, i.e.,

$$\bar{\ell}_{es} = \|\wp_{es}\|_{\boldsymbol{\mu}^*} = \min_{k\in\widehat{\mathcal{K}}_{es}} \left(\sum_g \varphi_{gek}\mu^*_{gs} \right), \qquad \forall e \in \mathcal{E}, \forall s \in \mathcal{S}, \tag{7.111}$$

where set $\widehat{\mathcal{K}}_{es}$ contains indices of paths in set $\widehat{\mathscr{K}_{es}}$. For each demand d and situation s, let $\wp_{ds} \in \widehat{\mathscr{P}_{ds}}$ be a shortest path in the upper layer among all allowable paths for demand d, surviving failure situation s, and let ℓ_{ds} be its cost with respect to the cost metrics $\boldsymbol{\kappa}$, i.e.,

$$\ell_{ds} = \|\wp_{ds}\|_{\boldsymbol{\kappa}} = \min_{p\in\widehat{\mathcal{P}}_{ds}} \left(\sum_e \delta_{edp}\kappa_{es} \right), \qquad \forall d \in \mathcal{D}, \forall s \in \mathcal{S}, \tag{7.112}$$

where set $\widehat{\mathcal{P}}_{ds}$ contains indices of paths in set $\widehat{\mathscr{P}_{ds}}$. We consider two versions of problem LRRBL:

- *full* problem LRRBL-FULL with the set of candidate paths for each demand d equal to $\widehat{\mathscr{P}_d}$ and the set of candidate paths for each link e equal to $\widehat{\mathscr{K}_e}$
- *limited* problem LRRBL-LIMITED with the set of candidate paths for each demand d equal to $\mathscr{P}_d$, where $\mathscr{P}_d \subseteq \widehat{\mathscr{P}_d}$ for all d, the set of candidate paths for each link e equal to $\mathscr{K}_e$, where $\mathscr{K}_e \subseteq \widehat{\mathscr{K}_e}$ for all e, and $\mathscr{P}_d \subsetneq \widehat{\mathscr{P}_d}$ for at least one demand d, and/or $\mathscr{K}_e \subsetneq \widehat{\mathscr{K}_e}$ for at least one link e.

Proposition 7.4. *Let (t^*, $\boldsymbol{f}^*$, $\boldsymbol{x}^*$, $\boldsymbol{z}^*$, $\boldsymbol{u}^*$; π^*,$\boldsymbol{\beta}^*$, $\boldsymbol{\lambda}^*$, $\boldsymbol{\sigma}^*$, $\boldsymbol{\tau}^*$, $\boldsymbol{\gamma}^*$, $\boldsymbol{\mu}^*$) be a saddle point of the Lagrangian for an instance of the limited problem LRRBL-LIMITED with given sets of candidate paths $\mathscr{P}_d$ for all $d \in \mathcal{D}$ and $\mathscr{K}_e$ for all $e \in \mathcal{E}$, in the upper and lower layers, respectively. Let $\wp_{es} \in \widehat{\mathscr{K}_{es}} \setminus \mathscr{K}_e$ be a shortest path in the lower layer realizing capacity of link e in situation s and let $\bar{\ell}_{es}$ denote its length (7.111). Similarly, let $\wp_{ds} \in \widehat{\mathscr{P}_{ds}} \setminus \mathscr{P}_d$ be a shortest path in the upper layer realizing demand d in situation s and let ℓ_{ds} denote its length (7.112). Then:*

- *If $\kappa_{es} = \bar{\ell}_{es}$ for all $e \in \mathcal{E}$, $s \in \mathcal{S}$ and $\mathcal{L}_{ds} = \ell_{ds}$ for all $d \in \mathcal{D}$ and $s \in \mathcal{S}$ then any optimal solution to LRRBL-LIMITED is also optimal for LRRBL-FULL, i.e., the solution is optimal in a wider sense. Notice that this is the case when problem LRRBL-FULL has no paths shorter than LRRBL-LIMITED.*

- *Otherwise, the current optimal solution may be possibly improved in any of the following three cases:*

Case 1: Suppose that for all $e \in \mathcal{E}$ and $s \in \mathcal{S}$ the length of the shortest path in the candidate path list $\mathscr{K}_e$ is equal to $\bar{\ell}_{es}$, i.e., $\kappa_{es} = \bar{\ell}_{es}$. Suppose also that there exists a demand d, situation s and a path $\wp_{ds} \in \widehat{\mathscr{P}_{ds}} \setminus \mathscr{P}_d$ with length $\ell_{ds} < \mathcal{L}_{ds}$. Then including path $\wp_{ds}$ into list $\mathscr{P}_d$ for at least one such demand d and situation s can improve the current optimal solution.

Case 2: Suppose that there exists a link e, situation s and a path $\wp_{es} \in \widehat{\mathscr{K}_{es}} \setminus \mathscr{K}_e$ with length $\bar{\ell}_{es} < \kappa_{es}$. Suppose also that there exists a demand d and situation s, for which, if $\wp_{es}$ is included into $\mathscr{K}_e$ implying the new cost vector $\boldsymbol{\kappa}'$, there exists a path $\wp_{ds} \in \widehat{\mathscr{P}_{ds}} \setminus \mathscr{P}_d$ with length $\ell_{ds} < \mathcal{L}_{ds}$, which is a shortest path for the demand and situation with respect to costs $\boldsymbol{\kappa}'$. Then including path $\wp_{es}$ into list $\mathscr{K}_e$ for at least on such link e and situation s, and including path $\wp_{ds}$ into list $\mathscr{P}_d$, for at least one such demand d and situation s, can improve the current optimal solution.

Case 3: Suppose there exists a link e, situation s and a path $\wp_{es} \in \widehat{\mathscr{K}_{es}} \setminus \mathscr{K}_e$ with length $\bar{\ell}_{es} < \kappa_{es}$. Suppose also that, when $\wp_{es}$ is included into $\mathscr{K}_e$ implying the new cost vector $\boldsymbol{\kappa}'$, for every demand d and situation s the candidate path list $\mathscr{P}_d$ contains at least one shortest (with respect to costs $\boldsymbol{\kappa}'$) path with length $\mathcal{L}_{ds} = \ell_{ds}$. Then including path $\wp_{es}$ into list $\mathscr{K}_e$, for at least one such link e and situation s, can improve the current optimal solution.

The proof of the above proposition is analogous to proof of Proposition 5.1 in Section 5.2, and therefore it is omitted here.

7.4.1 Algorithm for resolving LRRBL using PG

Problem LRRBL can be resolved with the same algorithm as LRUL (see section 7.3.1), modified as follows: i) in Step 1 problem LRRBL should be solved instead of LRRUL; ii) Step 2 should be replaced with the following:

Put $Z := \{s \in Z_1 : \lambda_s^* > 0\}$ and $t_s := t^*$ for each $s \in Z$. Put $Z_1 := Z_1 \setminus Z$. For each $s' \in Z$ run the following PG-test:

- For each link e run a shortest path algorithm, using $\boldsymbol{\mu}^*$ as links' costs in the lower layer, to find a shortest path $\wp_{es'}$ and its cost $\bar{\ell}_{es'}$. If there exists a path $\wp_{es'}$ such that $\bar{\ell}_{es'} < \kappa_{es'}$, and the same path $\wp_{es}$ has not been included yet to $\mathscr{K}_e$ for another situation $s' \in Z$, then include it to the routing list $\mathscr{K}_e$.

- For each demand d run a shortest path algorithm, using $\boldsymbol{\kappa}$ as L2 links' costs, to find a shortest path $\wp_{ds'}$ and its cost $\ell_{ds'}$. If there exists a path $\wp_{ds'}$ such that $\ell_{ds'} < \mathcal{L}_{ds'}$, and the same path $\wp_{ds'}$ has not been included yet to $\mathscr{P}_d$ for another situation $s' \in Z$, then include it to $\mathscr{P}_d$. If the path was already included while considering other s', skip it, but in Step 3 consider the current situation, as if new path was included for it.

7.5 Numerical example

The numerical example presented below illustrates the performance of the PG-based algorithms for the LRRLL, LRRUL and LRRBL problems, presented in the previous sections. We have chosen to illustrate the three PG-based algorithms on network N_{12}, which is described in Section 6.7.1. It is one of the networks used for numerical tests with the algorithms for the three problems without PG. Topologies of the two network layers are illustrated in Figure 6.6. Link costs for the lower layer network are given in Table 6.2 and the considered failure situations are listed in Table 6.3. The network data used for the numerical tests presented in this section is, however, not identical with that described in Section 6.7.1. What is different is the number of initial candidate paths in the routing lists for the demands in both layers. For the experiments presented here we have decreased the initial candidate path lists to only two paths per demand for all demands in both layers. The two paths were two shortest paths. This is due to our intention to illustrate the efficiency of the algorithms with minimal initial routing lists.

Of course, providing large set of good initial candidate paths would decrease the execution time of the PG-based algorithms, but on the other hand would require longer time for generating these paths and resolution of the problem relaxation with large candidate path lists would be more difficult. The investigation of the strategies for forming the initial path lists is out of the scope of this discussion. Thus the results provided in this section should be considered merely as an illustration of the efficiency of algorithms. Budget $B = 1000$ has been assumed for the experiments.

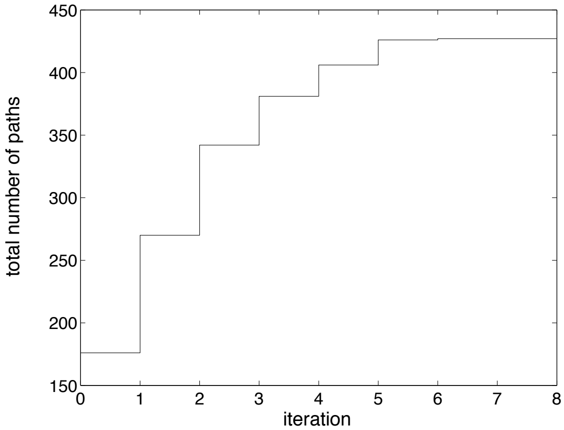
The algorithms have been implemented in AMPL ver 20010215 [119], using CPLEX 7.5 [49] as an optimizer. Tests have been run on a PC with a Pentium III 866MHz CPU and 256 MB RAM memory.

7.5.1 Results

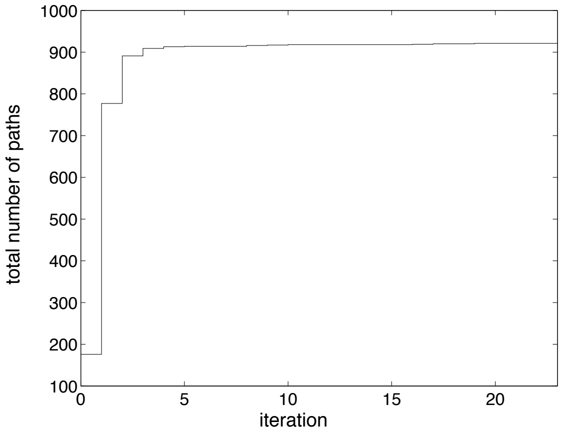
Table 7.1 gives a comparison between different reconfiguration options, using PG. In the table, R_{min} and R_{max} are the initial and final (after path generation) values of the revenue. Number of paths added is total for all demands (and situations) in both layers. *Time* gives a total execution time of the algorithm, including solving LP and running the shortest path algorithm. Table 7.2 presents an analogous comparison between the three options when using the algorithms without PG (presented in Section 6.6), but instead with 2 shortest paths predefined for each demand on each of the layers (thus resulting in 176 total paths). In this way one can compare what solution can be attained for the algorithms with the same minimal initial lists of candidate paths that are used as starting point for the PG-based algorithms. Figure 7.1 shows how the total number of paths (for all demands on both layers) in P^{cand} changes when the algorithms (using PG) progress. Number of paths that have been added in each iteration is indicated in Figure 7.2.

7.6 Discussion

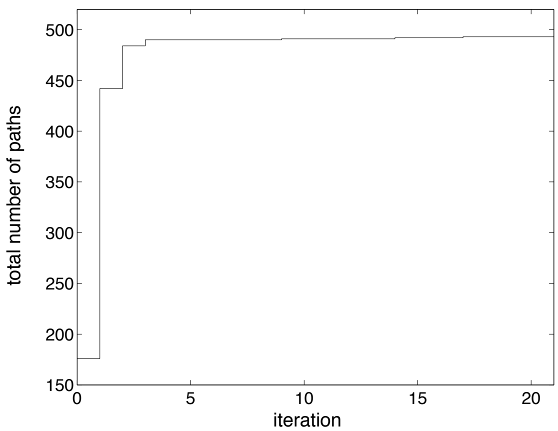
Comparing entries in Tables 7.1 and 7.2 one can see that for a relatively small 12-node network, the algorithms with PG take much longer time than those without PG. This is mostly caused by running the shortest path algorithm many times. Still, much better solutions (in terms of the objective function value) are reached by the algorithms using PG (see row R_{max} in the tables). Time-efficiency of PG algorithms is a tradeoff between solving a more difficult problem (with large lists of candidate paths) and solving multiple shortest path problems. Also, one has to take into account, that when using algorithms without PG, pre-calculation of paths is necessary, which can be time consuming for large network instances. Notice that when using PG, most of the shortest paths are added to the routing



(a) LRLL

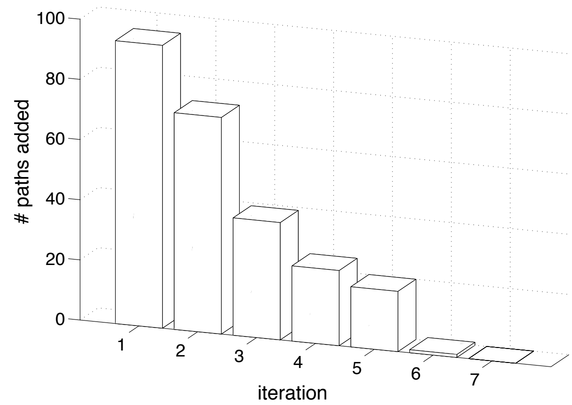


(b) LRUL

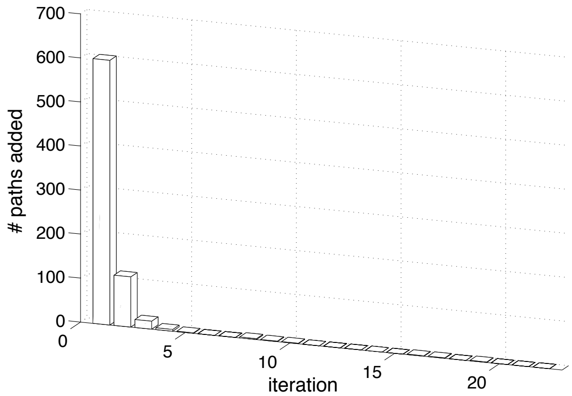


(c) LRBL

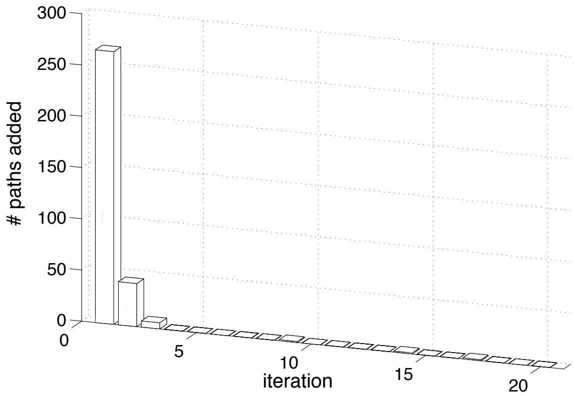
Figure 7.1: Total number of paths for the three reconfiguration options.



(a) LRLL



(b) LRUL



(c) LRBL

Figure 7.2: A number of paths added in each iteration for the three reconfiguration options.

Formulation	LRLL	LRUL	LRBL
R_{min}	-88,46	2,72	21,6
R_{max}	19,4	77,94	91,00
# iterations	7	22	20
# paths added	251	745	317
Time, [min]	5	38	14

Table 7.1: Selected results for different reconfiguration options, using PG.

Formulation	LRLL	LRUL	LRBL
R_{min}	-88,46	2,72	21,6
R_{max}	-88,46	36,22	49,95
# iterations	-	10	15
# predefined paths	176	176	176
Time, [min]	0,05	3	4

Table 7.2: Selected results for different reconfiguration options with predefined routing lists.

lists during first few iterations (see Figure 7.1). This fact can be used as a stopping criterion allowing for early stop of the algorithms after the first few iterations. As a result, execution times of the algorithms would considerably decrease at the expense of marginally worse objective function value.

7.7 Summary of the results

In this section we have discussed how to apply path generation (PG) to the four multi-layer network design problems through the use of dual variables as link/path costs. This interpretation forms a basis for implementing the PG algorithms to solve the problems. A numerical example gives a comparison of the efficiency of the algorithms for different reconfiguration options, as well as how do they compare to algorithms without PG. It is shown that algorithms with PG allow to reach higher revenue values and do not require long predefined routing lists. An approach used to design the algorithms could be easily adapted to solve networks with more layers or also other similar multi-layer network design problems.

CHAPTER 8

An iterative resolution framework for two-layer network design problems with different recovery mechanisms

8.1 Introduction

In Chapter 5.4.1 we have presented three models for a two layer network with unrestricted flow reconfiguration. What differs in the models is which layer performs recovery of flows after failure. Certainly, the assumed full reconfiguration mechanism is not realistic, since, in general, it assumes that potentially all network flows are disconnected when a failure occurs and then reconnected using the surviving network resources. However the results obtained for full reconfiguration made it possible to compare recovery efficiency in different layers. In this chapter we will present models for several realistic mechanisms for network resiliency, namely, restricted reconfiguration, path diversity, and hot-standby path protection.

In the multi-layer network different recovery mechanisms can be employed in the layers. Thus the number of network models with different combinations of recovery mechanisms is large. Instead of presenting all possible two-layer network models explicitly, we will instead present the models separately for each layer with different recovery mechanisms. The desired two layer model can the be assembled combining the appropriate models for the upper and lower layers.

After presenting models assuming different recovery mechanisms in different network layers we will present explicitly several two layer network design problems, and using them as an example, we will show how the (difficult) realistic network design problems can be resolved within the proposed iterative resolution framework.

8.1.1 Restricted reconfiguration

In contrast to the unrestricted reconfiguration mechanism which we assumed for the models discussed in the previous chapters, restricted reconfiguration (RR) mechanism assures that the flows reconfigured upon a failure are restricted only to those affected by the failure, i.e., non-affected flows stay at place. The models with the RR recovery mechanism that we will consider do not prevent bifurcation of flows. Therefore, the RR policy is applied on a per-path basis, as opposed to per-demand basis. This means that only flows on failing paths of a demand are disconnected and rerouted within the surviving resources. A failing flow can in general become bifurcated when restored on more than one path in a failure situation. Thus the RR mechanism in itself does not impose any restrictions on how many paths the failing flow is restored. Below we present two formulations modeling RR in the upper and lower network layer.

8.1.1.1 RR in the upper layer

Let us assume that the normal state is identified as $s = 0$, and denote flow for each demand d routed on path p in the normal state by x_{dp0}. Then the following problem formulation implies that if a path used to carry the flow in the normal state is not failing in some state $\hat{s}$ then the path is used to carry the flow also in $\hat{s}$. In other words, only the flows on failing paths are rerouted upon the failure, while other flows remain on their paths chosen for the normal state.

The following constraints define a set of feasible solutions for an upper layer network design problem with RR mechanism. In equation (8.3) variables y_{es} are auxiliary and used for connecting models of the upper and lower layers. Their values are implied by the lower and upper layer models together. Considering only the upper layer separately, capacities y_{es} can be assumed constant.

$$h_{ds} \le \sum_p x_{dps} \le H_{ds}\,, \qquad \forall d \in \mathcal{D},\ s \in \mathcal{S} \tag{8.1}$$

$$x_{dps} \ge \theta_{dps} x_{dp0}\,, \qquad \forall d \in \mathcal{D},\ p \in \mathcal{P}_d,\ s = 1, \ldots, S \tag{8.2}$$

$$\sum_d \sum_p \delta_{edp} x_{dps} \le M y_{es}\,, \qquad \forall e \in \mathcal{E},\ s \in \mathcal{S}, \tag{8.3}$$

$$x_{dps} \in \mathbb{R}^+ \qquad \forall d \in \mathcal{D},\ p \in \mathcal{P}_d,\ \forall s \in \mathcal{S} \tag{8.4}$$

$$y_{es} \in \mathbb{Z}^+\,, \qquad \forall e \in \mathcal{E},\ \forall s \in \mathcal{S}. \tag{8.5}$$

Recall, that θ_{dps} is an availability coefficient for path p of demand d in situation s, and is calculated as follows:

$$\theta_{dps} = \prod_{\{e:\delta_{edp}=1\}} \alpha_{es} \ , \qquad \forall d \in \mathcal{D},\ \forall p \in \mathcal{P}_d,\ s = 1, 2, ..., S, \tag{8.6}$$

where $\alpha_{es} \in \{0, 1\}$ is an availability coefficient of link e in situation s. Assuming that only lower layer links may fail, and if a link fails, it fails completely, α_{es} is calculated from the lower layer path availability coefficients θ_{eks} in the following way:

$$\alpha_{es} = \prod_{k \in \mathcal{K}_e} \theta_{eks} \ , \qquad \forall e \in \mathcal{E},\ \forall k \in \mathcal{K}_e,\ s = 1, 2, ..., S. \tag{8.7}$$

Coefficients θ_{eks} are analogous to θ_{dps}, but are calculated using availability coefficients of lower layer links, α_{gs}:

$$\theta_{eks} = \prod_{\{g:\varphi_{gek}=1\}} \alpha_{gs} \ , \qquad \forall e \in \mathcal{E},\ \forall k \in \mathcal{K}_e,\ s = 1, 2, ..., S. \tag{8.8}$$

Note that bounds h_{ds} and H_{ds} define the minimal, respectively maximal, number of demand modules that can be allocated to demand d. Thus, the bounds are not defined in terms of (absolute) bandwidth, but in terms of number of demand modules of size L.

8.1.1.2 RR in the lower layer

RR can also be assumed for recovery in the lower network layer. Similarly, as for the upper layer model discussed above, we assume that z_{ek0} denotes a flow realizing capacity of e on path k in the normal state. Then a set of feasible solutions for the lower layer network design problem with RR mechanism is defined by the following constraints:

$$\sum_k z_{eks} \;\geq\; y_{es} \;, \qquad \forall e \in \mathcal{E} \;, \forall k \in \mathcal{K}_e, \; \forall s \in \mathcal{S} \tag{8.9}$$

$$z_{eks} \;\geq\; \theta_{eks} z_{ek0} \;, \qquad \forall e \in \mathcal{E} \;, \forall k \in \mathcal{K}_e, \; s \in \mathcal{S} \setminus 0 \tag{8.10}$$

$$\sum_e \sum_k \varphi_{gek} z_{eks} \;\leq\; \alpha_{gs} N u_g \;, \quad \forall g \in \mathcal{G}, \; \forall s \in \mathcal{S} \tag{8.11}$$

$$\sum_g \xi_g u_g \;\leq\; B \;. \tag{8.12}$$

$$z_{eks}, y_{es} \in \mathbb{Z}^+ \;, \qquad \forall e \in \mathcal{E}, \; \forall k \in \mathcal{K}_e, \; \forall s \in \mathcal{S}. \tag{8.13}$$

$$u_g \in \mathbb{Z}^+ \qquad \forall g \in \mathcal{G}. \tag{8.14}$$

$$\tag{8.15}$$

In equation (8.9) variables y_{es} are used for connecting lower and upper layer network models. Coefficients θ_{eks} in (8.10) are calculated from (8.8).

8.1.2 Path diversity

A simple way to increase network resilience is by using Path diversity (PD) mechanism. Strictly speaking, PD is not a recovery mechanism—it merely implements the idea of deliberately splitting the whole demand volume into a (at least) a predefined number (n_d) of paths. If the candidate paths are link (node) disjoint then in a case of a single link (node) failure at most $100/n_d\%$ of the demand d volume is lost and the rest of it survives. An advantage of the approach is that (assuming that in each state a single link fails) only a part of the flow can be affect by failure and lost, while the rest of it survives. Thus, a single failure will never affect the whole demand volume. Besides, no recovery actions need to be taken when a link fails, this way simplifying the network equipment.

8.1.2.1 PD in the upper layer

The following constraints define a set of feasible solutions for the upper layer network design problem with PD. As before, variables y_e are auxiliary and can be assumed constant if this problem is considered separately (not in the model with both layers).

$$\lceil h_d/n_d \rceil \;\le\; x_{dp} \;\le\; \lceil H_d/n_d \rceil\,, \qquad \forall d \in \mathcal{D}\,, \forall p \in \mathcal{P}_d \tag{8.16}$$

$$h_d \;\le\; \sum_p x_{dp} \;\le\; H_d\,, \qquad \forall d \in \mathcal{D} \tag{8.17}$$

$$\sum_d \sum_p \delta_{edp} x_{dp} \;\le\; M y_e\,, \qquad \forall e \in \mathcal{E}. \tag{8.18}$$

$$x_{dp} \in \mathbb{R}^+ \qquad \forall d \in \mathcal{D},\; p \in \mathcal{P}_d \tag{8.19}$$

$$y_e \in \mathbb{Z}^+ \qquad \forall e \in \mathcal{E}. \tag{8.20}$$

$$\tag{8.21}$$

Constraint (8.16) implies that a flow allocated to any path of a demand must be at least as large as $1/n_d$ of the lower bound h_d, and at most as large as $1/n_d$ of the upper bound H_d. Other constraints have been previously discussed.

8.1.2.2 PD in the lower layer

PD in the lower is imposed by introducing a split factor n_e and requiring that any flow z_{ek} realizing capacity of an upper layer link e can carry at most $1/n_e$ fraction of the link capacity y_e. This is imposed by constraint (8.23). The constraints below define a set of feasible solutions for the lower layer network design problem with PD.

$$\sum_k z_{ek} \;\ge\; y_e\,, \qquad \forall e \in \mathcal{E}\,, \forall k \in \mathcal{K}_e \tag{8.22}$$

$$z_{ek} \;\le\; \lceil y_e/n_e \rceil\,, \qquad \forall e \in \mathcal{E},\; \forall k \in \mathcal{K}_e \tag{8.23}$$

$$\sum_e \sum_k \varphi_{gek} z_{ek} \;\le\; N u_g\,, \qquad \forall g \in \mathcal{G} \tag{8.24}$$

$$\sum_g \xi_g u_g \;\le\; B \tag{8.25}$$

$$z_{ek}, y_e \in \mathbb{Z}^+ \qquad \forall e \in \mathcal{E},\; k \in \mathcal{P}_e \tag{8.26}$$

$$u_g \in \mathbb{Z}^+ \qquad \forall g \in \mathcal{G}. \tag{8.27}$$

Note that a non-linear constraint (8.23) can, for practical purposes, be slightly modified and replaced by the linear constraint

$$z_{ek} \;\le\; (y_e/n_e) + 1\,, \qquad \forall e \in \mathcal{E},\; \forall k \in \mathcal{K}. \tag{8.28}$$

$$\tag{8.29}$$

8.1.3 Hot stand-by path protection

When a very high availability of resources is required, a hot stand-by (HS) path protection is used. It implies, that for each demand d, a primary path $\mathcal{P}_{dp}$ is set up, as well as a path $\mathcal{R}_{dp}$ protecting $\mathcal{P}_{dp}$. Identical resources are reserved on the hot-standby (protection) path, as those needed on primary path. So if the primary path fails, the traffic is automatically switched to the protection path, thus assuring 100% availability of the resources. The protection mechanism is analogous to $1:1$, or $1+1$ protection (the model is the same for both).

If the two paths, primary and protection, are chosen in such a way, that any failure which affects the primary path $\mathcal{P}_{dp}$ leaves the protection path $\mathcal{R}_{dp}$ intact, the paths are called *failure-disjoined*.

8.1.3.1 HS in the upper layer

The following constraints define a set of feasible solutions for the upper layer network design problem with HS path protection.

$$\sum_p v_{dp} \;=\; 1\,, \qquad \forall d \in \mathcal{D}\,, \forall p \in \mathcal{P}_d \tag{8.30}$$

$$v_{dp}h_d \;\leq\; x_{dp} \;\leq\; v_{dp}H_d\,, \qquad \forall d \in \mathcal{D}\,, \forall p \in \mathcal{P}_d \tag{8.31}$$

$$\sum_d \sum_p (\delta_{edp} + \hat{\delta}_{edp})x_{dp} \;\leq\; My_e\,, \quad \forall e \in \mathcal{E} \tag{8.32}$$

$$v_{dp} \in \mathbb{B}\,, x_{dp} \in \mathbb{R}^+\,, \qquad \forall d \in \mathcal{D},\; p \in \mathcal{P}_d \tag{8.33}$$

$$z_{ek}, u_g \in \mathbb{Z}^+\,, \qquad \forall e \in \mathcal{E},\; \forall k \in \mathcal{K}_e,\; \forall g \in \mathcal{G}. \tag{8.34}$$

In the constraints above $p = 1, 2, ..., P_d$ indexes candidate pairs of (primary, protection) paths $(\mathcal{P}_{dp}, \mathcal{R}_{dp})$ for realizing demand d. Variable v_{dp} is a binary variable which is equal to one only if the path pair p is chosen for realizing demand d, and zero otherwise. Coefficients δ_{edp} and $\hat{\delta}_{edp}$ in constraint (8.32) have the following meaning:

δ_{edp} $= 1$ if link e belongs to the primary path $\mathcal{P}_{dp}$ realizing demand d; 0, otherwise

$\hat{\delta}_{edp}$ $= 1$ if link e belongs to the protection path $\mathcal{R}_{dp}$, protecting path p of demand d; 0, otherwise.

The sum $\vartheta_{edp} = \delta_{edp} + \hat{\delta}_{edp}$ can be defined as:

$$\vartheta_{edp} = \begin{cases} 1, \text{if} \quad e \in \mathcal{P}_{dp} \cup \mathcal{R}_{dp} \setminus (\mathcal{P}_{dp} \cap \mathcal{R}_{dp}) \\ 2, \text{if} \quad e \in \mathcal{P}_{dp} \cap \mathcal{R}_{dp} \\ 0, \text{otherwise} \end{cases}$$

and directly used in constraints (8.32).

The formulation (8.30)-(8.34) can be modified by taking the following natural assumption into account:

Assumption: Since the capacity of the upper (IP/MPLS) layer links is virtual, assume that if primary path and its hot-standby protection path use the same (not-failed) link, then the capacity on that link is not reserved twice.

This is done by defining the link-path coincidence coefficient in yet another way, as follows:

$$\theta_{edp} = \begin{cases} 1, \text{if} \quad e \in \mathcal{P}_{dp} \cup \mathcal{R}_{dp} \\ 0, \text{otherwise.} \end{cases} \tag{8.35}$$

Then the constraints (8.32) can be replaced by the following:

$$\sum_d \sum_p \theta_{edp} x_{dp} \ \leq \ M y_e \ , \ e = 1, 2, \ldots, E. \tag{8.36}$$

In constraint (8.31) we use the so called “big M”. This “big M” is equal to H_d in (8.31). In fact, in the classical unsplittable flow formulations "big M" is not necessary since the demand volume (H_d) is fixed for each demand, and the path flows for demand d can be simply expressed as $x_{dp} = H_d \cdot v_{dp}, p = 1, 2, \ldots, P_d$. In our case, however, the actual demand volume is a variable (X_d) and the corresponding path flows are equal to $x_{dp} = X_d \cdot v_{dp}, p = 1, 2, \ldots, P_d$ which introduces a non-linearity if used directly. Therefore, the use of the forcing constraint (8.31) seems necessary. This is in fact not advantageous since using "big M" decreases the lower bounds provided by the linear relaxation of formulation (8.30)-(8.34), and hence can make the branch-and-bound approach less effective.

8.1.3.2 HS in the lower layer

HS path protection in the lower layer is modeled in the same way as the protection in the upper layer. We assume that $k = 1, 2, ..., K_e$ indexes candidate pairs of (primary, protection) paths $(\mathcal{P}_{ek}, \mathcal{R}_{ek})$ in the lower layer for realizing capacity of link e. For each link e and path k a binary variable v_{ek} is introduced, which is equal to one only if the path pair k is chosen for realizing capacity of link e, and zero otherwise. A set of feasible solutions for the lower layer network design problem assuming HS path protection is defined by the following constraints:

$$\sum_k v_{ek} = 1\,, \qquad \forall e \in \mathcal{E} \tag{8.37}$$

$$\sum_k z_{ek} \leq Y v_{ek}\,, \qquad \forall e \in \mathcal{E}\,, \forall k \in \mathcal{K}_e \tag{8.38}$$

$$\sum_k z_{ek} \geq y_e\,, \qquad \forall e \in \mathcal{E} \tag{8.39}$$

$$\sum_e \sum_k (\varphi_{gek} + \hat{\varphi}_{edp}) z_{ek} \leq N u_g\,, \quad \forall g \in \mathcal{G} \tag{8.40}$$

$$v_{ek} \in \mathbb{B}, z_{ek}, y_e, u_g \in \mathbb{Z}^+\,, \qquad \forall e \in \mathcal{E},\ \forall k \in \mathcal{K}_e,\ \forall g \in \mathcal{G}. \tag{8.41}$$

Link-path coincidence coefficients φ_{edp} and $\hat{\varphi}_{edp}$ in constraint (8.40) have the following meaning:

φ_{gek} $= 1$ if link g belongs to the primary path $\mathcal{P}_{ek}$; 0, otherwise

$\hat{\varphi}_{gek}$ $= 1$ if link g belongs to the protection path $\mathcal{R}_{ek}$ protecting path k of demand e; 0, otherwise

The constraint (8.40) can be written in a more compact form aggregating link-path coincidence coefficients in the same way as we did for the upper layer design problem.

Like for the upper layer design problem with HS path protection, in this formulation we use the so called “big M”. This “big M” is equal to Y in (8.38). Here we assume Y to be a number which is at least as large as the maximal value of y_e among all the links e.

8.2 Network design problems

We will now present a resilient network design problem for capacity dimensioning and flow allocation in the two-layer network with modular capacities in both layers. Several problem modifications are discussed, differing by objective function and recovery mechanism used. The three most often encountered objective functions– total network throughput maximization, network installation cost minimization (CM) and fair flow allocation– are considered, as well as two different mechanisms for recovery from failures occurring in the lower layer. These are protection via path diversity in the lower layer and hot-standby path protection in the upper layer. It is assumed that network topology, user demands and costs of different entities are given. For some of the problems, also the budget for installing the network is given.

By presenting this wide range of network design problems we wish to illustrate that all of them can be effectively resolved by the presented generic resolution framework.

We start by presenting two NDPs, each using a different recovery mechanism. For this we assume that the objective function in these problems is proportionally fair bandwidth sharing among the demands. Further we assume that hot-standby path protection is to be used for recovery and present two more NDPs, with TM and CM objectives.

8.3 PF bandwidth sharing and protection via path diversity in DWDM layer

The network design problem presented below consists in finding PF flow allocations in the upper (IP/MPLS) layer and in the lower (DWDM) layer, as well as link capacities in both layers, given the budget for installing capacity in the lower layer links. The following requirements are taken into account in the considered network design problem: 1) flow realizing each demand must be realized on a single path in the upper layer; 2) capacities of the links are installed in modules of a fixed size; 3) the available capacity is shared between demands according to the PF principle; 4) protection of flows in case of failures is assured in the lower layer using path diversity. The model for PD is the one presented in Section 8.1.2.2. The MIP formulation is as follows:

Problem PF-PD: a two layer network design problem with PF flow allocation and protection via path diversity in DWDM layer.

$$\max \qquad R = \sum_d w_d \log(X_d) \tag{8.42}$$

$$\text{s.t.:} \qquad \sum_g \xi_g u_g \;\le\; B\,, \tag{8.43}$$

$$\sum_p x_{dp} \;=\; X_d\,, \qquad \forall d \in \mathcal{D} \tag{8.44}$$

$$\sum_p v_{dp} \;=\; 1\,, \qquad \forall d \in \mathcal{D} \tag{8.45}$$

$$v_{dp}h_d \;\leq\; x_{dp} \;\leq\; v_{dp}H_d\,, \qquad \forall d \in \mathcal{D}\,, \forall p \in \mathcal{P}_d \tag{8.46}$$

$$\sum_d \sum_p \delta_{edp} x_{dp} \;\leq\; M y_e\,, \qquad \forall e \in \mathcal{E} \tag{8.47}$$

$$\sum_k z_{ek} \geq y_e\,, \qquad \forall e \in \mathcal{E} \tag{8.48}$$

$$z_{ek} \;\leq\; \lceil y_e/n_e \rceil\,, \qquad \forall e \in \mathcal{E},\; \forall k \in \mathcal{K}_e \tag{8.49}$$

$$\sum_e \sum_k \varphi_{gek} z_{ek} \;\leq\; N u_g\,, \qquad \forall g \in \mathcal{G} \tag{8.50}$$

$$v_{dp} \in \mathbb{B},\; x_{dp} \in \mathbb{R}^+\,, \qquad \forall d \in \mathcal{D},\; \forall p \in \mathcal{P}_d \tag{8.51}$$

$$z_{ek}, y_e, u_g \in \mathbb{Z}^+\,, \qquad \forall e \in \mathcal{E},\; \forall k \in \mathcal{K}_e,\; \forall g \in \mathcal{G}. \tag{8.52}$$

In the above formulation we use the so called "big M" to force unsplittable flows in the upper layer (this "big M" is equal to H_d in constraint (8.46)).

Constraints (8.44) give the total amount of flow, assigned for each demand d, which is then used in the objective function. Constraints (9.6)-(8.46) assure that only one path p (LSP) is used to realize flow for each demand d. At the same time constraints (8.46) impose lower and upper bound for the the flow realizing each demand d. We would like to emphasize the role of the upper and lower bounds. The upper bounds H_d serve as a means for limiting excessive allocations of network bandwidth to demands (in order to avoid cases when the reserved bandwidth will never be used). Similarly, the lower bounds h_d do not allow for too small bandwidth allocations and provide specific guarantees for the demands. Note that assigning the same value to lower and upper bound of any demand implies fixed bandwidth allocation to the demand.

Constraints (8.47) assure that all the flows routed on each link e do not exceed its capacity which is allocated in modules of size M. Modularity of the links is introduced in order to have modular demand volumes for the lower layer. All flows x_{dp} of the MPLS layer traversing a given link e sum up to the load of the link and imply its capacity My_e. Then the number of modules y_e of a given link e forms a demand volume for the DWDM layer, which has to be realized by the DWDM layer flows z_{ek}. Constraints (8.49) force the demand volumes y_e to be split into at least n_e paths in the DWDM layer. Constraints (8.50), similarly to (8.47), assure that all the flows routed on each physical link g do not exceed its capacity allocated in modules of size N. Finally, constraint (8.43) imposes budget B for installing capacity in the lower layer links (fibers).

Note that non-linear constraints (8.49) can, for practical purposes, be slightly changed and replaced by linear constraints

$$z_{ek} \le (y_e/n_e) + 1 , \quad \forall e \in \mathcal{E} , \forall k \in \mathcal{K}_e. \tag{8.53}$$

The meaning of capacity modules is discussed in Section 3.2.3.

8.3.1 Protection mechanism

As it was already mentioned, protection is achieved by splitting flows of the lower layer z_{ek} (realizing capacities y_e) into a predefined number (n_e) of diverse paths. If candidate paths for DWDM layer flows are link (node) disjoint then in a case of a single link (node) failure at most $100/n_e\%$ of the demand e volume is lost and the rest of it survives.

Main advantages of having recovery mechanisms in the lower layer are low number of recovery actions needed (since protection/restoration is performed on the coarsest granularity), as well as the failure is not propagated to the upper layer. A disadvantage of such a scheme, though, is that failures in the upper layer may not be resolved [24].

8.4 PF bandwidth sharing and hot-standby path protection in IP/MPLS layer

The following network design problem models a network scenario where single path routing and hot-standby path protection is used in the IP/MPLS layer, no resilience is implemented in the DWDM layer, and capacities are installed in modules in both layers. Bifurcation of flows in the DWDM layer is allowed. In the formulation below we use the model for HS protection in the upper layer presented in Section 8.1.3.1. Most of the entities used in this formulation are the same as for problem PF-PD presented before. Note, however, that here p indexes a pair of paths in the upper layer, i.e., a normal and protection path.

Problem PF-HS: a two layer network design problem with PF flow allocation and hot-standby path protection in IP/MPLS layer.

$$
\begin{aligned}
\max \quad & R = \sum_d w_d \log(X_d) && && (8.54)\\
\text{s.t.:} \quad & \sum_g \xi_g u_g \;\le\; B\,, && && (8.55)\\
& \sum_p x_{dp} \;=\; X_d\,, && \forall d \in \mathcal{D} && (8.56)\\
& \sum_p v_{dp} \;=\; 1\,, && \forall d \in \mathcal{D} && (8.57)\\
& v_{dp} h_d \;\le\; x_{dp} \;\le\; v_{dp} H_d\,, && \forall d \in \mathcal{D}\,, \forall p \in \mathcal{P}_d && (8.58)\\
& \sum_d \sum_p \theta_{edp} x_{dp} \;\le\; M y_e\,, && \forall e \in \mathcal{E} && (8.59)\\
& \sum_k z_{ek} \ge y_e\,, && \forall e \in \mathcal{E} && (8.60)\\
& \sum_e \sum_k \varphi_{gek} z_{ek} \;\le\; N u_g\,, && \forall g \in \mathcal{G}. && (8.61)\\
& v_{dp} \in \mathbb{B},\; x_{dp} \in \mathbb{R}^+\,, && \forall d \in \mathcal{D},\; \forall p \in \mathcal{P}_d && (8.62)\\
& z_{ek}, y_e, u_g \in \mathbb{Z}^+\,, && \forall e \in \mathcal{E},\; \forall k \in \mathcal{K}_e,\; \forall g \in \mathcal{G}. && (8.63)
\end{aligned}
$$

Most of the constraints are the same as for the problem PF-PD. Just the constraints (8.59) are different. These constraints assure that all the flows routed on each link e, including the bandwidth reserved for the hot-standby paths, do not exceed the capacity of link e, which is allocated in modules of size M. Coefficient θ_{edp} is defined in (8.35). Note, that in this formulation, constraints (8.57)-(8.58) assure that only one path pair p is used to realize the total flow for each demand d. Thus a single primary path $\mathcal{P}_{dp}$ is used to carry all the flow of demand d, and only a single protection path $\mathcal{R}_{dp}$ is used to protect the primary path against all (foreseen) failures.

The meaning of capacity modules is discussed in Section 3.2.3).

8.4.1 Protection mechanism and failure-disjoint paths

This problem formulation employs hot-standby path protection on IP/MPLS layer. It implies, that for each demand d, a primary path $\mathcal{P}_{dp}$ is set up, as well as a path $\mathcal{R}_{dp}$ protecting $\mathcal{P}_{dp}$. Identical resources are reserved on the hot-standby (protection) path, as those needed on primary path. So if the primary path fails, the traffic is automatically switched to the protection path, thus assuring 100%

availability of the resources. The protection mechanism is analogous to $1:1$, or $1+1$ protection (the model is the same for both).

If the two paths, primary and protection, are chosen in such a way, that any failure which affects the primary path $\mathcal{P}_{dp}$ leaves the protection path $\mathcal{R}_{dp}$ intact, the paths are called *failure-disjoined*.

Having recovery mechanism in the upper layer allows for protection against the upper layer failures, as well as provides more efficient resource utilization than recovery in the lower layer. A disadvantage, though, is that more recovery actions are needed to recover from a failure in the lower layer [24].

8.5 Throughput maximization and hot-standby path protection in IP/MPLS layer

Problem TM-HS: a two layer maximum-throughput network design problem with hot-standby path protection in IP/MPLS layer.

$$\max \qquad T = \sum_d X_d \tag{8.64}$$

$$\text{s.t.:} \qquad \text{(8.55)-(8.63).}$$

Problem TM-HS differs from the problem PF-HS just by the objective function and the solution space is defined by the same set of constraints. The objective function (8.64) provides total network throughput (T) maximization.

8.6 Cost minimization and hot-standby path protection in IP/MPLS layer

Problem CM-HS: a two layer minimum-cost network design problem with hot-standby path protection in IP/MPLS layer

$$\min \qquad C = \sum_g \xi_g u_g \tag{8.65}$$

$$\text{s.t.:} \qquad \sum_p v_{dp} = 1\,, \qquad \forall d \in \mathcal{D} \tag{8.66}$$

$$\sum_d \sum_p \theta_{edp} h_d v_{dp} \le M y_e\,, \qquad \forall e \in \mathcal{E} \tag{8.67}$$

$$\sum_k z_{ek} \geq y_e \,, \qquad \forall e \in \mathcal{E} \tag{8.68}$$

$$\sum_e \sum_k \varphi_{gek} z_{ek} \;\leq\; N u_g \,, \qquad \forall g \in \mathcal{G} \tag{8.69}$$

$$v_{dp} \in \mathbb{B} \,, \qquad \forall d \in \mathcal{D},\ \forall p \in \mathcal{P}_d \tag{8.70}$$

$$z_{ek}, y_e, u_g \in \mathbb{Z}^+ \,, \qquad \forall e \in \mathcal{E},\ \forall k \in \mathcal{K}_e,\ \forall g \in \mathcal{G}. \tag{8.71}$$

Objective function C (8.65) is related to the total network cost. Most of the constraints in CM-HS are the same as for problems PF-PD or PF-HS. What differs CM-HS from the other two problems is only constraint (8.67), which assures that all flows routed on link e, including the flows realizing hot-standby paths, do not exceed the capacity of link e which is allocated in modules of size M. One can see that constrain (8.67) is formed by combining constraints (8.46) and (8.47). This is possible due to the fact that the minimal cost C is achieved when flows x_{dp} attain the minimum admissible value, i.e., h_d. Since for each demand d, volume h_d is now a constant, the value of each flow x_{dp} is either h_d or 0, and this makes it possible to formulate problem CM-HS without using constraint (8.46) involving "big M", as it is the case in problems PF-PD and PF-HS.

8.7 Problem resolution methods

In this section we present a framework for solving two-layer network design problems. The framework is based on an efficient heuristic solution method, which is referred to as the *iterative method* (called *design through layer separation* in [120]). The method is based on the following two key components: 1) decomposition of the problem into two subproblems related to the two separate layers and performing optimization for the two layers individually and 2) allocation of flows to the shortest paths, based on the calculated costs of the paths. The procedure of exchanging the data between the layers and re-solving the problems of the two layers is iterated until satisfactory (hopefully near-optimal) solution is reached. The method is implemented as an iterative algorithm, as shown in Figure 8.1. This approach was also studied in [121] and [122] for designing cost-efficient multi-layer networks.

For the considered problems, an artificial cost of installing one module M on each of the upper layer links e is introduced. Then, for the upper layer a link capacity dimensioning problem with modular link capacities and single-path routing is solved. (Note that this problem alone is hard, but still simpler than the considered two-layer NDP. In fact, this single-layer NDP is used to test commercial MIP solvers [123]). The obtained modular link capacities of the upper layer

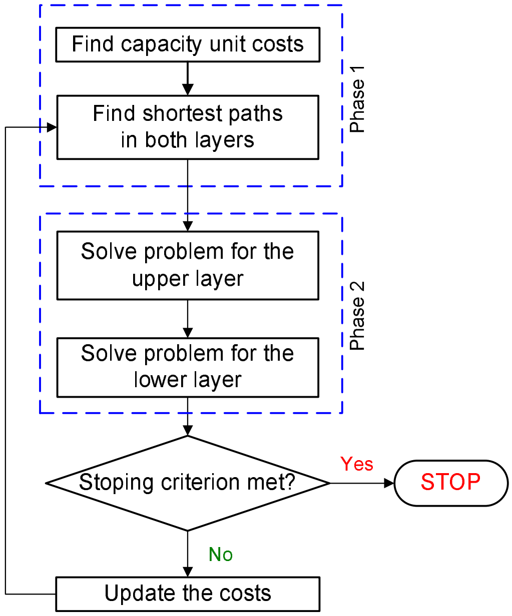


Figure 8.1: General algorithm for solving the two-layer NDPs.

are then used as demand volumes for solving the lower layer problem. When flow allocation in the lower layer is found, cost of the lower layer path, realizing the capacity of each link e, is re-calculated. Based on these re-evaluated costs, the upper layer link costs are adjusted and the problems re-solved, leading to an iterative solution process (see Figure 8.1).

Certainly, the iterative method requires solving more problems, but the problems are less complex than those solved by the direct application of a MIP solver to the original problem formulation (this will be called *direct method*). In the direct method less problems have to be solved, but they are more complex. Thus the proposed iterative approach combines the exact and heuristic solution methods.

The following sections present description of the solution method and show how the general solution algorithm is adapted to solve each of the subproblems. They also serve as a means of explaining the method itself by example.

8.8 Iterative method for solving problem PF-PD

In this section we will discuss how to resolve PF-PD problem presented in Section 8.3 using the iterative method. In the sequel the IP/MPLS layer will also be referred to as layer 2 (L2) and the DWDM layer—as layer 1 (L1).

The method described below consists in iterative execution of two phases (see Figure 8.1) until a satisfactory (hopefully near-optimal) solution is found.

In *Phase 1*, for each L2 link e, the lengths of the given number n_e (recall, that n_e is a diversity factor in PD recovery) shortest paths realizing link e in L1 are calculated. These lengths are used to find the unit capacity cost for each link e of L2. Based on the L2 link costs, the length of the shortest path in L2 realizing each demand d is calculated.

In *Phase 2* two consecutive separate optimization problems, for L2 and L1, are solved. First the problem for L2 is solved resulting in the L2 flows and link capacities. Each demand d is allocated to a single shortest path (with respect to the costs calculated in Phase 1) in L2. Capacities of the L2 links are then used as demand volumes for L1, and the problem for L1 is solved. As a result, capacities of L1 links, as well as L1 flows are found. For each L2 link e, its flows in L1 are split to n_e shortest paths, with respect to the lengths calculated in Phase 1.

After Phase 2, the updated costs of links and lengths of paths are recalculated (in Phase 1), and so the algorithm proceeds, until a satisfactory solution is found. The algorithm is described formally in the next section.

Note that in the algorithm below we assume that the logarithmic function has been linearized using a piece-wise linear approximation, and thus the objective function for problems PF-PD and PF-HS has been substituted by $R = \sum_d w_d f_d$.

8.8.1 Approximate iterative algorithm

Step 1: Calculate initial costs $\zeta_g, g \in \mathcal{G}$ and $\varsigma_e, e \in \mathcal{E}$ as described in Section 8.8.3.

Step 2: For each demand d find one path, $\mathcal{P}_d$, shortest with respect to the link costs ς_e. (Each such path is defined as $\mathcal{P}_d = \{e : \delta_{ed} = 1\}$, so its length is $|\mathcal{P}_d| = \sum_e \delta_{ed}\varsigma_e$.) The path index, p, is omitted in the variables and constants of the following formulation, because only one path, $\mathcal{P}_d$, is used for each demand d. Solve the following design problem for L2.

$$\max \quad R = \sum_d w_d f_d \tag{8.72}$$

$$\text{s.t.:} \quad h_d \leq x_d \leq H_d \,, \qquad \forall d \in \mathcal{D} \tag{8.73}$$

$$\sum_d \delta_{ed} x_d \leq M y_e \,, \qquad \forall e \in \mathcal{E} \tag{8.74}$$

$$\sum_e \kappa_e y_e \leq B \;, \tag{8.75}$$

$$f_d \leq a_i x_d + b_i \;, \qquad \forall i \in \mathcal{I} \;, \forall d \in \mathcal{D}. \tag{8.76}$$

After solving the above MIP problem (where y_e are integers) the current loads of the L2 links are given by:

$$\hat{y}_e = \sum_d \delta_{ed} x_d \;\;,\;\; e = 1, 2, \ldots, E, \tag{8.77}$$

where $\hat{y}_e$ are non-negative continuous variables. The link loads $\hat{y}_e$ will be used to update the costs of links in Step 4. Also, let R^* be the optimal value of (8.72).

Step 3: For each link e let $\mathcal{K}_e$ be the set of indices k of n_e shortest paths in L1 with respect to the link costs ζ_g. Solve the following design problem for L1, realizing the given L2 link capacities (y_e) by flows z_{ek} in L1.

$$\min \quad \sum_g \xi_g u_g \tag{8.78}$$

$$\text{s.t.:} \quad \sum_{k \in \mathcal{K}_e} z_{ek} = y_e \;, \qquad \forall e \in \mathcal{E} \tag{8.79}$$

$$z_{ek} \leq \lceil y_e / n_e \rceil \;, \qquad \forall e \in \mathcal{E} \;, \forall k \in \mathcal{K}_e \tag{8.80}$$

$$\sum_e \sum_{k \in \mathcal{K}_e} \gamma_{gek} z_{ek} \leq N u_g \;, \qquad \forall g \in \mathcal{G}. \tag{8.81}$$

After solving the above problem, the current loads of the L1 links are given by:

$$\hat{u}_g = \sum_e \sum_{k \in \mathcal{K}_e} \gamma_{gek} z_{ek} \;\;,\;\; g = 1, 2, \ldots, G, \tag{8.82}$$

where $\hat{u}_g$ are non-negative continuous variables. The link loads $\hat{u}_g$ will be used to update the costs of links in Step 4 of the algorithm.

Step 4: Store the current solution. Update the costs as described in Section 8.8.4. If the resulting cost vector $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \ldots, \zeta_G)$ is the same as the cost vector used in one of the previous iterations (or if the iteration limit is exceeded) stop the computations. Select a feasible solution with the highest value of R^* (a highest value of T^* for problem TM-HS, and a lowest value of C^* for problem CM-HS). Else goto Step 2.

8.8.2 Remarks

When solving the MIP problem in Step 2 it is assumed that budget B is sufficiently large, so that the problem is feasible, irrespectively of which set of the shortest paths is chosen to realize the demands. Note, that if the problem is relaxed (i.e., constraint (8.74) for each link e is replaced by $\sum_d \delta_{ed} x_d \leq y_e$ and y_e is made continuous), constraints (8.73) and (8.76) are removed, and the original objective function (8.42) is used, then there exists an explicit solution for the resulting problem (see Chapter 8 in [120]):

$$x_d^* = \frac{B w_d}{\sum_d w_d |\mathcal{P}_d|}. \tag{8.83}$$

8.8.3 Defining initial costs

In Step 1 of the algorithm the initial costs of links and lengths of the shortest paths in both layers are calculated as follows.

Let us define the cost of installing one capacity unit in a link of L1 as:

$$\zeta_g = \xi_g / N \quad , \quad g = 1, 2, ..., G. \tag{8.84}$$

For each link e find a set of n_e shortest paths in L1 with respect to the link costs ζ_g. Let $\mathcal{K}_e$ be a set of indices ks of these paths. Then the length of (a cost of installing one capacity unit on) path k is defined as:

$$\mu_{ek} = \sum_g \gamma_{gek} \zeta_g \quad , \quad e = 1, 2, ..., E \quad , \quad k \in \mathcal{K}_e, \tag{8.85}$$

and the *average* cost of installing one capacity unit on a shortest path in L1 for realizing capacity of L2 link e is defined as:

$$\kappa_e = (\sum_{k \in \mathcal{K}_e} \mu_{eq}) / n_e \quad , \quad e = 1, 2, ..., E. \tag{8.86}$$

Then the cost of installing one capacity unit on L2 link e is:

$$\varsigma_e = \kappa_e / M \quad , \quad e = 1, 2, ..., E. \tag{8.87}$$

The presented algorithm is based on allocation of flows in the network on the shortest paths, and therefore requires finding one shortest path (one shortest pair of disjoint paths for problems PF-HS, TM-HS and CM-HS) in L2 for realizing each demand d , as well as n_e shortest paths ($n_e = 1$ for problems PF-HS, TM-HS and CM-HS) in L1 for flows realizing capacities of links e. There are two approaches of finding the shortest paths needed for the calculations. First, one can search for the shortest paths only in the predefined lists of candidate paths. Second, a global search for the paths (using a shortest-path algorithm) can be performed in the graph of the corresponding layer. In both cases, the ζ_g and ς_e costs are used for the calculations.

8.8.4 Updating the costs

After each iteration of the algorithm, the costs have to be updated. The goal is to have as little modules installed on the links and as high links utilization as possible, since this leads to the smallest overall network cost. One way to achieve this is to take into account the utilization of the installed capacity modules on the links, when calculating capacity unit costs. If the installed modules on a link are under-utilized the capacity unit cost on this link will be increased. This objective can be realized by the adjusted cost calculations which are performed as follows. The cost of installing one capacity unit on L1 link g is calculated as:

$$\zeta_g = \xi_g u_g / \hat{u}_g \ , \quad g = 1, 2, ..., G, \tag{8.88}$$

and the cost of installing one capacity unit on L2 link e is:

$$\varsigma_e = \kappa_e y_e / \hat{y}_e \ , \quad e = 1, 2, ..., E, \tag{8.89}$$

where κ_e is calculated as in (8.86). The remaining required entities are calculated as in Section 8.8.3, using costs ζ_g and ς_e calculated from (8.88) and (8.89), respectively.

8.9 Iterative method for solving problem PF-HS

In this section we will discuss how to resolve Problem PF-HS presented in Section 8.4 using the iterative method.

Analogously to problem PF-PD, the resolution method for problem PF-HS consists in iterative executing of its two phases (see Figure 8.1) until a satisfactory solution is found.

In *Phase 1*, for each L2 link e the length (cost) of its shortest path in L1 is calculated. These costs are used to find capacity unit costs of the L2 links. Based on these L2 link costs, a shortest pair of failure-disjoint paths in L2 realizing each demand d is found and its length is calculated.

In *Phase 2* two consecutive separate optimization problems, for L2 and L1, are solved. First, the problem for L2 is solved resulting in the L2 flows and link capacities. Each demand d is allocated to one of the shortest (primary, backup) path pairs (with respect to the lengths calculated in Phase 1) in L2. Resulting capacities of the L2 links are then used as demand volumes for L1, and the problem for L1 is solved. As a result, capacities of the L1 links, as well as the L1 flows are found. There are no restrictions imposed on how flows in L1 shall be allocated. Thus, flow bifurcation in L1 is allowed in solving the modular dimensioning problem for L1.

After Phase 2 the updated costs of links and lengths of paths are recalculated (in Phase 1), and so the algorithm proceeds, until a satisfactory solution is found. Initial costs and updated costs for this problem are calculated in the same way as for problem PF-PD (see Sections 8.8.3—8.8.4), except that now a number of diverse paths in the lower layer n_e is set to 1. The algorithm is described formally in the following section.

8.9.1 Approximate iterative algorithm

The iterative algorithm for problem PF-HS is very similar to that of Problem PF-PD, presented in Section 8.8.1. Below only the differences with respect to the previous algorithm are presented. Steps 1 and 4 are exactly the same for both algorithms.

Step 2: For each demand d find one of the shortest pairs of failure-disjoint paths, $(\mathcal{P}_{dp}, \mathcal{R}_{dp})$, with respect to the link costs ς_e. (Each such path pair is defined as $(\mathcal{P}_{dp}, \mathcal{R}_{dp}) = \{e : \theta_{edp} = 1\}$, so its length is $|(\mathcal{P}_{dp}, \mathcal{R}_{dp})| = \sum_e \theta_{edp}\varsigma_e$.) The path pair index p is omitted in the variables and constants in the formulation, because only one (shortest) predefined pair of paths, $(\mathcal{P}_d, \mathcal{R}_d)$, is used for each demand d. Solve the following design problem for L2:

obj (8.72)

s.t.: (8.73), (8.75)-(8.76) and

$$\sum_d \theta_{ed} x_d \le M y_e\ , \qquad \forall e \in \mathcal{E}, \tag{8.90}$$

where θ_{ed} is defined only for the selected shortest pair of failure-disjoint paths $(\mathcal{P}_d, \mathcal{R}_d)$. Let R^* be the optimal value of (8.72). Calculate L2 link loads from:

$$\hat{y}_e = \sum_d \theta_{ed} x_d \;\;, \;\; e = 1, 2, \ldots, E. \tag{8.91}$$

Step 3: Solve the following design problem for L1, realizing the given L2 link capacities (y_e) by flows z_{ek} in L1.

obj	(8.78)	
s.t.:	(8.79) and (8.81).	(8.92)

Calculate the current loads of the L1 links from (8.82).

8.10 Iterative method for solving problem TM-HS

In this section we will discuss how to resolve Problem TM-HS presented in Section 8.5 using the iterative method. The method and algorithm for solving the problem TM-HS are almost identical to those for the problem PF-HS. The only difference is that in the optimization problem solved in Step 2 of the algorithm (see Section 8.9.1) the constraint (8.76) is dropped and the objective function (8.72) is replaced by the following:

$$\text{maximize} \quad T = \sum_d x_d \tag{8.93}$$

Thus the problem to be solved is defined by (8.93), (8.73), (8.75) and (8.90). Also, let T^* be the optimal value of (8.93).

8.11 Iterative method for solving problem CM-HS

In this section we will discuss how to resolve Problem CM-HS presented in Section 8.6 using the iterative method. Like in the case of TM-HS, the method and algorithm for solving the problem CM-HS are similar to those for the problem PF-HS. Only the optimization problem solved in Step 2 of the algorithm (see Section 8.9.1) has to be replaced by the following:

$$\min \qquad C = \sum_e \varsigma_e y_e \tag{8.94}$$

$$\text{s.t.:} \qquad \sum_d \theta_{ed} h_d \;\leq\; M y_e \;, \qquad \forall e \in \mathcal{E}, \tag{8.95}$$

where θ_{ed} is defined only for the selected shortest pair of failure-disjoint paths $(\mathcal{P}_d, \mathcal{R}_d)$. Also, let C^* be the optimal value of (8.94).

8.12 Numerical examples

The proposed iterative algorithms for the considered four problems were tested on a set of randomly generated network topologies. Network sizes were ranging from 12 to 60 nodes in the lower (DWDM) layer. The upper (IP/MPLS) layer usually contained slightly less nodes. The selected test networks are summarized in Table 8.1. The aim of the numerical experiments was to examine the time efficiency and solution quality (depending on the problem in question: it was maximum of revenue R^* or throughput T^*, or minimum value of cost C^*) of the iterative algorithms, as compared to the direct algorithms (applications of the CPLEX MIP solver to the problem formulations in their original form). The original MIP problems were also solved directly with relaxed MIP optimality by 10% (entry RO in the tables), and also the (partial) linear relaxations of the MIP problems were solved, where the variables z_{ek}, y_e and u_g were assumed to be continuous (entry LR in the tables). The results in the LR rows are the upper bounds for problems PF-PD, PF-HS and TM-HS, and lower bounds for problem CM-HS. For the experiments the values $M = 64$, and $N = 128$ were used.

All the programs were implemented in the mathematical modeling language AMPL ver. 20021031 [119], and CPLEX 9 was used as a solver. PF-PD and PF-HS problems were solved on a PC with Pentium-IV 3 GHz CPU and 2 GB of RAM, running the Windows XP Pro operating system. TM-HS and CM-HS problems were solved on a PC with Pentium-III 866 MHz CPU and 256 MB RAM, running the Fedora Core 3 operating system. Table 8.2 shows the computational results for both (iterative and direct) resolution methods as applied to problem PF-PD. Table 8.3 shows corresponding results for problem PF-HS. Tables 8.4 and 8.5 shows the results for the problems TM-HS and CM-HS correspondingly. The solution time indicated in the tables is a wall-clock time.

As it can be seen from Tables 8.2-8.5, the smaller is the network (simpler NDP), the closer are the solution times of the iterative and direct algorithms. And vice versa, the larger is the network (more complicated NDP), the bigger is the difference between the solution times of the iterative and direct algorithms. For the considered network examples the iterative algorithms (for all the problems) had much better time efficiency, except for two cases: problems PF-PD and CM-HS for network N_{12} were solved slightly faster using the direct approach. Although the direct algorithms produced marginally better solutions, the gap between the heuristic and the exact solutions was very small. For problems PF-PD and PF-HS in all the cases the gap between the optimal revenues R^* obtained by the direct and iterative algorithms was below $0,01\%$. For problem TM-HS the optimal throughput T^* for both approaches was the same in all the cases. More variation in results and bigger gaps were noticed for problem CM-HS. There the biggest gap of $8,66\%$ between the solutions was encountered for problem N_{41}.

For network N_{60} and problems TM-HS and CM-HS, computations using direct approach, relaxed optimality, or linear relaxation were aborted after 3 hours if they did not reach the solution. Therefore almost no results are available for these cases in the Tables 8.4 and 8.5. Still, this long computation time of 3 hours serves for time efficiency comparisons of the two solution approaches.

Comparing the solution for the iterative algorithm and the RO case, one can see that for problems PF-PD, PF-HS and TM-HS the iterative algorithm produced the same or better result (in terms of revenue or throughput) and in all but one cases (PF-PD for N_{12})—also in a shorter time. This also confirms that the solution of the iterative algorithm in the considered examples is within 10% from optimality. For problem CM-HS, the iterative algorithm produced the same or better results (in terms of cost) than those for RO for networks N_{30} and N_{60}, while slightly worse results for networks N_{12} and N_{41}. However, the solution times of the iterative algorithm for all the cases, except one (for N_{12}), were even for this problem shorter than those for RO.

Comparing the results of the iterative algorithm and for the LR case, the iterative algorithm in most of the cases produced the results faster than LR. The results for LR represent a lower bound for the cost in the minimization problems, and an upper bound for revenue/throughput in the maximization problems. In all the cases, the results achieved by the iterative algorithm are within 10% of the respective bound, as can be seen from the corresponding LR entries in the tables.

Finally, we note that in some cases (e.g., PF-PD for N_{60}, PF-HS for N_{41} and CM-HS for N_{30}), the results (cost, revenue, or throughput) of the iterative algorithm are marginally better than those for the direct algorithm due to numerical inaccuracies.

8.12.1 Convergence Issues

For an example, we now examine convergence issues of the iterative algorithm for problem PF-PD. Figures 8.2 and 8.3 illustrate how the iterative algorithm for problem PF-PD converges when solving for N_{41}. Figure 8.2 shows evolution of the L2 objective function (8.72) value (referred to as the L2 objective), as well as an optimal value of R^* obtained by the direct solution method (referred to as the optimal value). Figure 8.3, in turn, depicts the evolution of the L1 objective function (8.78) value, as well as the value of budget B.

In the first step of the algorithm (see Section 8.8.1), the initial costs of links and lengths of the shortest paths are calculated. Then the L2 problem is solved in Step 2 producing the solution value depicted in Figure 8.2 for iteration counter equal to 1. The resulting modular link capacities of L2 are then used as demand volumes for the L1 problem solved in Step 3. The resulting value of the L1 objective function (8.78) appears to be above the available budget B, as can be seen in Figure 8.3 for iteration 1. Although both problems, for L2 and L1, are formally feasible, the overall solution is infeasible, because the cost (8.78) exceeds the assumed budget B. It is due to the fact that the costs κ_e are in general inconsistent with the costs ξ_g implying that the cost of the L2 links, $\sum_e \kappa_e y_e$, is not equal to cost (8.78). This is the reason why the cost (8.78) can exceed the assumed budget B, making the overall solution infeasible. Hence, although the objective function value of L2 is the highest among all the iterations, it is not the final solution because it is achieved on expense of exceeding the budget constraint for the L1 link capacities. In Step 4 of the algorithm the costs are updated to take

Network	layer	# nodes	# links	# paths per demand	# demands
N_{12}	L_2	12	22	6-14	66
	L_1	12	18	2-3	–
N_{30}	L_2	25	56	4-5	300
	L_1	30	68	4-5	–
N_{41}	L_2	21	37	6	209
	L_1	41	72	3	–
N_{60}	L_2	50	125	3	1225
	L_1	60	142	3	–

Table 8.1: Networks used for experiments.

Network		N_{12}	N_{30}	N_{41}	N_{60}
Time	Iterative	0.51	0.89	1.35	2.91
	Direct	0.45	33.92	104.71	220.39
	RO	0.44	30.75	22.53	220.61
	LR	0.38	4.23	2.1	22.49
R^*	Iterative	1188194	5400880	3129648	22053589
	Direct	1188194	5400880	3129883	22053576
	RO	1188194	5356737	3121111	22053576
	LR	1188194	5400880	3130178	22053591

Table 8.2: Computational results for problem PF-PD.

Network		N_{12}	N_{30}	N_{41}	N_{60}
Time	Iterative	0.84	7.94	2.02	42.93
	Direct	4.48	20.13	30.80	2458.75
	RO	4.46	20.06	30.79	2016.08
	LR	1.79	5.28	4.10	26.21
R^*	Iterative	1188194	4989079	3780614	21815445
	Direct	1188194	4989249	3780299	21816236
	RO	1188194	4989249	3780299	21708169
	LR	1188194	4989428	3780616	21817057

Table 8.3: Computational results for problem PF-HS.

Network		N_{12}	N_{30}	N_{41}	N_{60}
Time	Iterative	2.66	6.70	3.97	$85,09$
	Direct	14	1414.50	8042.60	> 10800
	RO	14	1015.54	7379.85	> 10800
	LR	7.37	18.67	149.13	> 10800
T^*	Iterative	$6.6e+07$	$3e+08$	$2.1e+08$	$1.2e+09$
	Direct	$6.6e+07$	$3e+08$	$2.1e+08$	$-$
	RO	$6.6e+07$	$2.99e+08$	$2.09e+08$	$-$
	LR	$6.6e+07$	$3e+08$	$2.1e+08$	$-$

Table 8.4: Computational results for problem TM-HS.

into account the installed link capacities and the actual link loads, and iteration 2 of the algorithm starts. Problems for L2 and L1 are solved again in Step 2 and Step 3, respectively, and the costs are updated again. In this way the algorithm proceeds until in iteration 4 the resulting L2 link capacities can be implemented

Network		N_{12}	N_{30}	N_{41}	N_{60}
Time	Iterative	$2,9$	$6,5$	$6,2$	53
	Direct	$2,3$	91	$9,3$	> 10800
	RO	1.4	55.9	7.1	> 10800
	LR	1.4	12.7	7.1	68.17
C^*	Iterative	58363	1040293	1743379	1865448
	Direct	54784	1041923	1604460	–
	RO	54784	1055191	1604460	–
	LR	54767	1041860	1604268	1865379

Table 8.5: Computational results for problem CM-HS.

with a cost slightly lower than the assumed budget B (see values for iteration 4 in Figure 8.3). The costs are recalculated and it appears that the resulting cost vector $\boldsymbol{\zeta}$ is the same as in the previous iteration (the cost vector which resulted in the solution of iteration 4). So the algorithm would normally terminate here. Although, for illustrative purposes, one more iteration was performed. Thus the problems for the two layers are solved once more in iteration 5, producing the same solution as in the previous iteration, since the cost vector $\boldsymbol{\zeta}$ is the same as in iteration 4. Recalculating costs again results in the same cost vector $\boldsymbol{\zeta}$ as in the previous iteration so the algorithm terminates.

The algorithms for the other problems converge to the final solution in a similar way.

The presented numerical results demonstrate that the proposed iterative algorithms converge quickly to near-optimal solutions. So far this is just an experimental observation and we cannot claim it is a general property. To the best of our knowledge no convergence results are known for this type of iterative procedures applied to similar problems (in [120], [121], [122] such results are not reported). Hence, since there is no theoretical evidence on the convergence, the presented approach should be regarded as a promising heuristic.

One observation that supports the idea of the proposed layer-separation approach (based on adjusting layer-specific link marginal cost coefficients) is that in most cases it works fine for linear relaxations of two-layer design problems (and also for more layers), i.e., for analogous problems as the four problems considered in this chapter, but with continuous link capacities and bifurcated continuous path flows. In Section 12.1.4 of [120] it is demonstrated how to find an optimal solution for a simple linear two-layer design problem in just one iteration. This is done in two steps. First, we use marginal costs $\xi_g, g = 1, 2, \ldots, G$ to define

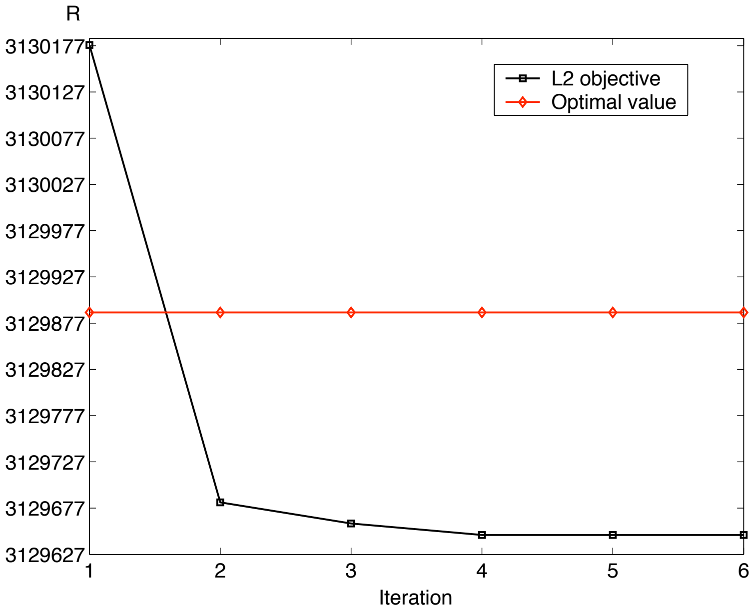


Figure 8.2: Evolution of L2 objective value for N_{41} in problem PF-PD.

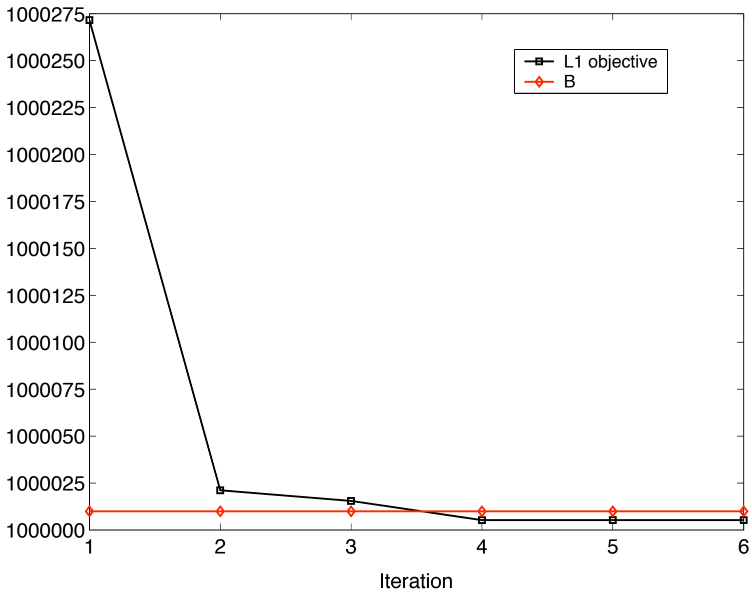


Figure 8.3: Evolution of L1 objective value for N_{41} in problem PF-PD.

marginal cost coefficients $\varsigma_e, e = 1, 2, \dots, E$ for links of the upper layer (as the lengths of the shortest paths in the lower layer computed with respect to ξ_g), and use them to compute the shortest paths in the upper layer. Second, we allocate demands to the shortest paths in the upper layer, compute the resulting link loads y_e, and finally realize the loads y_e on the shortest paths in the lower layer. This procedure can be almost directly applied to the linear relaxation of problem TM-HS, and, after some modifications, to linear relaxations of the remaining problems.

The above procedure does not require iterations and is called shortest path allocation. In fact, as discussed in Section 12.4.1 of [120], the shortest path allocation rule does not apply in the general case of a linear design problem, and in general the optimal dual variables corresponding to constraints (8.61) and (8.59) should be used for ζ_g and ς_e, respectively. The values of the optimal dual variables can be found through sub-gradient maximization of the decomposable dual problem. In fact, even though we are able to find the optimal dual variables, they may not reveal the exact primal optimal solution (when the shortest paths are not unique). Still, the shortest path allocation rule applied to optimal dual variables gives in many cases very good heuristic solutions. In our iterative approach we are in fact following such an approach for non-linear (MIP) problems, hoping that the costs ζ_g and ς_e play a similar role as the optimal dual variables in the linear case.

8.13 Conclusions

In this chapter we considered four design problems for (two-layer) IP/MPLS over DWDM networks with meshed topology and developed a general framework for their heuristic resolution. The presented design problems differ in objective functions and assumed recovery mechanism. The problems for proportionally fair bandwidth sharing, total network throughput maximization, and total network cost minimization were studied. The considered recovery mechanisms were: 1) protection only in the DWDM layer, where a simple protection mechanism—path diversity is used; 2) protection only in the IP/MPLS layer by hot-standby path protection.

All the design problems are difficult to resolve. Therefore we propose a generic heuristic iterative design method and derive the resulting algorithms for solving the particular problems. The method is based on the idea of decomposing the multi-layer network problem into subproblems for each of the layers, and resolving in an iterative manner the relevant optimization subproblems for each layer separately. The efficiency and solution quality of the algorithms is illustrated with numerical examples. The examples reveal that the heuristic algorithms provide very good near-optimum solutions and exhibit execution times that are drastically shorter than execution times of direct exact algorithms, especially for large networks.

Good performance (convergence to near-optimum solutions in short execution time) of the heuristic iterative algorithms for solving the four different multilayer network design problems suggests that the proposed generic iterative method can be applied for efficient resolution of other two-layer network design problems, as well as extended to the problems with more layers. Still, so far we are not able to give general comments on the convergence of the proposed iterative approach—this issue requires further research along the lines of the approximation theory.

CHAPTER 9

Integrated recovery, routing, and load balancing strategy

9.1 Introduction

A number of factors influence network performance and efficiency of its resource utilization. Such main factors are recovery mechanism, routing and load balancing strategy. By using a capacity-efficient recovery mechanism some of the capacity, previously dedicated for protection, can be saved and used for transporting the revenue generating traffic. On the other hand, an efficient routing strategy can ensure a good trade-off between a well balanced load and the consumed capacity. Load balancing, among other things, decreases probability of rejection for future requests due to lack of resources in some part of the network. A capacity-efficient recovery mechanism integrated with a routing strategy and load balancing on network links is developed in this chapter.

We focus on IP/MPLS network. Traffic demands are assumed to have priority levels. High priority traffic can preempt low priority traffic. We assume that the high priority (HP) label switched paths are long-lived, e.g., they are used for virtual private networks (VPNs) or as virtual leased lines and thus are not reconfigured often, and also that they constitute a significant amount of network traffic. Bandwidth reserved for backup LSPs of HP demands can be used by low priority LSPs in the normal state. Since the high priority LSPs require 100% survivability, it is important how the network resources are used for their protection because this implies how many low priority LSPs are discarded from the network when backup paths for HP demands are activated. The demands in the network are divided into two categories–high priority demands (with priority= 0) and low priority demands with priorities ranging from 1 to 4; thus, there are 5 priority classes in total. An LSP of demand with a higher priority level (the lower priority number) can preempt LSPs of demands with the lower priority levels. We assume that only HP demands require 100% survivability and other demands can be recovered only if resources are available, in the order of priority. Based on this differentiation of the demands we propose a network protection and routing strategy, which is as follows. Failure-dependent backup path protection mechanism with sharing of backup capacity is used for assuring availability of the HP demands. The selection of primary and failure-dependent backup paths for HP connections is performed by an off-line network design problem defined as a mixed integer program. It is valid to assume that the set of HP demands is known at this design stage, since such a global optimization can be performed not only during the greenfield network design stage (using estimated demand volumes), but also periodically (however, with a long period, e.g., half a year) during the network operation. For routing the HP demands we propose a strategy combining shortest path routing and load balancing on the links. We propose a load balancing function using proportional fairness rule applied to residual bandwidths. The low priority connections are routed using on-line constrained shortest path first (CSPF) algorithm.

The proposed protection and routing strategy has been tested both by numerical experiments and simulations. The results show that for the considered network examples less LSPs are disrupted on the average due to failures using the proposed protection mechanism and routing strategy, as compared with the other recovery strategies. Thus, with the given link capacities the network is able to retain considerably more LSPs in the case of failures.

The rest of the chapter is organized as follows. First, the related work is reviewed and an IP/MPLS network model is presented. Then the mathematical problem formulation for finding nominal and backup paths for HP demands is presented. Next, the proposed routing and load balancing criterion is discussed, leading to objective function for the presented network design problem. Finally, some numerical experiments are presented, followed by final conclusions.

9.2 Related work

There are a number of novel recovery mechanisms and routing strategies proposed in the recent literature, conserving network resources used for assuring network resilience. Most of such recovery mechanisms assume a kind of backup capacity multiplexing. Several of the proposed protection mechanisms are reviewed below.

In [124] a demand-wise shared protection (DSP) is presented. It combines advantages of dedicated and shared path protection. In DSP, a capacity is reserved for each demand and is not shared with other demands, but only among the paths of the same demand. The paths of a demand are not explicitly divided into nominal (working) and backup, and in general the whole demand volume is assumed to be split among several working paths.

In [125] and [126] two schemes, called original quasi-path restoration (OQPR) and improved quasi path restoration (IQPR), are presented. In QPR schemes, a failing path is divided into three parts: a subpath from source node to a critical node on the same side of the failing link, the failing link, and the subpath from the critical node on the other side of the failing link to the destination node. Then restoration implies rerouting only one of the subpaths, while the other subpath is left intact. Traffic affected by the failure is rerouted on single/multiple paths using spare capacity in the network.

In [127] a problem of lightpath routing and protection in wavelength division multiplexing networks with dynamic traffic demands is considered. The authors propose a new multiplexing technique called primary-backup multiplexing and shows that it increases network resource utilization and improves the network blocking performance. The technique allows to share one wavelength channel among one primary and one or more backup lightpaths.

In [128] an $L + 1$ fault tolerance routing strategy is studied. The main idea is to transform the original network with L links into a collection of $L + 1$ sub-networks, where one of the links in the original (base) network is removed, this way modeling each single-link failure scenario. A connection is accepted to the network only if it can be routed in all of the subgraphs. When a failure occurs the connectivity is restored by putting the network to the state of the particular (failure-dependent) subgraph. The major drawback is that this strategy can potentially require a full network reconfiguration upon a single link failure (a.k.a., unrestricted reconfiguration).

In [129] a dynamic failure-dependent path protection scheme is developed. The study assumes a WDM network with heterogeneous grooming architectures. A method is developed to assign primary and (failure-dependent) backup paths to requests, in a way which makes it possible to survive a single shared risk link group (SRLG) failure at any given time. The routing strategy used is Available Shortest Path (ASP), where Dijkstra's algorithm is used to find shortest paths, first checking the resource availability on them. In the study a restricted reconfiguration is assumed, i.e., only a failing primary connection is reassigned to its backup path, while unaffected connections remain in place.

In [130] algorithms for dynamic routing of bandwidth-guaranteed connections are studied. Shared protection and possibility to protect each working path by one or multiple protection paths are assumed. The proposed algorithms are presented in the context of the protection rearrangement framework, allowing to adaptively reroute allocated protection paths before failure, when traffic conditions change. This results in a more efficient network resource utilization.

9.2.1 This work

The proposed protection and routing strategy is related to the studies reviewed above in several ways. First of all, the chosen failure-dependent backup path protection mechanism with restricted reconfiguration corresponds to that presented in [129], except that in our case the proposed method is for off-line computation and for IP/MPLS network, while the one in [129] is on-line and is dedicated to WDM network. Also, the routing strategy we propose is different from ASP in [129] and takes load balancing on links into account. The off-line network design with failure-dependent routing also resembles the $L+1$ approach in [128], except that in our case routing in all the failure states is designed simultaneously, and the reconfiguration is restricted.

Secondly, the primary and backup paths used in our protection and routing strategy are *failure-disjoint*. This means that the primary and backup paths for a given demand do not share the failing element in the given failure scenario. The non-failing resources on a primary path may be reused by the backup paths. Also, the unused backup resources can be used by other demands. In this way, our work relates to [124], except that in our case sharing of resources is also allowed between demands. Also, there is a similarity to QPR schemes [125; 126; 131]. However, differently from QPR, the proposed strategy does not restrict how the non-failing resources of the primary path are reused by its backup path. There are also similarities with the primary-backup multiplexing method in [127].

Also, the method we propose differs from all the reviewed studies in that we assume traffic requests with different priorities.

9.3 Protection of HP demands with single backup paths

The following section presents an NDP implying demand protection by single backup paths (SBPs). For the problem the candidate path list contains a number of completely disjoint paths for each demand. Informally the problem can be stated as follows: given a set of user demands, link capacities and a candidate path lists, find nominal and single backup paths for each demand, so that some objective function is optimized. The problem assumes single path routing, thus flows are not bifurcated.

Problem SB-LSP: an offline design problem for IP/MPLS network with single-path routing and single backup path (LSP) protection. A set of feasible solutions for the problem is defined by the following constraints:
constraints

$$\sum_{p} v_{dpl} = 1\,, \qquad \forall d \in \mathcal{D},\ \forall l \in \{0,1\} \tag{9.1}$$

$$\sum_{l} v_{dpl} \leq 1\,, \qquad \forall d \in \mathcal{D}\,, \forall p \in \mathcal{P}_d \tag{9.2}$$

$$x_{dpl} = v_{dpl} h_d\,, \qquad \forall d \in \mathcal{D}\,, \forall p \in \mathcal{P}_d\,, \forall l \in \{0,1\} \tag{9.3}$$

$$\sum_{d} \sum_{p} \delta_{edp} x_{dpk} = y_{el}\,, \qquad \forall e \in \mathcal{E}\,, \forall l \in \{0,1\} \tag{9.4}$$

$$\sum_{l} y_{el} \leq c_e\,, \qquad \forall e \in \mathcal{E}. \tag{9.5}$$

In the formulation above $l \in \{0,1\}$, where $l = 0$ indicates a nominal path and $l = 1$ indicates a backup path. A binary variables v_{dpk} is equal to 1 if path p of demand d is used as a nominal path (for $l = 0$)/protection path (for $l = 1$), and 0 otherwise. Variables x_{dpl} are auxiliary and can be eliminated from the formulation by substituting the right-hand side of (9.3) into (9.4). Each variable x_{dpl} denotes a bandwidth assigned to flow carried by path p of demand d. If $x_{dpl} > 0$ for $l = 0$ then the bandwidth assigned to path p is used to carry the flow of demand d in the nominal state, where if $x_{dpl} > 0$ for $l = 1$ then the bandwidth is used in failure states, i.e., when the nominal path for d fails and the protection path is activated. Variables y_{el} denote load of each link e by nominal paths (for $l = 0$) and protection paths (for $l = 1$).

Constraint (9.1) forces single path routing, i.e., assures that only a single path for each demand d is used as the nominal path, and also that a single path is used as the protection path. Constraint (9.2) prohibits using the same path p as the nominal and as the protection path for demand d. Constraint (9.3) forces assigning the whole demand volume h_d to the chosen single nominal and protection paths for each demand. The load on each link e is constituted by two parts– flow on nominal and protection paths. For each of the two parts the load y_{el} is calculated as a sum of all flows traversing that link on the nominal, respectively– protection, paths of all demands, as expressed by the left-hand side of (9.4). Then the total load of each link e is the sum of the two parts, as given by the left-hand side of constraint (9.5), which is forced not to exceed the link capacity c_e.

There are alternative formulations for modeling the considered scenario (see, e.g., [31]). The advantage of the formulation presented above, however, is that it allows to apply different routing criteria to nominal and backup paths. This will be illustrated later in this chapter, when discussing the objective function for the problem. Observe that solving problem SB-LSP without an objective function results in paths assigned to demands that may not necessary be proper in some desirable sense.

9.4 Protection of HP demands with failure-dependent backup paths

The idea of failure-dependent backup paths is as follows. For a given list of possible failure scenarios affecting the nominal path, backup paths depending on particular failures are pre-planned. When, during the network operation, a particular failure occurs, specific backup paths matching the failure are activated and used to carry the flows until the nominal paths are repaired. The nominal and backup paths are failure-disjoint. And non-failing resources of a given nominal path can be reused by a backup path of a demand.

Example 9.1. *Consider the network depicted in Figure 9.1. Assume that there is an LSP established between nodes* s *and* d*, which uses the path* $\mathcal{P} = \{1, 4\}$*. Then two failure-dependent backup LSPs can be defined:* $\mathcal{R}_1 = \{2, 3, 4\}$ *(to be used when link* $e = 1$ *fails) and* $\mathcal{R}_4 = \{1, 5, 6\}$ *(to be used when link* $e = 4$ *fails). This can be compared to a case when a single backup path* $\mathcal{R}_{1,4} = \{2, 3, 5, 6\}$ *is used in both failure situations.*

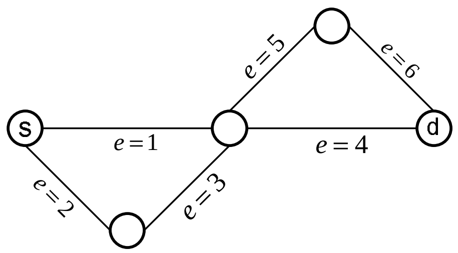


Figure 9.1: Illustration of failure-dependent LSPs.

In general, failure-dependent backup paths (FDBPs) allow for a more effective way of using the network resources than using single (failure-independent) backup paths (SBPs) in all failure situations. This is because a space of feasible solutions for NDP with SBP protection is in general more constrained than the solution space for NDP with FDBP protection, and an optimal solution to the

SBP problem is also a feasible solution to the FDBP problem. The efficiency of failure-dependent recovery mechanisms was illustrated in e.g., [128; 129].

The following mathematical problem is used to select primary and backup paths for HP demands for all predefined failure situations. Paths for each demand are chosen from predefined candidate path lists. A feasible solution to this problem guarantees 100% survivability for the HP demands.

9.4.1 Mathematical problem formulation

The following mathematical problem is used to select primary and backup paths for HP demands for all predefined failure situations. The paths for each demand are chosen from predefined candidate path lists. A feasible solution to this problem guarantees 100% survivability for the HP demands.
Problem FD-LSP: an offline design problem for IP/MPLS network with single-path routing, failure-dependent backup paths (LSPs) and restricted reconfiguration of flows (unaffected flows are not rerouted). A set of feasible solutions for the problem is defined by the following constraints:
constraints

$$\sum_p v_{dps} \;=\; 1\,, \qquad \forall d \in \mathcal{D}\,, \forall s \in \mathcal{S} \tag{9.6}$$

$$x_{dps} = v_{dps} h_d\,, \qquad \forall d \in \mathcal{D}\,, \forall p \in \mathcal{P}_d\,, \forall s \in \mathcal{S} \tag{9.7}$$

$$x_{dps} \;\geq\; \theta_{dps} x_{dp0}\,, \qquad \forall d \in \mathcal{D}\,, \forall p \in \mathcal{P}_d\,, \forall s \in \mathcal{S} \tag{9.8}$$

$$\sum_d \sum_p \delta_{edp} x_{dps} \;\leq\; y_{es}\,, \qquad \forall e \in \mathcal{E}\,, \forall s \in \mathcal{S} \tag{9.9}$$

$$y_{es} \leq \alpha_{es} c_e\,, \qquad \forall e \in \mathcal{E}\,, \forall s \in \mathcal{S} \tag{9.10}$$

$$x_{dps} \in \mathbb{R}^+,\; v_{dps} \in \mathbb{B}\,, \qquad \forall d \in \mathcal{D}\,, \forall p \in \mathcal{P}_d\,, \forall s \in \mathcal{S} \tag{9.11}$$

$$y_{es} \in \mathbb{R}^+\,, \qquad \forall e \in \mathcal{E}\,, \forall s \in \mathcal{S}. \tag{9.12}$$

Recall that $\theta_{dps} \in \{0, 1\}$ is a binary availability coefficient equal to 1 if path p of demand d is available in situation s, and 0 otherwise. It is calculated as follows: $\theta_{dps} = \min\{\alpha_{es} : \delta_{edp} = 1\}$, where $\alpha_{es} \in \{0, 1\}$ are link availability coefficients. Constraint (9.6) assures that only one single path p for each demand d is used in each situation s, while constraint (9.7) assures that the whole demand volume h_d is assigned to the chosen path p. For each demand d constraint (9.8) forces using the normal path p (used in normal situation $s = 0$) in all situations in which this path is not affected by a failure. Due to this, flows unaffected by failures are not rerouted. Path availability coefficients θ_{dps} are used in (9.8) to indicate if a given path p is affected by a failure of a link e, and are calculated from link availability coefficients α_{es} as shown above. Constraints (9.9) and (9.10) are essentially one constraint; it is split into two constraints for the clarity of presentation when referring to link loads given by the left hand side of constraint (9.9). Thus, the load of each link e in situation s is given by a sum of all flows traversing that link in the given situation. Then constraint (9.10) assures that the link loads do not exceed available link capacities in each situation expressed as $\alpha_{es}c_e$. Note that in the problem formulation above variables x_{dps} are auxiliary and are included in the model only for clarity of presentation.

FD-LSP is a $\mathcal{NP}$-hard (mixed integer programming problem–MIP), since a simpler problem of single path allocation for only normal state is already $\mathcal{NP}$-hard [31]. General purpose MIP solvers such as CPLEX or XPRESS-MP can be used for resolving the problem. Observe that solving problem FD-LSP without an objective function results in paths assigned to demands that may not necessary be proper in some desirable sense.

9.5 The classical routing strategy for HP demands

By choosing an appropriate objective function for problems SB-LSP and FD-LSP, different routing strategies can be implied. The "classical" routing strategy is shortest path (minimum hop) routing of both nominal and backup path. This strategy is imposed by using the following objective function with problem SB-LSP:

$$\min \quad \sum_l \sum_e \sum_d \sum_p \xi_e \delta_{edp} v_{dpl}. \tag{9.13}$$

Similarly, the following objective function, when used with problem FD-LSP, results in shortest nominal and failure-dependent backup paths for each demand:

$$\min \quad \sum_s \sum_e \sum_d \sum_p \xi_e \delta_{edp} v_{dps}. \tag{9.14}$$

Objective functions (9.13) and (9.14) imply minimization of the lengths of the paths for all demands (for problem FD-LSP–also in all failure situations), when link costs $\xi_e \equiv 1$ (and thus can be interpreted as a hop-count). If the costs are different than 1, the objective functions imply using the cheapest paths instead.

Furtheron problem SB-LSP with the objective (9.13) will be referred to as SB-LSP-MINHOP, and problem FD-LSP with the objective (9.14) will be referred to as FD-LSP-MINHOP.

9.6 The novel routing strategy for HP demands

The strategy proposed in this section combines load balancing on the links with shortest paths, as opposed to the widely used shortest path (minimum hop) routing alone. In this discussion by shortest path (SP) routing we mean that an arbitrary chosen shortest path (if there are few) is used for demand flows.

It is common for both the proposed strategy and the "classical" minimum hop routing strategy that shortest path routing is used for backup LSPs of HP demands. This decision stems from the natural reasoning that an HP demand with the shorter backup path will preempt less low priority LSPs than the demand with the longer backup path. Therefore minimizing the length of the backup path(-s) possibly decreases the number of disrupted low priority LSPs.

For routing nominal paths a criterion implying load balancing on the links is used in the proposed strategy, as opposed to the shortest path routing used in the classical strategy. The idea behind the load balancing is to distribute traffic in a network in order to reduce the number of rejected future requests (blocking probability) because of the insufficient available capacity on some links. This is related to the principle of "minimum interference routing" [132]. What is also important in the resilience context is that the load balancing mechanism reduces the risk of several flows that traverse the same network segment between two given endpoints being disrupted by the same failure. That is due to spreading the flows on different paths, as opposed to a solution where all of them use the same shortest (sub-)path through that network segment.

The proposed routing strategy has not been considered before (to the best of our knowledge). As it was mentioned above, this strategy combines balancing of the load on the links implied by nominal paths and minimization of the length of the protection paths. First, we will discuss the two aspects of the proposed strategy, namely—load balancing and shortest path routing—and then we will present ways of combining them.

9.6.1 Load balancing on the links

Various objective functions can be used for problem FD-LSP in order to improve load balancing on the links. Before presenting the load balancing criteria, let us first define *residual bandwidth* (RB) on each link in the nominal situation as: $r_{e0} = c_e - y_{e0}$.

In the expression above r_{e0} and y_{e0} have slightly different (problem-specific) meaning, depending on whether problem SB-LSP or FD-LSP is considered. For problem SB-LSP, these two entities are equivalent with r_{ek} and y_{ek}, respectively, where $k = 0$, whereas for problem FD-LSP they are equivalent with r_{es} and y_{es}, respectively, where $s = 0$.

It is important what criterion is chosen for distribution of load on the links. There exist several possibilities, e.g., maximizing the total residual bandwidth on the links (expressed as $\sum_e r_{e0}$), max-min fair allocation of the residual bandwidths to the links, etc. Here we discuss three of the possible criterion and illustrate their load balancing effect with the help of the following example.

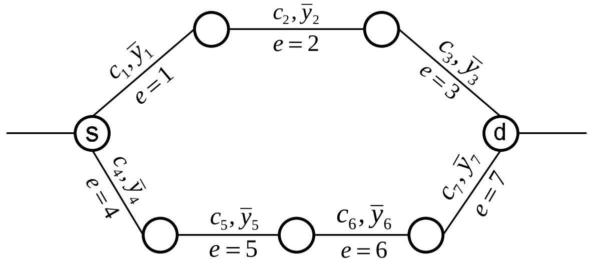


Figure 9.2: A network segment.

Example 9.2. *Consider a segment of a network as depicted in Figure 9.2, where* e *is a link index,* c_e *denotes link capacity and* $\bar{y}_e$*—link load. Assume that network is already partially loaded and suppose that as a part of a process of provisioning a new LSP in the network, a decision has to be taken by the load balancing mechanism, whether that new LSP has to pass the network segment between nodes* s *and* d *on path* $\mathcal{P}_1 = \{1, 2, 3\}$ *or* $\mathcal{P}_2 = \{4, 5, 6, 7\}$*. Here* $\{\cdot\}$ *denotes the set of indices of the links belonging to the path. Bandwidth to be reserved on the chosen path is* $h_d = 10$*. Consider three cases which will help illustrating the trade-offs between the path length and load balancing that each of the considered objective functions (criteria) imply: 1) capacities of all links are the same and equal to 100, i.e.,* $c_e = 100$*; loads of all links are the same and equal to 10, i.e.,*$\bar{y}_e = 10$*; 2)* $c_e = 100$*, for all* e*;* $\bar{y}_e = 11$ *for all* $e \in \mathcal{P}_1$ *and* $\bar{y}_e = 10$ *for all* $e \in \mathcal{P}_2$*; 3)* $c_e = 100$*, for all* e*;* $\bar{y}_e = 50$ *for all* $e \in \mathcal{P}_1$ *and* $\bar{y}_e = 10$ *for all* $e \in \mathcal{P}_2$*. Case illustrates what happens when a shorter or longer path is chosen for carrying additional flow if both paths are equally loaded. Case 2 illustrates what happens when a shorter but marginally more loaded path is chosen or a longer but less loaded path. Case 3 is provided for illustrating what happens when a shorter but considerably more loaded path is chosen versus a longer but much less loaded path.*

9.6.1.1 Max-min fair allocation of RBs

One possibility for balancing the loads is to use the max-min fairness principle. When applied to a vector of residual bandwidths $\boldsymbol{r}_0 = (r_{e0} : e = 1, 2, \dots, E)$, an MMF principle maximizes the minimum residual bandwidth available on all links. After the minimum value of RB is found for all links, the RB is increased further on the links on which this is possible, and the process iterates. It was discussed in Section 4.3.2.1 that finding a MMF allocation is equivalent to finding a lexicographically maximal vector among all feasible vectors sorted in non-decreasing order. Let us define a vector of residual bandwidths $\boldsymbol{r}_0$ sorted in the non-decreasing order as $\hat{\boldsymbol{r}}_0$, and a set of all such feasible vectors by $\bar{R}$. Then MMF allocation of residual bandwidths results from solving the following problem:

$$\text{lex max} \quad \{\hat{\boldsymbol{r}}_0 \ : \ \hat{\boldsymbol{r}}_0 \in \bar{R}\} \tag{9.15}$$

When solved, the problem provides a solution implying balanced load on the links just for the nominal state. This is the first step of a general algorithm, where the problem is similarly solved for every other state. Algorithms for resolving the problem were discussed in Section 5.3.5.

Let us now examine the possible outcomes of using MMF for load balancing. This we will do by revisiting Example 9.2 and the cases defined there. In the optimal solution of the problem, path $\mathcal{P}_1$ will be used in Case 1, while $\mathcal{P}_2$ will be used in the rest of cases. For the purpose of illustration, consider for example Case 1. The additional demand will be routed on path $\mathcal{P}_1$, since the resulting optimal RB vector $\hat{\boldsymbol{r}}_0^* = (80, 80, 80, 90, 90, 90, 90)$ in this case is lexicographically greater than the resulting vector $\hat{\boldsymbol{r}}_0 = (80, 80, 80, 80, 90, 90, 90)$ in the case where path $\mathcal{P}_2$ is used. Thus one may see that MMF criterion implies the most fair solution (i.e., the best balanced load), but at the expense of excessive bandwidth consumption. This is well illustrated by the result for Case 2, where the marginal difference of path loadings results in using the longer path $\mathcal{P}_2$ and consuming 10 capacity units more as compared to the case when $\mathcal{P}_1$ is used. In cases 1 and 3 load balancing using the MMF criterion results in the desired outcome.

9.6.1.2 Maximization of the total residual bandwidth

Another criterion for load balancing is maximization of the total residual bandwidth in the network (which is equivalent to minimization of the total load of network links):

$$\max \quad \sum_e r_{e0}. \tag{9.16}$$

Let us examine the load balancing effect of (9.16) when it is used with problem FD-LSP for the cases considered in the Example 9.2. In each case (9.16) will attain its maximum when the additional flow is routed on the shortest path $\mathcal{P}_1$. Thus this function is clearly another extreme case; namely, it implies routing of flows on shortest paths, so that as little as possible bandwidth is consumed. Thus, the function (9.16) does not take into account the load on a given link, as compared to the load on other links. Therefore it may not really be called a load balancing function. However, minimizing usage of network bandwidth is also a very important factor, and therefore this function has been included to the discussion. Objective function (9.16) behaves as desired only in Case 1 of Example 9.2.

9.6.1.3 Proportionally fair allocation of RBs

We suggest using the principle of proportional fairness for load balancing on the links. PF allocation is achieved by using the following objective:

$$\max \quad \sum_e \log\ r_{e0}. \tag{9.17}$$

Because of the particular properties of the logarithmic function, this objective will tend to distribute the residual capacity to links equally, at the same time avoiding allocation of very small residual bandwidth (this is the common case in the maximization of the total residual bandwidth), and not forcing maximization of the minimal allocation (the case of max-min fairness). These observations are confirmed by examining the load balancing effect of (9.17) used with problem FD-LSP for the cases defined in Example 9.2. The maximum of objective function (9.17) value is attained when the additional demand is routed on path $\mathcal{P}_1$ in Cases 1-2, and path $\mathcal{P}_2$ is used in Case 3. The fact that optimal solution implies using path $\mathcal{P}_1$ in Case 1 shows that a shorter path is prioritized if paths are equally loaded, thus saving the network bandwidth. As to Case 2, the fact that using path $\mathcal{P}_1$ is implied by the optimal solution shows that even if a shorter path is marginally more loaded than the longer path, it is still more profitable to use the shorter path $\mathcal{P}_1$ from the consumed bandwidth point of view for routing the additional demand. However, if path $\mathcal{P}_1$ is much more loaded than $\mathcal{P}_2$, as in Case 3, then the optimal solution implies using $\mathcal{P}_2$. Thus, objective function (9.17) behaves as desired in all the three cases. Therefore, PF is a reasonable trade-off between maximization of the total residual capacity and its max-min fair allocation.

Objective (9.17) is non-linear, which, combined with the non-convex solution space of FD-LSP, makes the problem very difficult to solve. In order to somewhat simplify the problem resolution, the logarithmic function in (9.17) can be linearized by using the piece-wise linear approximation discussed in Section 4.3.4. The approximation implies that the following constraints are added to the problem

$$f_e \leq a_i r_{e0} + b_i \ , \ \forall i \in \mathcal{I}, \ \forall e \in \mathcal{E}. \tag{9.18}$$

and objective function (9.17) is replaced with:

$$\max \quad \sum_e f_e. \tag{9.19}$$

9.6.2 Shortest path routing of backup paths

Minimizing the length of backup paths is desirable in order to minimize a number of disrupted low priority LSPs. The following objective function, when used with problem FD-LSP, implies minimization of the length of backup paths:

$$\min \quad \sum_{s \neq 0} \sum_e \sum_d \sum_p \xi_e \delta_{edp} v_{dps}. \tag{9.20}$$

The objective function (9.20) implies minimization of the lengths of the paths for all demands in all failure situations ($s \neq 0$), when link costs $\xi_e \equiv 1$ (and thus can be interpreted as a hop-count). If the costs are different than 1, the objective function implies using the cheapest (in some desirable sense) paths instead.
Similarly, the objective function for problem SB-LSP is as follows:

$$\min \quad \sum_e \sum_d \sum_p \xi_e \delta_{edp} v_{dp1}. \tag{9.21}$$

9.6.3 Combining the two criteria

Sections 9.6.1 and 9.6.2 have dealt with two important aspects of the proposed routing strategy, namely–load balancing in the nominal network state and using shortest backup path(-s) in case of failures.

One way of combining the PF load balancing (9.19) for nominal state and minimization of lengths of backup paths (9.21) or (9.20) is by solving a problem in question (SB-LSP or FD-LSP) in two stages. For example, in case of FD-LSP, first the problem (9.19), (9.6)-(9.10) and (9.18) is solved for the nominal situation. Then flow allocation pattern for nominal situation is fixed, and problem (9.20),(9.6)-(9.10) is solved for all failure situations $s \neq 0$.

Another approach (it was used for the experiments reported in this chapter) is to combine the two criteria in a single objective function. For problem FD-LSP the combined objective is as follows:

$$\min \quad -A\sum_e f_e + (1-A)\sum_{s\neq 0}\sum_e\sum_d\sum_p \xi_e\delta_{edp}v_{dps}. \tag{9.22}$$

Similarly, for problem SB-LSP the objective function is:

$$\min \quad -A\sum_e f_e + (1-A)\sum_e\sum_d\sum_p \xi_e\delta_{edp}v_{dp1}. \tag{9.23}$$

Coefficient A is used for weighting the two criteria in the objective function. In the numerical experiments reported in the next section A was set to 0.5. The final problem with failure-dependant backup path protection is then defined as follows:

Problem FD-LSP-COMB:

$$\begin{aligned}
&\text{obj} \quad (9.22) \\
&\text{s.t.:} \quad \sum_p v_{dps} = 1\,, && \forall d\in\mathcal{D}\,,\forall s\in\mathcal{S} && (9.24)\\
&v_{dps} \geq \theta_{dps}v_{dp0}\,, && \forall d\in\mathcal{D}\,,\forall p\in\mathcal{P}_d\,,\forall s\in\mathcal{S} && (9.25)\\
&\sum_d\sum_p \delta_{edp}v_{dps}h_d \leq y_{es}\,, && \forall e\in\mathcal{E}\,,\forall s\in\mathcal{S} && (9.26)\\
&y_{es}\leq \alpha_{es}c_e\,, && \forall e\in\mathcal{E}\,,\forall s\in\mathcal{S} && (9.27)\\
&f_e \leq a_i r_{e0} + b_i\,, && \forall e\in\mathcal{E}\,,\forall i\in\mathcal{I} && (9.28)\\
&v_{dps}\in\mathbb{B}\,, && \forall d\in\mathcal{D}\,,\forall p\in\mathcal{P}_d\,,\forall s\in\mathcal{S} && (9.29)\\
&y_{es}\in\mathbb{R}^+\,, && \forall e\in\mathcal{E}\,,\forall s\in\mathcal{S}. && (9.30)
\end{aligned}$$

In the problem formulation presented above we have eliminated auxiliary variables x_{dps} by substituting the right-hand side of constraint (9.7) into constraints (9.8) and (9.9), this way obtaining constraints (9.25) and (9.26).

9.7 Numerical experiments

Several numerical experiments have been conducted in order to test the effectiveness of the proposed protection and load balancing strategy. The networks used for experiments were: network N_{12} with 12 nodes, 22 links, 6 to 14 paths per demand and 66 demands; network N_{25} with 25 nodes, 56 links, 10 paths per demand and 300 demands. The experiments were composed of two stages. First, the optimization problem FD-LSP-COMB, defined in AMPL, was solved (also in the two-phase form described in Section 9.6.3) by CPLEX 9.1 solver for the considered networks. The solution time was below one minute for the N_{12} network and below two minutes for the N_{25} network on a PC with Pentium 4 HT 3GHz CPU and 2GB of RAM. All the single-link failure states were assumed in the experiments. The solution of the optimization problem has found, for every high priority (HP) demand, a primary path and backup paths for all failure situation. All the HP demand volumes were set to the same value and were equal to about 2% of a single link capacity. Link capacities were the same for all links. The demand volumes have been chosen such that in all situations the network is loaded by nominal flows of all the HP demands to about 25%.

In the second stage the routing information of the HP demands, produced by the optimization program, was sent to the Advanced CSPF Simulator [133], which, based on the input information, performed a setup of the HP LSPs. Then additional demand requests were generated, with priorities raging from 1 to 4 (hence, not with the high priority 0) and demand volume uniformly distributed between 1% and 5% of a link capacity. In the sequel we will refer to these as low priority (LP) demands. The probability of existence of a demand between a given source-destination pair was the same for all the pairs. There were 1500 such demands generated for network N_{12} and 2300 for the network N_{25}. Just nominal paths were found using CSPF and provisioned for these dynamically arriving LP LSPs and no backup paths provisioned. The number of the LP LSPs was chosen so that after placing them the network would be (nearly) saturated. Simulation section describes in detail different strategies used for routing these additional LSPs. When all the additional LSPs that could be placed were placed, the network failure states were simulated one by one. All the single link failure scenarios were considered. Our goal was to measure the number of surviving LSPs in the network in the case of each failure scenario. No new arrivals occurred in the failure states of the network. However, the failures required backup paths of HP LSPs to be activated, preempting in turn some of the LP LSPs. Then an attempt was made by the simulator to restore as many of the preempted LP LPSs as possible, by dynamically finding new paths for them. Only when all the preemption and possible restorations were over, the number of surviving LSPs in the network was recorded. The different strategies for restoration of the LP LSPs are presented in the section below.

In the experimental framework described above we have compared the proposed FDBP protection mechanism with off-line pre-planned SBP protection (model SB-LSP-MINHOP), as well as with the dynamic online SBP recovery mechanism. For the online routing mechanism, we have considered two options for link metrics.

9.7.1 Link metrics in simulations

Two different link metrics for the on-line routing of LP LSPs were considered in the simulator. They were used by the CSPF protocol for finding a path for the dynamically arriving LP LSP request, as well as when a restoration path for the LP LSPs had to be found.

The first metrics is an interior gateway protocol (IGP) cost. It is simply some administrative cost ξ_e, assigned to links. In the experiments we have assumed $\xi_e = 1$ for all links, thus IGP merely representing a hop count.

The second metrics is an inverse of the priority-based residual bandwidth R^j_{es}, which for an LSP request with priority j and link e in situation s is expressed as: $R^j_{es} = r_{es} + \mu^{H(j)}_{es} + \eta^{L(j)}_{es}$. Here, r_{es} is, as before, the free (unused) bandwidth on link e in situation s, $\mu^{H(j)}_{es}$ is a bandwidth reserved on link e for all backup paths of HP demands (in general, LSPs with priority levels higher than j). This bandwidth can be used by LP LSPs if the protection paths of HP LSPs are not activated in a given situation s. Similarly, $\eta^{L(j)}_{es}$ is a bandwidth reserved on link e in situation s for all primary LSPs of the demands with priority levels lower than j, and thus can be preempted by j (mind that lower priority number implies higher priority level). Finally, when finding a constrained shortest path, the quantity $1/R^j_{es}$ is used as the link metrics.

9.7.2 Test scenarios

Because of multiple dimensions of the input data and possible results, further in this discussion we focus only on the case of the larger network N_{25}. The test scenarios are summarized in Table 9.1.

In Cases I and II, both nominal and backup paths are calculated off-line. Problem FD-LSP-COMB is used to find nominal and protection paths in Case I, whereas in Case II the problem FD-LSP-COMB is modified to imply single backup paths. In cases III and IV only the nominal paths were calculated off-line by using problem FD-LSP-COMB, and the (single) protection paths were found on-line and provisioned together with the nominal path. Cases III and IV differ by the link metrics (see Table 9.1), which was used for finding the protection paths. In case III, protection paths were calculated using $1/R^j_{e0}$ as link metrics, whereas in case IV the metrics was the IGP cost. In cases V and VI both nominal and single protection paths were calculated on-line, where the link metrics for finding primary paths was $1/R^j_{e0}$. The protection paths were calculated using $1/R^j_{e0}$ as a link metrics in Case V and IGP cost in Case VI.

Case	Placement strategy	
	HP nominal paths	HP backup paths
I	off-line	off-line, FDBP
II	off-line	off-line, SBP
III	off-line	on-line, $1/R^j_{e0}$
IV	off-line	on-line, IGP
V	on-line, $1/R^j_{e0}$	on-line, $1/R^j_{e0}$
VI	on-line, $1/R^j_{e0}$	on-line, IGP

Table 9.1: Test scenarios.

Case	DNF	+F,%	CL	HCL,%
I	39.43	–	2.84	57.34
II	43.95	11.46	2.93	61.73
III	43.70	10.83	2.93	58.79
IV	42.57	7.96	2.89	57.96
V	42.59	8.01	2.91	57.39
VI	43.05	9.18	2.95	57.38

Table 9.2: Comparison of the recovery mechanisms.

Thus, in Cases I and II the calculation of primary and backup paths for HP demands was performed completely off-line. Cases III and IV present a combined approach, where primary paths were calculated off-line and the protection paths were calculated on-line. And in Cases V and VI all the paths were calculated on-line. Nominal paths for the LP traffic were always calculated on-line using the quantities $1/R^j_{e0}$ as link metrics, and no backup paths for the LP traffic were provisioned.

9.7.3 Results and discussion

Table 9.2 presents results for the cases described in Section 9.7.2. The considered recovery mechanisms are compared in terms of the average number of failing LSPs per failure situation (column DNF in the Table 9.2). For each case it is calculated as a difference between number of LSPs placed in the nominal situation and the average number of LSPs surviving after the failures. Second column (+F) in the table shows increase in failing LSPs for Cases II-VI as related to case I (it is calculated from the values in the DNF column). The number of LSPs placed in the nominal situation was very close in all the cases and was on average equal to 1546.33. The table also shows the maximum cascading level (column CL) and the percentage of heavily congested links (column HCL) for each case (both numbers are averages over all the failure situations). Cascading level is related to preemption mechanism and LSP priorities. Each LSP can preempt any of the lower priority LSPs, which in turn can preempt yet lower priority LSPs and so on. Thus CL gives how many priority levels down does the preemption process propagate to. A link is regarded heavily congested if it is loaded not less than by 70% of its capacity.

As can be seen from Table 9.2, using the proposed protection mechanism with failure-dependent backup paths resulted in almost 11% less disrupted LSPs per failure situation (on average), as compared to the single backup path protection, where the paths are calculated off-line (Case II). Thus, when using the FDBP protection mechanism it is possible to retain more traffic in the network in the case of failures as compared to SBP protection, using the same network resources. This results in better network resource utilization. Actually, SBP with off-line calculated backup paths appeared to be the worst among all the considered options. The closest result to the FDBP case is achieved by using an off-line calculated nominal path and on-line calculated single disjoint backup path (see entry IV in Table 9.2), using IGP as the link metrics. Since all the IGP costs are equal to 1, this implies shortest path routing for the backup paths. This result also confirmed that choosing to minimize the length of the failure-dependent backup paths in the problem FD-LSP-COMB is beneficial. The results for complete on-line routing cases V and VI were worse than for Case IV but better than for case III. The CL and HCL values were also best (lowest) for the FDBP case. The worst CL was for Case VI, and the worst HCL was again for Case II (SBP). The lowest HCL value for the FDBP case is due to the load balancing. The lower CL value can in general imply lower number of preempted LSPs.

9.8 Conclusion

In this chapter we considered a combined problem of routing, load balancing and traffic recovery in an IP/MPLS network carrying traffic with different priority classes. An efficient recovery mechanism (protection with failure-dependent backup paths) is studied and a routing strategy combining shortest path routing and load balancing is proposed. A combined numerical and simulation-based study has revealed that for the considered network examples using the proposed protection mechanism and routing strategy, the average number of discarded LSPs per failure situation is lower by at least 7% than for the other considered recovery options. The off-line single backup path protection was worst among all the considered cases, outperformed even by the complete on-line routing and recovery strategies. Also, the experiments confirmed that using the shortest (hop count) paths for protection of nominal paths results in a better performance.

CHAPTER 10

Contributions, conclusions and future work

10.1 Scientific achievements

In this thesis we have studied topics in fair flow allocation and network resilience applied to design of single and two-layer networks carrying combined elastic and non-elastic traffic. In particular, network dimensioning problems have been studied for both normal network state and taking failure states into account. Solutions of such problems consist in optimal link capacities and demand flows that can be achieved within a given budget for installing link capacities. The studied optimization models assume fair bandwidth allocation among the demands. The developed models employ fairness in two aspects: fair bandwidth distribution between demands in a given network operation state, and fair distribution of revenues between the states. Since the studied problems are of the multi-criteria type, we have presented two algorithms based on lexicographical maximization. The combinations of max-min fairness, proportional fairness, and revenue maximization have been investigated, and it has been shown that the best solution to the tradeoff between total network throughput and fairness of flow allocation is achieved when using PF for assigning bandwidth to demands, and MMF for distributing the revenue between the network operation states.

Furthermore, assuming that such two-dimensional fairness is assumed in a multi-layer network, we have studied efficiency of performing recovery in different network resource layers. For this we have proposed network models and algorithms where the fairness model and different recovery options are combined. We have studied the efficiency of performing recovery in different layers of a two-layer network by relaxing the associated MIP problems and investigating an upper bound on total network revenue attained by each recovery option. It appeared that, although allowing coordinated reconfiguration in both layers allows for the most efficient network bandwidth usage, a recovery only in the upper network layer is quite close in terms of cost performance if symmetrical demands are imposed. For networks with asymmetrical demands, coordinated recovery in both layers outperforms recovery in the upper layer. It has also been shown that both recovery options significantly outperform recovery only in the lower layer. It should be noted, however, that recovery time and complexity aspect is not taken into account in the investigations. (In general, recovery mechanisms only in the lower layer provide the fastest and simplest recovery.)

For the discussed single and two-layer network design problems we have developed path generation algorithms and showed what maximum gain could be achieved by introducing a new path to a routing list.

Realistic network features and functionality can in fact be captured only by MIP (mixed-integer programming) models which in general are difficult to resolve. The presence of multiple layers of resources make the resolution of optimization models even more difficult as they increase the number of variables and constraints. For multi-layer problems we have developed a generic resolution framework and an efficient heuristic algorithm. The algorithm consists in decomposing a multi-layer network into separate resource layers and solving a design sub-problem for each of them separately, while exchanging certain cost/demand information between the design problems and iterating the whole process. A variety of design problems with different objective functions and recovery mechanisms can be effectively resolved within the introduced framework. Actually, the framework combines a heuristic part with a part resolving a MIP sub-problem. Certainly, the iterative method requires solving many sub-problems, but the sub-problems are much less complex than the original problem that is to be solved by a direct application of a MIP solver. Also, the MIP task solved as a part of the iterative algorithm is relatively small and thus in most cases effectively handled by a commercial solver. Besides, it is important that the introduced framework allows to effectively take modular link capacities into account. Time and solution quality of the proposed iterative method have been compared with exact optimal solutions, with solutions for which optimality is relaxed by 10%, and with partial linear relaxation of the optimization model. Numerical examples show that the gap between the heuristic solution of the iterative algorithm and exact solutions for the revenue and throughput maximization problems is close to zero, while for the cost minimization problems it is less than 10%. The heuristic algorithm exhibits computation time much shorter than the exact algorithm. The heuristic solution is equally good or better than the one produced by a commercial solver with the relaxed optimality requirement, and in all but one cases the solution time was shorter for the heuristic approach. A numerical study demonstrates that algorithms derived from the iterative method converge quickly to near-optimal solutions. This suggests rather good performance of the iterative method for a variety of two-layer network design problems with various recovery options and goal functions.

Finally, we have developed an integrated recovery, routing, and load balancing strategy for an IP/MPLS network. It is our belief that efficiency of network resource utilization can be improved by combining all these three aspects, and taking them into account during the network planning phase. The presented model assumes label switched paths (LSPs) with different priority levels, where demands with higher priority can preempt those with lower priority. Protection is provided only for the highest priority LSPs by a failure-dependent backup path protection mechanism, using shortest (by hop-count) paths for protection of nominal paths. We discuss different ways of balancing the load on links and suggest distributing residual link bandwidths according to the proportional fairness principle. The proposed integrated strategy is compared with single backup path protection and a number of combined on-line/off-line recovery strategies. The numerical study shows that the proposed strategy results in the lowest number of failing LSPs in case of a failure, the lowest cascading level (when a preempted path can preempt other paths with lower priority) and the lowest number of highly congested links in the network.

To summarize, the path followed in the thesis starts by studying fair bandwidth allocation in single and two-layer networks. For the two-layer network problems the efficiency of performing recovery in different network layers is studied and selected models for various recovery mechanisms are stated. Path generation algorithms are developed for all the considered single and two-layer network problems. A framework for resolving two-layer MIP network design problems is then presented and problem-specific heuristic algorithms are derived for several selected problems. Finally, an integrated routing, recovery, and load balancing strategy for IP/MPLS networks is proposed. In effect, the models and algorithms presented in the thesis cover a broad range of the NGI network design problems, including fair resource allocation, recovery in different resource layers, path generation, load balancing, and a strategy to effectively combine different network design aspects into the single design problem.

10.2 Future work

The work presented in the thesis can be extended in many different aspects, such as extending network models, developing new resolution algorithms for the specified problems or even performing a testbed study for a selected set of the problems.

Modeling-wise, the following extensions would be a natural follow-up of the presented work. Optimization models presented in the thesis do not differentiate among services to which demands correspond. Thus, no distinction is made on whether the bandwidth requested for a demand is, for example, related to FTP file transfers or an interactive video. It would be interesting to extend our multi-layer model by introducing an additional (top) layer of service overlays aiming at prescribing demands to particular service classes. Modifying problems of Chapters 5 and 6 in this way, one could define a minimum bandwidth requirement for capacity on links of a particular overlay network (service), and require that the excess bandwidth on links is distributed in a fair way among different overlay networks. For a particular service class, bandwidth can also be distributed among demands in, for example, proportionally fair way. Such an assumption could be of interest for network operators who, in a presence of constantly increasing P2P traffic, want to reserve a required amount of bandwidth in their networks for other important services, such as e-mail. Each such service can be defined as an overlay in the proposed model.

Another extension to the models could be to consider link capacity modules of different size, in particular in the iterative resolution framework presented in Chapter 8. Also, one could explicitly allow user demand requests in different network layers, not only in the uppermost layer. Such an extension would allow to account for requests of different granularity.

For the considered optimization models there are various possibilities to investigate different exact, heuristic, and approximate resolution methods. In particular, an approximate method proposed by Garg et al. in [134] can be applied to relaxations of the problems considered in Chapters 5 and 6, as well as a part of the considered path generation algorithms (Chapter 7) for resolving the pricing problems. A closer look at the method of Garg et al. and the iterative method presented in Chapter 8 of the thesis can probably even lead to efficient combined method for multi-layer problems, which can further be combined with path generation algorithms. Also, B&C algorithms can be developed, and local branching [135] method, Feasibility Pump [136; 137] method, or a method [138] combining Feasibility Pump with local branching, can be adapted for the considered MIP problems, such as the one in the iterative resolution method (Chapter 8).

Results of the design methods could be tested, compared and validated using a testbed.

Bibliography

[1] D. Comer. *Internetworking With TCP/IP, Volume 1: Principles Protocols, and Architecture, 4th edition*. Prentice Hall, 2000.

[2] Wikipedia, the Free Encyclopedia, `http://www.wikipedia.com`.

[3] W. D. Grover. *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*. Prentice Hall PTR, Upper Saddle River, New Jersey, USA, 2003.

[4] Architecture of optical transport networks. *ITU-T Rec. G.872*, November 2001.

[5] Architecture for the automatic switched optical network (ASON). *ITU-T Rec. G.8080/Y.1304*, November 2001.

[6] Requirements for automatic switched transport networks (ASTN). *G.807/Y.1302*, 7 2001.

[7] Generalized multi-protocol label switching architecture. *IETF draft-ietf-ccamp-gmpls-architecture-07.txt*, May 2003.

[8] S. Tomic, B. Statovci-Halimi, A. Halimi, W. Muellner, and J. Fruehwirth. ASON and GMPLS – overview and comparison. *Photonic Network Communications*, 7(2):111–130, March 2004.

[9] J.-P. Vasseur, M. Pickavet, and P. Demeester. *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP and MPLS*. Morgan Kaufmann Publishers (Elsevier), 2004.

[10] ITU-T Standardization Organization. ITU-T Recommendation M.3000. Overview of TMN reccomendations, `http://www.itu.int`, 1995.

[11] W. Stallings. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2 (3rd Edition)*. Addison-Wesley Professional, 1999.

[12] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP tunnels. *IETF RFC3209*, December 2001.

[13] L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty, and A. Malis and. Constraint-based LSP setup using LDP. *IETF RFC3212*, January 2002.

[14] D. Katz, K. Kompella, and D. Yeung. Traffic engineering (TE) extensions to OSPF version 2. *IETF RFC3630*, September 2003.

[15] H. Smit and T. Li. Intermediate system to intermediate system (IS-IS) extensions for traffic engineering (TE). *IETF RFC3784*, June 2004.

[16] L. Berger. Generalized multi-protocol label switching (GMPLS) signaling functional decsription. *IETF RFC 3471*, January 2003. http://www.ietf.org/rfc/rfc3471.txt.

[17] Generalized multi-protocol label switching (GMPLS) signaling resource reservation protocol-traffic engineering (RSVP-TE) extensions. *IETF RFC3473*, January 2003.

[18] Generalized multi-protocol label switching (GMPLS) signaling constraint-based routed label distribution protocol (CR-LDP) extensions. *IETF RFC3472*, January 2003.

[19] M. To and P. Neusy. Unavailability analysis of long-haul networks. *IEEE Journal on Selected Areas in Communications*, 12(1):100–109, January 1994.

[20] D. Crawford. Fiber optic cable dig-ups – causes and cures. *Network Reliability: A Report to the Nation – Compendium of Technical papers*, June 1993.

[21] W. Lay and D. McDysan. Network hierarchy and multilayer survivability. *IETF RFC 3386*, November 2002.

[22] Colle D., De Maesschalck S., Develder C., Van Heuven P., Groebbens A., Cheyns J., Lievens I., Pickavet M., Lagasse P., and Demeester P. Data-centric optical networks and their survivability. *IEEE Journal on Selected Areas in Communications*, 20(1):6–20, 2002.

[23] Pin-Han Ho and H. T. Mouftah. A framework for service-guaranteed shared protection in WDM mesh networks. *IEEE Communications Magazine*, 40(2):97–103, 2002.

[24] T. Jensen. Planning dependable network for IP/MPLS over optics. *Telektronikk*, 99(3/4):128–162, 2003.

[25] J. Zhang and B. Mukherjee. A review of fault management in WDM mesh networks: Basic concepts and research challenges. *IEEE Network*, 18(2):41–48, March/April 2004.

[26] J. Wang, L. Sahasrabuddhe, and B. Mukherjee. Path vs. subpath vs. link restoration for fault management in IP-over-WDM network: Performance comparisons using GMPLS control signaling. *IEEE Communications Magazine*, 40(11):80–87, November 2002.

[27] S. Ramamurthy and B. Mukherjee. Survivable WDM mesh networks, part II - restoration. In *International Conference on Communications (ICC), 3*, pages 2023–2030, Vancouver, BC, 1999.

[28] A. Fumagalli and L. Valcarenghi. IP restoration vs. WDM protection: Is there an optimal choice? *IEEE Network*, 14(6):34–41, November/December 2000.

[29] L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee. Fault management in IP-Over-WDM networks: WDM protection versus IP restoration. *IEEE Journal on Selected Areas in Communications*, 20(1):21–33, January 2002.

[30] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computations, and Application*. John Wiley & Sons, New York, USA, 1986.

[31] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers (Elsevier), USA, 2004.

[32] M. Minoux. *Mathematical Programming: Theory and Algorithms*. John Wiley & Sons, 1986.

[33] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, New Jersey, USA, 1982.

[34] E. L. Lawler and D. E. Wood. Branch-and-bound method: A survey. *Operations Research*, 14(4):699–719, July–August 1966.

[35] A. H. Land and A. G. Doig. An authomated method of solving discrete programming problems. *Econometrica*, 28(3):497–520, July 1960.

[36] L. A. Wolsey. Strong formulations for mixed integer programming: A survey. *Mathematical Programming*, 45:173–191, 1989.

[37] E. Balas, S. Ceria, and G. Cornuejols. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, (42):1229–1246, September 1996.

[38] M. Padberg. Classical cuts for mixed-integer programming and branch-and-cut. *Mathematical Methods of Operations Research*, 53(2):173–203, 2001.

[39] F.R.B. Cruz, G.R. Mateus, and J. Macgregor-Smith. A Branch-and-Bound algorithm to solve a multi-level network optimization problem. *Journal of Mathematical Modelling and Algorithms 2*, pages 37–56, 2003.

[40] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, 1998.

[41] M. Padberg. *Linear Optimization and Extensions*. Springer-Verlag, 1991.

[42] O. Günlük. A branch-and-cut algorithm for capacitated network design. *Mathematical Programming*, A(86):17–39, 1999.

[43] C. Bienstock and G. Muratore. Strong inequalities for capacitated survivable network design problems. Technical report, Technical Report 3, CORC Report, Mathematical Programming, Columbia University, 1999.

[44] M. Minoux. Network synthesis and optimum network design problems: Models, solution methods and applications. *Network*, 19:313–360, 1989.

[45] C. Barnhart, C. Hane, and P. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.

[46] J. Geffard. A solving method for singly routing traffic demand in telecommunication networks. *Annales des Telecommunicationes*, 56(3-4):140–149, 2001.

[47] Z. Gu, G. L. Nemhauser, and M. W. P. Savelsbergh. Lifted cover inequalities for 0–1 integer programs: computation. *INFORMS Journal on Computing*, 10:427–438, 1998.

[48] M. Belaidouni and W. Ben-Ameur. Superadditive approach to solve the minimum cost single path routing porblem: Preliminary results. In *Proceedings INOC'2003*, pages 67–71, 2003.

[49] ILOG CPLEX. `http://www.ilog.com`.

[50] XPRESS-MP Software. `http://www.dashoptimization.com`.

[51] S. Shenker. Fundamental design issues for the future Internet. *IEEE Journal on Selected Areas in Communications*, 13(7):1176–1188, 1995.

[52] Peer-to-peer in 2005. Cachelogic corporation, `http://www.cachelogic.com`, 2005.

[53] Marc Morin. Network neutrality: a broadband wild west? Sandvine Incorporated, `http://www.sandvine.com`, 2005.

[54] P2P survey 2006 - extended abstract. Ipoque, `http://www.ipoque.com/pub/P2P-Survey-2006.pdf`, 2006.

[55] Joost. `http://www.joost.com`.

[56] Lucy Sherriff. BBC ponders P2P distribution. *The Register*, February 17 2004.

[57] Gary Flood. BBC turns to P2P for VOD. Streamingmedia, `http://www.streamingmedia.com`, November 29 2005.

[58] Dense wavelength division multiplexing (DWDM). Technical report, International Engineering Consortium.

[59] Introduction to DWDM for metropolitan networks. Technical report, Cisco Systems, 2000.

[60] Mark Laubach. Residential area CATV broadband Internet technology: Current status. *The Internet Protocol Journal*, 1(3):13–26, December 1998.

[61] Cable (CATV) data networks. Tom Sheldon's Linktionary, `http://www.linktionary.com/`.

[62] Kieren McCarthy. BT admits to bandwidth restrictions for file-sharing sites. *The Register*, October 8 2001.

[63] John Oates. NTL/Telewest admits limiting unlimited web access. *The Register*, November 21 2006.

[64] Chris Williams. Virgin throttles national cable network. *The Register*, May 8 2007.

[65] Chris Williams. Another ISP throttles bandwidth. *The Register*, May 23 2007.

[66] Corey Boles. FCC launches inquiry into net neutrality. *Wall Street Journal*, March 22 2007.

[67] Keep the Internet free and open! Common Cause, `http://www.commoncause.org`.

[68] Cable service providers: How to reduce costs and attract new business. Allot Communications, `http://www.allot.com`.

[69] Managing peer-to-peer traffic with Cisco service control technology. Cisco Systems, `http://www.cisco.com`.

[70] A.C. Pigou. *Wealth and Welfare*. Macmillan, 1912.

[71] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, N.J., London, England, 1987.

[72] G. Fodor, G. Malicsko, M. Pióro, and T. Szymanski. Path optimization for elastic traffic under path constraints. In *17^{th} International Teletraffic Congress*, pages 667–680, Salvador de Bahia, Brasil, 2001.

[73] Y.T. Hou, H.H-Y. Tzeng, and S.S. Panwar. A generic weighted-based network bandwidth sharing policy for ATM ABR service. In *IEEE ICC'98*, pages 1492–1499, Atlanta, GA, USA, 1998.

[74] K. Karand, S. Sarkar, and L. Tassiulas. Optimization based rate control for multipath sessions. In *17^{th} International Teletraffic Congress*, pages 805–816, Salvador de Bahia, Brasil, 2001.

[75] F. Kelly, A. Maulloo, and D.K.H. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252, 1998.

[76] T. Cinkler and P. Laborczi. Joint routing and bandwidth allocation for protected elastic traffic. In *Proc. of 10th International Telecommunication Network Strategy and Planning Symposium, Networks 2002*, Munchen, Germany, June 23-27 2002.

[77] E. A. Pazner. Pitfalls in the theory of fairness. *Journal of Economic Theory*, 14:458–466, 1977.

[78] R. Denda, A. Banchs, and W. Effelsberg. The fairness challenge in computer networks. In J. Crowcroft, J. Roberts, and M. Smirnov, editors, *QofIS 2000, LNCS*, volume 1922, pages 208–220. Springer-Verlag Berlin Heidelberg, 2000.

[79] John Rawls. *A Theory of Justice*. The Belknap Press of Harvard University Press, 1971.

[80] J. Kleinberg J, Y. Rabani, and E. Tardos. Fairness in routing and load balancing. In *IEEE 40th Annual Symposium on Foundations of Computer Science*, pages 568–578, 1999.

[81] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, January/February 1997.

[82] G. Bannock, R.E. Baxter, and E. Davies. *The Penguin Dictionary of Economics*. Penguin Books Ltd, 1998.

[83] L. Massoulie and J. Roberts. Bandwidth sharing: objectives and algorithms. *IEEE/ACM Transactions on Networking*, 10(3):320–328, June 2002.

[84] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, October 2000.

[85] C.-Q. Yang and A. V. S. Reddy. A taxonomy for congestion control algorithms in packet switched networks. *IEEE Network*, 9(4):34–45, July/August 1995.

[86] D. Hercog. A new classification of mechanisms for flow/congestion control. In *EUROCON 2003. Computer as a Tool. The IEEE Region 8*, volume 1, pages 247–250, Ljubljana, Slovenia, September 22-24 2003.

[87] J.-Y. Le Boudec. Rate adaptation, Congestion Control and Fairness: A Tutorial. Technical report, Ecole Polytechnique Federal de Lausanne (EPFL), December 21 2006.

[88] J. Mahdavi and S. Floyd. The TCP-Friendly Website. `http://www.psc.edu/networking/projects/tcpfriendly/`, June 1999.

[89] D. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avaoidance in computer networks. *Computer Networks and ISDN Systems*, (17):1–14, 1989.

[90] M. Vojvonovic, J.-Y. Le Boudec, and C. Boutremans. Global fairness of additive-increase and multiplicative-decrease with heterogenous round trip times. In *Proc. of IEEE INFOCOM 2000*, volume 3, pages 1303–1312, Tel-Aviv, Isreael, March 26-30 2000.

[91] P. Hurley, J.-Y. Le Boudec, and P. Thiran. A note on the fairness of additive increase and multiplicative decrease. In *Proc. of 16th International Teletraffic Congress (ITC 16)*, pages 467–478, Edinburgh, Scotland, June 1999.

[92] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts. On the scalability of Fair Queueing. In *ACM SIGCOMM HotNets III*, San Diego, CA, USA, 2004.

[93] D. Lin and R. Morris. Dynamics of random early detection. In *Proc. of ACM SIGCOMM*, 1997.

[94] I. Stoica, S. Shenker, and H. Zhang. Core -stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks. In *SIGCOMM'98*, pages 118–130, 1998.

[95] J. Nagle. On packet switches with infinite storage. RFC 970, IETF, 1985.

[96] P. Goyal, H. Vin, and H. Cheng. Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks. *IEEE/ACM Transactions on Networking*, 5(5):690–704, October 1997.

[97] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. *IEEE/ACM Transactions on Networking*, 4(3):375–385, June 1996.

[98] Congestion Management Overview. Cisco Systems, `http://www.cisco.com`, 2006.

[99] B. Bensaou, Y. Wang, and C. C. Ko. Fair medium access in 802.11 based wireless ad-hoc networks. In *Proc. of Mobile and Ad Hoc Networking and Computing (MobiHOC)*, pages 99–106, Boston, MA, USA, 2000.

[100] M. A. Marsan, A. Bianco, E. Leonardi, A. Morabito, and F. Neri. All-optical WDM multi-rings with differentiated QoS. *IEEE Communications Magazine*, 37(2):58–66, 1999.

[101] Z. Dziong and L. G. Mason. Fair-efficient call admission control policies for broadband networks-a game theoretic framework. *IEEE/ACM Transactions on Networking*, 4(1):123–136, February 1996.

[102] Y.-C. Lai and Y.-D. Lin. A novel admission control for optimal fairness. In *IEEE Global Telecommunications Conference (GLOBECOM '02)*, volume 2, pages 1905–1909, November 17-21 2002.

[103] A. Arulambalam, X. Chen, and N. Ansari. Allocating fair rates for available bit rate service in ATM networks. *IEEE Communications Magazine*, 34(11):92–100, 1996.

[104] E. J. Hernandez-Valencia, L. Benmohamed, S. Chong, and R. Nagarajan. Rate control algorithms for the ATM ABR service. *European Transactions on Telecommunications*, 8(1):7–20, 1997.

[105] S. Xu and B. Xu. A fair admission control scheme for multimedia wireless network. In *Proc. of Wireless Communications, Networking and Mobile Computing*, volume 2, pages 859–862, September 23-26 2005.

[106] S. Thiagarajan and A. K. Somani. Capacity fairness of WDM networks with grooming capabilities. In I. Chlamtac, editor, *SPIE: Proc. of Optical Networking and Communications (OptiComm 2000)*, volume 4233, pages 191–201, September 2000.

[107] K. Mosharaf, J. Talim, and I. Lambadaris. A call admission control for service differentiation and fairness management in WDM grooming networks. In *Proc. of Broadband Networks (BroadNets 2004)*, pages 162–169, October 25-29 2004.

[108] D. Nace and M. Pióro. A tutorial on max-min fairness and its applications to routing and load-balancing in telecommunication networks. To apper in IEEE Communications Surveys and Tutorials.

[109] P. Nilsson and M. Pióro. Solving dimensioning tasks for proportionally fair networks carrying elastic traffic. *Performance Evaluation*, 49(1–4):371–386, September 2002.

[110] M. Pióro, G. Malicsko, and G. Fodor. Optimal link capacity dimensioning in proportionally fair networks. In *Networking 2002*, pages 277–288, Pisa, Italy, 2002.

[111] P. Nilsson. *Fairness in communication and computer network design*. PhD thesis, Lund University, 2006.

[112] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[113] L.S. Lasdon. *Optimization Theory for Large Systems*. Macmillan, New York, USA, 1970.

[114] M. Pióro. A robust dimensioning problem for proportionally fair networks. Technical Report CODEN: LUTEDX(TETS-7183)/1-3/(2002)&local 8, Department of Communication Systems, LTH, Lund, Sweden, February 2002.

[115] E. Marchi and J.A. Oviedo. Lexicographical optimality in the multiple objective linear programming: the nucleolar solution. *European Journal of Operational Research*, 57(3):355–359, 1992.

[116] W. Ogryczak. On the lexicographic minimax approach to location problems. *Euoropean Journal of Operational Research*, 100(3):566–585, 1997.

[117] A. Tomaszewski. A polynomial algorithm for solving a general max-min fairness problem. In *2nd Polish-German Teletraffic Symposium*, pages 253–258, Gdansk, Poland, 2002.

[118] Survivable fixed telecommunication Network Design library (SNDlib), `http://sndlib.zib.de`.

[119] A Modeling Language for Mathematical Programming (AMPL). `http://www.ampl.com`.

[120] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufman, 2004.

[121] M. Pióro. ATM network design. *Technical report. SWAP Project No. P10352-1 (Swedish National Board for Industrial and Technical Development NUTEK), Lund University*, 1998.

[122] S. De Maesschalck, M. Pickavet, D. Colle, and P. Demeester. On the optimization of the feedback loop for solving the multi-layer traffic grooming problem. In *Proceedings INOC'2003*, pages 195–200, Evry-Paris, 2003.

[123] A. Chabrier, E. Danna, C. Le Pape, and L. Perron. Solving a network design problem. *Technical Report, Number 02-005, ILOG*, 2002.

[124] Claus G. Gruber, Arie M.C.A. Koster, Sebastian Orlowski, Roland Wessäly, and Adrian Zymolka. A new model and computational study for demand-wise shared protection. Technical Report ZIB-Report 05-55, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Berlin-Dahlem, Germany, December 2005.

[125] V. Jain, S. Alagar, S. I. Baig, and S. Venkatesan. Optimal quasi-path restoration in telecom backbone networks. In *13th International Conference on System Engineering*, pages CS–175–CS–180, Las Vegas, USA, August 1999.

[126] M. Patel, R. Chandrasekaran, and S. Venkatesan. Improved quasi-path restoration scheme and spare capacity assignment in mesh networks. Technical Report UTDCS-14-03, University of Texas at Dallas, Richardson, TX, USA, April 2003.

[127] G. Mohan, C. Siva Ram Murthy, and Arun K. Somani. Efficient algorithms for routing dependable connections in WDM optical networks. *IEEE/ACM Transactions on Networking*, 9(5):553–566, October 2001.

[128] Michael T. Fredric and Arun K. Somani. A single-fault recovery strategy for optical networks using subgraph routing. In *7th IFIP Working Conference on Optical Network Design & Modelling (ONDM-2003)*, Budapest, Hungary, February 2003.

[129] Srinivasan Ramasubramanian. On failure dependent protection in optical grooming networks. In *International Conference on Dependable Systems and Networks (DSN'04)*, pages 440–449, Florence, Italy, June 28–July 1 2004.

[130] T. Cinkler, D. Mesko, A. Mitcsenkov, and G. Viola. Adaptive shared protection rearrangement. In *Proc. of 5th International Workshop on Design of*

Reliable Communication Networks, 2005 (DRCN 2005), pages 429–435. IEEE, 2005.

[131] Maulin Patel, R. Chandrasekaran, and S. Venkatesan. A comparative study of restoration schemes and spare capacity assignment in mesh networks. In *The 12th International Conference on Computer Communications and Networks (ICCCN 2003)*, pages 399–404, Dallas, TX, USA, 2003.

[132] Murali S. Kodialam and T. V. Lakshman. Minimum interference routing with applications to MPLS traffic engineering. In *Proc. of INFOCOM (2)*, pages 884–893, 2000.

[133] Advanced CSPF Simulator. `http://suraj.lums.edu.pk/~te/code.htm`.

[134] N. Garg and J. Konemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proc. of 39th Annual Symposium on Foundations of Computer Science*, pages 300–309. IEEE Comput. Soc, 1998.

[135] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1-3):23–47, 2003.

[136] M. Fischetti, F. Glover, and A. Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, 2005.

[137] T. Achterberg and T. Berthold. Improving the feasibility pump. *Discrete Optimization*, 4(1):77–86, 2007.

[138] M. Fischetti and A. Lodi. Repairing MIP infeasibility through local branching. *Computers and Operations Research*, 35(5):1436–1445, 2008.