



HAL
open science

Optimization methods for network design under variable link capacities

Yoann Fouquet

► **To cite this version:**

Yoann Fouquet. Optimization methods for network design under variable link capacities. Other. Université de Technologie de Compiègne, 2015. English. NNT : 2015COMP2233 . tel-01417257

HAL Id: tel-01417257

<https://theses.hal.science/tel-01417257v1>

Submitted on 15 Dec 2016

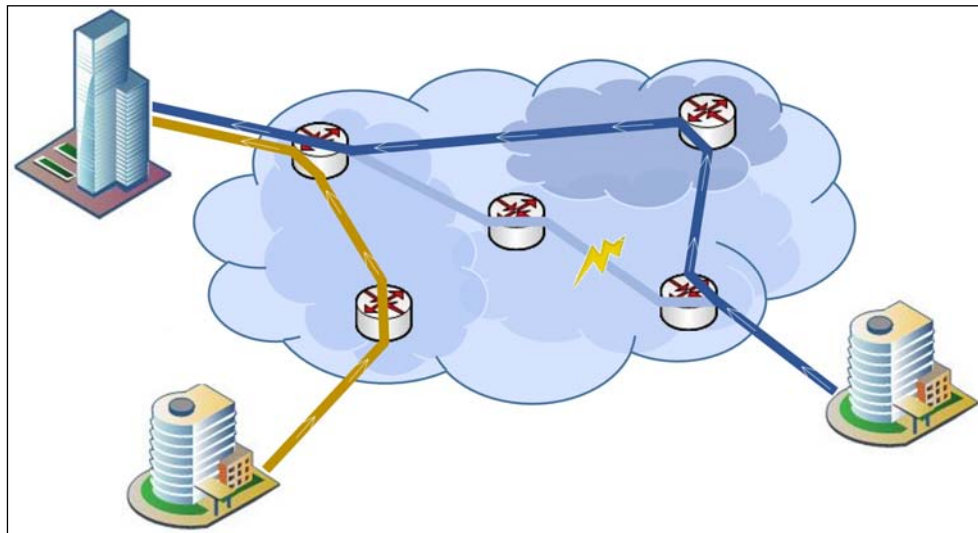
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Par Yoann FOUQUET

Optimization methods for network design under variable link capacities

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenu le 10 Novembre 2015
Spécialité : Information Technologies and Systems

D2233



Sorbonne Universités Université de
Technologie de Compiègne Laboratoire
Heudiasyc UMR CNRS 7253.

PhD Thesis

submitted in fulfillment of the requirements for the degree of
Doctor in Information Technologies and Systems by

Yoann FOUQUET

Optimization methods for network design under variable link capacities

Submitted 10 November 2015

Thesis Committee:

Prof.	Jacques Carlier	Université de Technologie de Compiègne	
Prof.	Bernard Fortz	Université Libre de Bruxelles	(Reviewer)
Dr.	Eric Gourdin	Orange Labs Research	
Prof.	Dritan Nace	Université de Technologie de Compiègne	(Supervisor)
Dr.	Adam Ouorou	Orange Labs Research	(Reviewer)
Dr.	Michael Poss	Université de Montpellier	

To my family

Acknowledgements

This PhD thesis was carried out at the Heudiasyc Laboratory in the University of Technology of Compiègne from November 2012 to October 2015. It was financed with a grant from the French National Education and Research Ministry.

At the end of this work, I would like to address my deepest gratitude to my supervisor Prof. Dritan Nace, for this precious guidance, encouragement and availability during this work to bring it to its completion. His enthusiasm, his continuous support, experience and friendship, have a great influence in my first sets in researches and it will be greatly helpful in my future carrier. I am deeply thankful to my supervisor because he believes in me and care about my future in every minnutes during my thesis.

I would like to express my gratitude to Prof. Michal Piòro and Dr. Michael Poss for their valuable advices and constructive assistance. I'm really grateful for their constant and precious help. I would like to thank all my co-authors, Dr. Ilir Shinko and Dr. Mateusz Zotkiewicz, for their participation in this work, and for assistance.

I would like to thank the members of the jury Dr. Adam Ouorou and Prof. Bernard Fortz to accept taking part of my jury and for their constructive comments and advices.

I would like to thank the Labex MS2T, Control of Technological Systems of Systems, for its financial support that gives me the opportunity to present this research work in several congresses. I really want to thank Laurie Herlin, representing the Labex, for her kindness and her assistance.

I would also like to thank all my friends and colleagues in the University of Technology of Compiègne.

Finally, and most importantly, my warmest thanks go to my wife, Ilda, for her love, her patience and her unconditional and constant support. I am very grateful to my parents for being a constant source of encouragement; to my lovely brother and sisters, Gregory, Sandy and Juna, for their continuous love and infinite support; and to all my family.

Compiègne, July 22, 2015.

Abstract

This thesis summarizes the work we have done in optimization of resilient communication networks. More specifically, the main goal is to propose appropriated recovery mechanisms for managing the demand traffic in a network under partial failures, i.e. when some part of the network (one or some links and/or nodes) is operational with reduced capacity. The main criterion in deciding the efficiency of the proposed recovery scheme is the dimensioning cost of the network while keeping the management cost at reasonable levels. Our main contribution is the design of two restoration strategies named *Flow Thinning* and *Elastic Flow Rerouting*.

This document is organized in three main parts. In the first part, we present the problematic of the thesis. It includes an introduction on the protection and rerouting state-of-art strategies, together with their mathematical models and resolution methods. The second part presents in depth the first protection strategy named *Flow Thinning*. This strategy manages partial failures by decreasing appropriately the bandwidth on some flows routed through one of perturbed links. This implies overdimensionning of the network in the nominal state to ensure demand traffic in all failure states. The third and last part deals with the second rerouting strategy called *Elastic Flow Rerouting*. This strategy is a bit more complex than the first one because, in a failure state, we need to distinguish demands which are disturbed and the one which are not. If a demand is disturbed, it can increase the traffic on some of its paths. If it is not disturbed, it can release bandwidth on paths at the condition it remains non-disturbed. All this allows for further reducing the dimensioning cost but at a higher cost in terms of recovery process management. Note that the dimensioning problems for each strategy are shown to be NP -hard in their general form.

The work of the thesis has been published in: three journal articles (Fouquet et al. (2015b), Pióro et al. (2015), Shinko et al. (2015)), two invited articles (Fouquet and Nace (2015), Fouquet et al. (2014c)) and height articles in international conferences (Fouquet et al. (2015a; 2014d;a;b;e), Pióro et al. (2013b;a), Shinko et al. (2013)). Note that Pióro et al. (2013b) has been rewarded by a "Best Paper Award" from the RNDM conference.

To conclude, note that this thesis was realized in the Heudiasyc laboratory, from the Université de Technologie de Compiègne (UTC). It was financed by the French Ministry of Higher Education and Research¹ with the support of the Labex MS2T² of the UTC.

¹<http://www.enseignementsup-recherche.gouv.fr/>

²<https://www.hds.utc.fr/labex-ms2t-484/>

Résumé

Cette thèse porte sur l'optimisation des stratégies de reroutage dans les réseaux de télécommunications. Plus précisément, l'objectif est de proposer ou d'adapter des mécanismes permettant de router le trafic du réseau après une panne partielle, c'est-à-dire, après une baisse de la bande passante d'un ou plusieurs liens du réseau, tout en minimisant le coût de dimensionnement du réseau. Nos contributions principales sont la proposition de deux stratégies de protection/routage nommée *Flow Thinning* et *Elastic Flow Rerouting*.

La thèse est organisée en trois parties. Dans la première partie, nous présentons la problématique de la thèse avant de passer en revue les stratégies de protection et reroutage de la littérature, leur modélisation et méthode de résolution. La deuxième partie présente en détails la première stratégie de protection appelée *Flow-Thinning*. Cette stratégie gère les pannes partielles en diminuant la bande passante de certains flots qui passent par le ou les arc(s) perturbés. Cela implique un surdimensionnement du routage nominal permettant d'assurer le trafic en cas de perturbations. La troisième et dernière partie concerne la deuxième stratégie de routage dénommée *Elastic Flow Rerouting*. Cette stratégie est un peu plus complexe que la première dans le sens où, en cas de panne, une distinction est faite entre les demandes perturbées ou non. Si une demande est perturbée, elle peut augmenter le trafic sur ces chemins. Si elle ne l'est pas, elle peut libérer de la bande passante sous la condition qu'elle ne devienne pas perturbée à son tour. Notons que ces deux stratégies sont assez difficiles du point de vue de leur complexité.

Cette thèse a fait l'objet de divers travaux écrits: trois articles (acceptés ou en révision) dans des journaux (Fouquet et al. (2015b), Pióro et al. (2015), Shinko et al. (2015)), deux articles invités (Fouquet and Nace (2015), Fouquet et al. (2014c)) et huit articles dans des conférences internationales (Fouquet et al. (2015a; 2014d;a;b;e), Pióro et al. (2013b;a), Shinko et al. (2013)). Notons que Pióro et al. (2013b) a reçu le "Best Paper Award" de la conférence RNDM 2013.

Pour finir, notons que cette thèse a été réalisée au laboratoire Heudiasyc de l'Université de Technologie de Compiègne (UTC). Elle a été financée par le Ministère de l'enseignement et de la recherche français³ avec le soutien du labex MS2T⁴ de l'UTC.

³<http://www.enseignementsup-recherche.gouv.fr/>

⁴<https://www.hds.utc.fr/labex-ms2t-484/>

Contents

Contents	i
I Optimization methods for resilient networks	1
1 Introduction	3
1.1 Context and Motivation	3
1.1.1 Wireless communications	3
1.1.2 Wired networks	5
1.2 Our contribution	6
1.3 Outline of this thesis	7
2 Background and Literature Review	11
2.1 Introduction to Recovery Mechanisms	11
2.2 Strategies for total link failures	14
2.2.1 Protection Strategies	14
2.2.2 Restoration Strategies	15
2.2.3 MPLS feature - MPLS Fast Rerouting	19
2.3 Strategies for partial link failures	19
2.3.1 Probabilistic Routing - PR	19
2.3.2 Elastic Traffic Flows - ETF	20
2.3.3 Global Rerouting - GR	20
2.3.4 MPLS feature - MPLS TE Auto-bandwidth Adjustment	21
3 Network Dimensioning Problem	23
3.1 Notation	23
3.2 Study of the nominal state	25
3.2.1 Dual problem and dual separation	26
3.2.2 Path generation	27
3.3 Study of Global Rerouting	27
3.3.1 Dual problem and dual separation	29
3.3.2 Path generation	29
3.4 Mathematical formulation of the restoration methods	30
3.4.1 End-to-end rerouting with stub release - RR^+	30
3.4.2 End-to-end rerouting without stub release - RR^-	32
3.4.3 Local Rerouting - LR	32
3.4.4 Path Diversity - PD	33
3.4.5 Shared Robust Rerouting - ShRR	34
3.5 Numerical results for all mechanisms	35
3.5.1 Dimensioning cost	35
3.5.2 Efficiency of Path Generation	35

II	Flow Thinning	39
4	Flow Thinning - Introduction and Optimization Problem	41
4.1	Introduction to Flow Thinning	42
4.2	FTOP—the basic FT optimization problem	44
4.2.1	Notations	44
4.2.2	Formulation of FTOP	46
4.2.3	Complexity of FTOP and global rerouting	47
4.3	Dual problem and dual separation	48
4.3.1	Dual problem formulation	48
4.3.2	Dual separation	49
4.4	Path generation	51
4.4.1	Pricing problem	52
4.4.2	Binary formulation of PP	53
4.4.3	PG algorithm	54
4.5	Computational complexity of PP, DSP, and special cases	55
4.6	Single-link failures	57
4.6.1	Single-link failures – compact FTOP formulation	57
4.6.2	Single-link failures - proof of Lemma 3	59
4.7	Single-node failures	60
4.7.1	Single-node failures – the pricing problem	60
4.7.2	Single-node failures – compact FTOP formulation	63
5	Efficiency of Flow Thinning Optimization	67
5.1	Cost efficiency of flow thinning	68
5.1.1	Single partial link failure scenarios	69
5.1.2	Single partial node failure scenarios	70
5.1.3	Double partial failure scenarios	72
5.2	Computational efficiency of flow thinning optimization	73
5.2.1	Path generation	73
5.2.2	Evolution of the PG iterations	74
5.2.3	Running times	76
5.3	A lower bound formulation	77
5.3.1	LBP – Lower Bound Problem	77
5.3.2	Comparing LBP and FTOP	77
6	FTS - Applications, implementation and practicability	81
6.1	Affine versions of FTS	81
6.1.1	Affine flow thinning and the related version of FTOP	82
6.1.2	Restricted affine flow thinning	83
6.2	Implementation issues	84
6.2.1	Implementation issues of FT	84
6.2.2	Practicability of FTOP and its affine version	85
6.3	Thickening versions of FTS	87
6.3.1	Flow Adjustment – The flow thickening extension of FTOP	87
6.3.2	Performance of Flow Adjustment	88
6.3.3	Why Flow Adjustment should be avoided	89
6.4	Concluding remarks	92

III Elastic Flow Rerouting	95
7 Elastic Flow Rerouting for total link failures	97
7.1 Introduction to Elastic Flow Rerouting	97
7.2 Mathematical model of EFR	100
7.2.1 Notation	100
7.2.2 EFR-P formulation	101
7.3 Complexity discussion	102
7.4 Heuristic method	105
7.4.1 Description of the heuristic	106
7.4.2 Formulation of EFR-PF problem	107
7.4.3 Dual problem	108
7.4.4 Pricing problem	109
7.4.5 Binary formulation of the pricing problem	110
8 EFR — Practical applications to Multiple Partial Link Failures	113
8.1 Application to Fixed Wireless Networks	113
8.1.1 Introduction to EFR _{PLF} and Notations	114
8.1.2 Mathematical formulation of EFR _{PLF} -P	115
8.1.3 Modulation Schemes management	116
8.2 Towards a distributed version of EFR	118
9 EFR - Efficiency of the strategy	121
9.1 Dimensioning cost	121
9.2 Heuristic performance	123
9.3 Efficiency of EFR applied to partial failures	123
9.4 Restoration time	125
9.5 Conclusion on the efficiency of EFR.	126
Conclusion	127
A Appendices	129
A.1 MPLS Networks	130
A.1.1 Label Switching	130
A.1.2 Routing Principle	131
A.2 EFR - Implementation issues	132
A.3 SNDlib - Graph details	134
Bibliography	135
List of Publications	141
List of Abbreviations	143
List of Figures	144

Part I

Optimization methods for resilient networks

The first part includes an introduction chapter, together with a literature review of state-of-the-art recovery mechanisms. In Chapter 1, we present the context of our work, our contribution, and the outlines of the thesis. Chapter 2 is dedicated to a presentation of existing recovery mechanisms such as *Global Rerouting*, *Local Rerouting*, *Path Diversity*, etc. Note that a focus will be put on the mechanisms used in MPLS networks. This review of the literature is followed by an introduction to the network dimensioning problem in Chapter 3. The mathematical formulation of the dimensioning problem under these strategies is done using continuous linear programming (LP) or mixed integer linear programming (MILP). Some of these strategies lead to *NP*-hard network dimensioning problems (end-to-end rerouting with stub release for instance). Furthermore, compact arc-node formulations are not always possible or not convenient to handle realistic instances and the preferred models are those based on path-flow variables. Hence, we use Path Generation (PG) to avoid the use of the full set of paths and to reduce the size of the problems. This chapter is concluded by a numerical study of the efficiency of the strategies.

Introduction

1

Contents

1.1 Context and Motivation	3
1.1.1 Wireless communications	3
1.1.2 Wired networks	5
1.2 Our contribution	6
1.3 Outline of this thesis	7

The work done in this thesis has taken place in the Heudiasyc laboratory, in the Université de Technologie de Compiègne. It was financed by the French Ministry of Higher Education and Research with the support of the Labex MS2T of the UTC. This thesis focuses on network dimensioning problems, and more specifically on the design of recovery mechanisms to manage the traffic in case of failures.

1.1 Context and Motivation

The thesis addresses resilience of communication networks in both wired and wireless context. These networks are subject to total failures when link capacities are totally lost, and partial failures when the capacities are only partially reduced. In the following, we introduce the failure management aspect in both networks, starting with wireless networks.

1.1.1 Wireless communications

Fixed broadband wireless communications sector holds great promise for providing private high-speed data connections by means of microwave radio (Anderson 2003, Lehpamer 2010) or free space optical (FSO) transmission (Willebrand and Ghuman 2002, Son and Mao 2010). Both microwave and optical transmission refer to terrestrial point-to-point digital communications, usually employing highly directional antennas within the line-of-sight (LOS). What is important, the two technologies operate at unlicensed frequency bands, contrary to radio communications based on the Wi-Fi IEEE 802.11-family standards. Moreover, both microwave and

FSO are sensitive to interference—the feature we utilize in our optimization model. Despite recent advances in optimization of fixed broadband wireless networks, a variety of questions remain unaddressed in this area. Particularly, capacity planning in fixed wireless networks is quite different from that in wired networks. In fact, environment conditions, especially weather, play an important role since they can introduce instantaneous variations into the communication channel, likely leading to outage events which can be modeled as multiple partial link failures. As a common practice, operators tend to highly over-provision bandwidth during network planning to avoid traffic bottlenecks under adverse scenarios (when the performance of some links deteriorates). This approach, however, incurs additional costs that do not result in resource- and cost-efficient networks, leading to an inefficient use of the radio spectrum.

Even though FSO and microwave radio have a lot in common, this introduction will focus on microwave radio communications. Radio frequency spectrum is a limited resource regulated worldwide by the International Telecommunications Union (ITU). In conjunction with the ITU regulations, national legislation instruments establish the availability of frequency bands for specific applications and the procedures to use licenses. A license (assignment) is an authorization given by administration for a radio station to use a radio frequency under specified conditions. In most cases, the price of frequency spectrum for a single microwave link is a function of the amount of spectrum (bandwidth) in MHz with which the license is associated.

Commonly, to support broadband applications, modern microwave systems use quadrature amplitude modulation (QAM). Note that QAM

An m -QAM scheme presents m combinations of amplitude and phase, each one representing an n -bit pattern called a symbol (with $n = \log_2 m$ and integer). Given the channel bandwidth B and the m -QAM scheme in use, we can approximate the channel capacity C by:

$$C [Mb/s] = nB [MHz]. \quad (1.1)$$

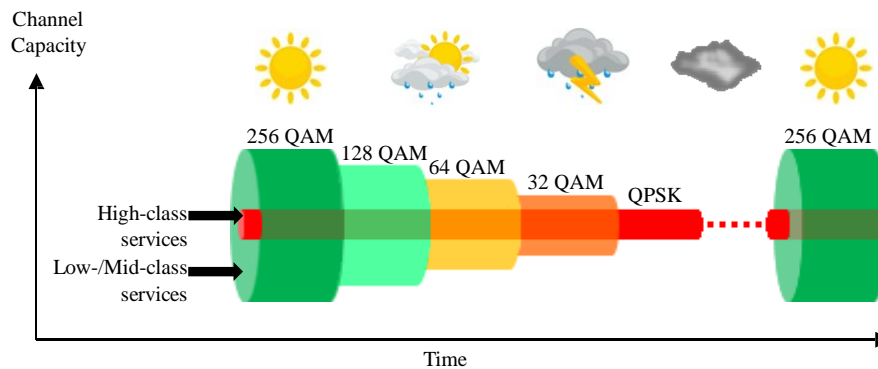
The m -QAM schemes with high values of m assure bandwidth efficiency, but are more susceptible to errors due to channel impairments. As the modulation scheme changes to accommodate higher data rates, the signal-to-noise ratio (SNR) requirement increases to preserve the bit error rate (BER). Moreover, the transmitted signal suffers deep fades and hence microwave links are susceptible to outage events. To overcome this issue, modern microwave systems employ adaptive modulation and coding which has been proven to considerably enhance link performance (Goldsmith and Chua 1997). To keep the BER performance, this technique entails the variability of the link capacity.

In the thesis, we will exploit the capability of the above considered networks to adapt the capacity of links to meet the current SNR requirement. We assume that a list of states, typically corresponding to degraded weather condition in a particular area, is given. Each state is represented by a vector of link capacities where a set of links (corresponding to the degraded area) have their capacity decreased due to the modulation scheme applied to cope with the weather condition in the area, while the others remain unchanged. Note that adverse weather conditions most affect low-

Table 1.1 – *Bandwidth efficiency, SNR requirement and capacity.*

Modulation scheme	Bandwidth efficiency	SNR requirement	Capacity for 7MHz	Capacity for 28 MHz
QPSK	2 bps/Hz	14.21 dB	14 Mbps	56 Mbps
16-QAM	4 bps/Hz	21.02 dB	28 Mbps	112 Mbps
32-QAM	5 bps/Hz	25.24 dB	35 Mbps	140 Mbps
64-QAM	6 bps/Hz	27.45 dB	42 Mbps	168 Mbps
128-QAM	7 bps/Hz	31.10 dB	49 Mbps	196 Mbps
256-QAM	8 bps/Hz	33.78 dB	56 Mbps	224 Mbps

class and mid-class services. However, extreme environmental conditions can lead to total failures as the lowest SNR requirements cannot be met, and high-class services will be affected.

Figure 1.1 – *Adverse weather conditions decrease wireless channel capacity.*

Note that the above considerations hold also for FSO networks. Factors such as beam dispersion, atmospheric absorption, rain, fog, snow, and scintillation, among other factors, can lead to capacity limitations (Xiaoming and Kahn 2002, Puryear 2011). Furthermore, QAM modulation can be applied (Harjeevan and Rajan 2013).

1.1.2 Wired networks

Broadband cable communications are in place for decades. Nowadays, these transmissions are done through cable modem or optical fibers (Berthold et al. 2008). Failures in high-speed networks have always been a concern of high importance. The specificity of these networks is the fact that a link failure corresponds to the total failure of a link (i.e. no capacity remains on the link). These failures, typically link cuts or router damages, can lead to heavy losses of traffic. The management of failures in wired networks has led to a large set of mechanisms, ensuring traffic requirements after a failure if possible. A brief listing of existing mechanisms is presented in Chapter 2.

Note that contrary to wireless networks, capacities in wired networks are fixed in advance. Indeed, installing new fiber, with the right-of-way and installation costs, is very expensive and there is no way to increase capacities by increasing transmission power.

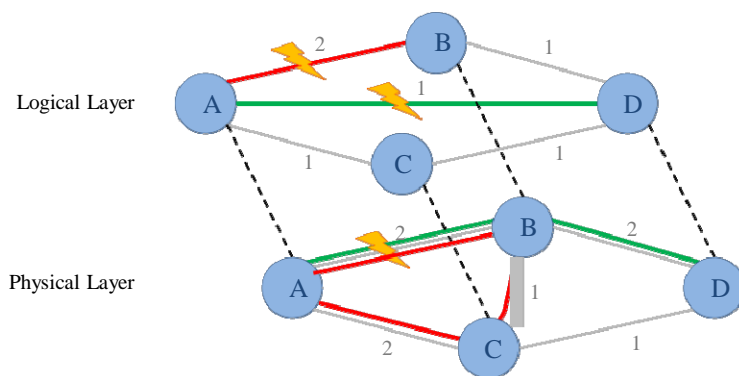


Figure 1.2 – *Partial failures in logical layers of wired networks.*

Survivability can be addressed in many layers in a network. Even though partial failures seem impossible in the physical layer of wired networks, partial link failures can appear in the upper layer of these wired networks. Figure 1.2 illustrates this phenomenon. First note that a network can be represented by a graph where routers are seen as nodes, and links as edges. In this network, link (A, B) in the logical layer is physically routed through paths $A - B$ and $A - C - B$. Similarly, link (A, D) in the upper layer is routed through path $A - B - D$ in the physical layer. Hence, when the physical link (A, B) is cut, link (A, D) in the upper layer totally fails but link (A, B) suffers a degradation of only 50% of its capacity as physical path $A - C - B$ is not affected by the failure.

1.2 Our contribution

In this thesis, we investigate network optimization problems related to the design and configuration of networks which can suffer partial link failures. The study is pursued also for total failures. We are concerned with a general class of problems expressed in terms of minimum cost multicommodity flow (MCMCF) problems, which are largely used for optimal design and dimensioning of telecommunication networks (Kennington 1978, Minoux 2006). These problems basically consist of transporting different commodities, from their respective sources to their destinations, which simultaneously use the network and are coupled through either by links' capacities or the cost function to be minimized. Obviously, there should be enough capacity in the network to simultaneously carry all traffic requirements. Various special cases of the MCMCF problem are reported in (Minoux 2006, Fortz et al. 2013, Botton et al. 2011), each of them associated with an appropriate choice of link cost function. Generally, the optimization criterion refers to the total cost of the equipment to be installed on various links of the network. When the cost function is considered linear, then the MCMCF problem can be formulated as a large scale continuous linear program (LP), and many efficient algorithms are available to tackle it. These mathematical models are the basis of the mathematical formulation in our work.

In telecommunication networks, the idea of failure is largely limited to scenarios in which every link can fail but one at a time. A failure scenario consists on the total failure of one link, i.e., the temporary lost of a

link. This assumption may be considered valid for large scale wired networks using for instance optical fiber. In this thesis, we consider a link as perturbed if its modulation scheme has changed and so its capacity decreased. Despite a large amount of strategies available to recover from total failures (global rerouting, N:M protection (Mannie and Papadimitriou 2006), path diversity (Pióro and Medhi 2004a), ...), only a few of them seem to be able to manage partial failures.

Our contribution can be summarized as follows. First, we have studied state-of-the-art recovery mechanisms. These mechanisms have been studied in deep, their mathematical model presented, solved and compared. Our objective is to take the advantages of the existing strategies while limiting their drawbacks.

Hence, we have designed a protection strategies denoted *Flow Thinning* (FT), which handles partial link failures. We formulated the mathematical LP model for network design problem under such recovery strategy. FT is shown to be *NP*-hard. We chose to use decomposition algorithms to solve this problem. The column generation (CG) algorithm differs from usual CG algorithm in the sense that it requires solving Mixed Integer Linear Program (MILP) using binaries. In addition, a complexity study has been made and special cases were presented and solved using arc-node formulations. This work has been done in collaboration with my supervisor Prof. Dritan Nace with Prof. Michal Pióro from Warsaw University of Technology (Poland) and Lund University (Sweden) and Dr. Michael Poss from the Université de Montpellier (France).

Moreover, the work on FT was done in parallel with the design of a second strategy named *Elastic Flow Rerouting* (EFR). We studied the mathematical modelisation of this strategy, still using CG algorithms. Contrary to FT, EFR required branch-and-bound algorithm coupled with CG. Hence, we had to develop a heuristic to obtain near-optimal solutions in a limited period of time. This work was done in collaboration with Dr. Mateusz Żotkiewicz from Warsaw University of Technology and Dr. Ilir Shinko from Polytechnic University of Tirana (Albania) in addition to the group working on FT.

Finally, we have addressed the practicability of the two strategies and the implementation issues.

1.3 Outline of this thesis

This thesis is composed of three main parts.

The first part is divided into three chapters and includes this introduction chapter, together with a literature review of state-of-the-art recovery mechanisms. Recovery mechanisms can be categorized in protection or restoration mechanisms. We present three protection strategies, denoted 1+1 Protection Mannie and Papadimitriou (2006), N:M Protection Mannie and Papadimitriou (2006) and Path Diversity (Pióro and Medhi 2004a). We also present several restoration strategies. In those strategies, the rerouting can be local, i.e. around the failure (Local Rerouting), or end-to-end from the source to the destination of the demand (Global Rerouting, end-to-end rerouting with and without stub release). We also introduce Shared Robust Rerouting (Fundo et al. 2013), a hybrid mechanism in which some

non-affected demands can release capacities in a failure state. We analyze through brief explanations and examples the idea of each strategy. All these strategies, but Global Rerouting, can only manage total link failures. Thus, we present some mechanisms handling partial link failures as Elastic Traffic Flows (Tomaszewski 2014) or Probabilistic Routing (Claßen et al. 2011). A focus will be put on the mechanisms used in MPLS networks. Hence, we introduce some MPLS specific mechanisms such as MPLS Fast Reroute (Pan et al. 2005) and MPLS-TE Auto-bandwidth adjustment (Dhody et al. 2015). This review of the literature is followed by an introduction to the network dimensioning problem. The mathematical formulation of most strategies is presented using continuous linear program (LP) or mixed integer linear program (MILP). Some of these strategies lead to NP -hard network dimensioning problems (end-to-end rerouting for instance with stub release). Furthermore, compact arc-node formulations are not always possible or not convenient to handle realistic instances and the preferred models are those based on path-flow variables. Hence, we use Path Generation (PG) to avoid the use of the full set of paths and to reduce the size of the problems. This part is concluded by a numerical study of the efficiency of the strategies.

The second part, composed of Chapters 4, 5 and 6 is dedicated to the *Flow-Thinning* strategy (FTS) (Pióro et al. 2013b). The main idea behind FTS is as follows. Following the idea of path diversity, we assume that each demand is in general routed over several paths, not necessarily disjoint, with over-dimensioned path-flows to ensure an assumed level of survivability. Unlike conventional end-to-end restoration strategies, FTS guarantees survivability solely by releasing a fraction of flow on the affected routing paths. Consequently, no flow is rerouted or increased. It is important that this thinning of path-flows is done in an appropriate failure-state dependent way. Chapter 4 introduce the strategy through examples. Then, we present the basic FT optimization problem (FTOP) followed by a discussion on its complexity. FTOP is NP -hard at least for two-links failure scenarios. Thus, we study the separation problem and propose an original Path Generation algorithm using a binary LP. Next, in Chapter 5, we study the efficiency of FTS. We first compare the dimensioning cost of the network using FTS with other strategies presented in Chapter 2. This study is followed by a comparison with a lower bound formulation of the problem. In the last chapter (Chapter 6), we present an affine version of FTS that reduces the number of failure states for which the strategy needs to be optimized (propose in (Pióro et al. 2015)). Next, we deal with implementation issues of FTS and its affine variant. Finally, we present an extension of FTS that allows for thickening and we explain the reasons why this was not considered in the final version of FTS.

The third and last part is composed of Chapters 7, 8 and 9. This part is dedicated to the *Elastic Flow Rerouting* (EFR) strategy (Fouquet et al. 2014e) which works as follows. In the nominal state (when all links are operational), traffic is routed over a given set of paths for each traffic demand. In each failure state, EFR uses the same set of (nominal) routing paths but with possibly modified path-flow values. The main idea behind EFR is to restore traffic of the affected demands by means of increasing the flow on their routing paths, possibly together with decreasing flows of the unaf-

affected demands. Notice that no new paths are allowed to be established. Summarizing, flows (on existing paths) can be increased due to capacity released on affected paths, capacity possibly released on some paths of unaffected demands or protection capacity present on links due to network over-dimensioning. In Chapter 7, we introduce the mathematical formulation for total failure scenarios (EFR-P) together with a complexity study. EFR-P is NP -hard and requires branch-and-price to be solved to optimality due to the distinction between affected and unaffected demands. Thus, we propose a greedy heuristic method to solve EFR. Note that once again, EFR-P is solved using Path Generation with a pricing problem requiring a binary problem. As this thesis deals with both total and partial failures, in chapter 8, we study practical applications of EFR to partial failures. Following the idea of microwave communications (presented in 1.1.1), we introduce channels and modulation schemes in our formulation and show how EFR can be combined with FTS to give a strategy capable to handle both type of failures. Finally, we study a distributed variant of EFR. In the last chapter of this part, Chapter 9, we study the efficiency of EFR and its formulations followed by an analysis of the heuristic method performance. We conclude this part with a study of restoration time in case of failure of both centralized and distributed versions of EFR.

In the conclusion, we summarize the work done in the thesis. We analyze our contribution to the literature about rerouting strategies and network design. Finally, we present possible future works on these strategies.

Background and Literature Review

2

Contents

2.1 Introduction to Recovery Mechanisms	11
2.2 Strategies for total link failures	14
2.2.1 Protection Strategies	14
2.2.2 Restoration Strategies	15
2.2.3 MPLS feature - MPLS Fast Rerouting	19
2.3 Strategies for partial link failures	19
2.3.1 Probabilistic Routing - PR	19
2.3.2 Elastic Traffic Flows - ETF	20
2.3.3 Global Rerouting - GR	20
2.3.4 MPLS feature - MPLS TE Auto-bandwidth Adjustment	21

This chapter is dedicated to a general presentation of recovery mechanisms for communication networks. We have chosen MPLS networks (see appendix A.1) to illustrate the mechanisms. Obviously, these methods can be applied to other transport protocols, especially to tunneling protocols. The first section will briefly introduce the strategies. In the next section, we consider total link failures and study the most well-known recovery mechanisms divided in protection and rerouting strategies. In the third section, we analyze strategies of the literature for partial link failures. Finally, the last section will sum up the advantages and drawbacks of each strategies.

2.1 Introduction to Recovery Mechanisms

Resiliency in telecommunication networks is currently an important issue and a hot research topic in network optimization. The main problem remains the design of efficient recovery techniques in terms of management effort and cost-effectiveness. Generally speaking, in a network we distinguish between the nominal state (where all elements are operational) and failure states (where one or more elements have ceased to function). Each

failure state is identified by the subset of the failed elements. The restoration procedure then works as follows: given a fixed routing in the nominal state, in cases of failure backup **Label Switch Paths** (LSPs) are used to restore the disrupted traffic following a precise recovery process. Before discussing in details the mechanisms that are the focus of this chapter let us recall briefly the main existing recovery techniques. Roughly speaking, recovery techniques can be separated into protection strategies and restoration strategies. Figure 2.1 presents a classification of the recovery mechanisms presented in this chapter. In this figure, strategies able to manage total link failures are represented with a **T** while strategies that can handle partial failures are indicated with a **P**. Note that GR can manage both type of failure scenarios.

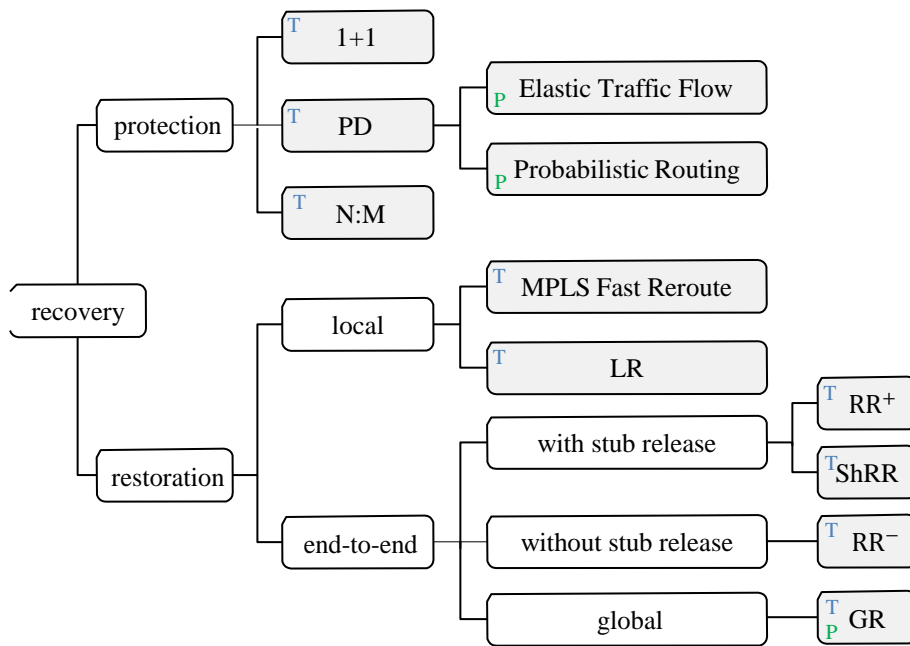


Figure 2.1 – Classification of recovery mechanisms from the literature.

Let us recall some basic notions. A traffic demand is represented by a pair of nodes and a flow metric that needs to be satisfied between these nodes. The traffic demands are summarized in a demand matrix. The routing in the network is the flow which satisfies a given traffic demand, i.e. a set of LSPs between the two extremities of all demands with a non-negative flow value. These LSPs are called nominal routing paths. As stated above, we distinguish between the **nominal state** (NS) where the entire network is operational and the **failure state** (FS) where at least one link is not operational. In this chapter, we will focus on cases where only one link can fail at any given time. We consider here total link failures, where the failed element is assumed to have totally ceased to function, and partial link failures, where only a part of the link capacity is lost. Rerouting in cases of link failure involves reorganizing the routing of some demands, which may or may not have been disrupted by the failure, in order to restore the lost traffic. The lost traffic is thus (re)routed through some backup LSPs or restoration paths. Protection schemes use preplanned recovery paths that are fully defined before the failure occurs. Protection

schemes can be divided into three main groups: *Protection 1 + 1*, *Protection N : M*, and *Path Diversity* (PD). Rerouting is a way of restoring traffic that is an alternative to protection schemes. It ensures restoration using the capacity available in the network to route the disrupted traffic. The recovery level can be very high (up to 100%). At the same time, the residual capacities can be shared between different backup paths, which makes this strategy more cost-effective than protection strategies. The main drawback of rerouting is that when failures occur the recovery time and the number of paths used are significantly higher compared to protection strategies. The best-known rerouting strategies are *local rerouting* (LR) and end-to-end rerouting, depending on whether the traffic is rerouted between the extremities of the failed link or the extremities of the disrupted demand. Hence, end-to-end rerouting handles failures by rerouting the traffic from the source to the destination nodes on alternative LSPs. When a failure occurs in a network and a set of demands is disrupted, the following end-to-end rerouting strategies may be used:

- *Global Rerouting* (GR) if nominal paths are removed after the failure and a set of paths is created to route the traffic,
- *End-to-end Rerouting without stub release* (RR^-) if only disturbed traffic is rerouted and restoration is done using available capacities in the network,
- *End-to-end Rerouting with stub release, or Restricted Restoration* (RR^+) when restoration is done using available capacities in the network and capacities released by disturbed paths.

Some other strategies will be presented as they are an extension of the previously described strategy (ShRR extends RR^+), or specific MPLS mechanisms (MPLS Fast Reroute and Auto-Bandwidth Adjustment).

Example 2.1 *In this section, every strategy will be illustrated using the same example. Let*

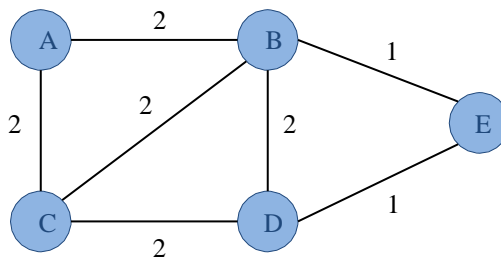


Figure 2.2 – The network.

consider a network with 5 nodes A, B, C, D and E and 7 links (A,B), (A,C), (B,C), (B,D), (B,E), (C,D), (D,E). All links have the same unit cost $c_e = 1$ and a capacity of 1 to 2 unit(s). We also consider two demands (A,E) and (A,D), both requiring 1 unit of traffic.

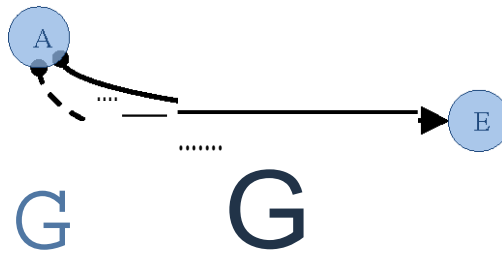


Figure 2.3- The demands.

2.2 STRATEGIES FOR TOTAL LINK FAILURES

In this section, we present mechanisms that are designed to manage total link failures. First, protection strategies are presented, followed by restoration strategies. Finally, MPLS Fast Reroute, a dedicated MPLS mechanism will be presented.

2.2.:1 Protection Strategies

2.2.:1.:1 Protection 1+1

1+1 Protection (Mannie and Papadimitriou 2006) is composed of one working LSP, one protection LSP, and a permanent bridge. At the ingress node, the normal traffic is permanently bridged to both the working and protection LSP. At the egress node, the normal traffic is selected from the better of the two LSPs. Due to the permanent bridging, the 1+1 protection does not allow an unprotected extra traffic signal to be provided.

Figure 2-4 illustrates the behavior of 1+1 Protection. In the nominal state, two LSPs are set from node A to node E with the same capacity of 1 unit of traffic. However, only path {A-B-E} carry out the traffic of the demand. Once link (B,E) fails, the traffic is switched to path {A-C-D-E}.

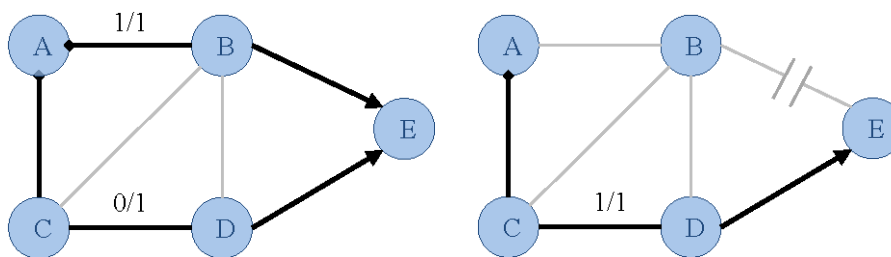


Figure 2.4- 1+1 Protection before (left) and after (right) failure of link (B,E).

2.2.:1.2 Protection N:M

N:M protection (Mannie and Papadimitriou 2006) has N working LSPs carrying normal traffic and M protection LSP that may carry extra-traffic. At the ingress, the normal traffic is permanently connected to its N working LSPs and to one of its M protection LSPs. At the egress node, the normal traffic is selected from either its working or one of the protection

LSP. Unprotected extra traffic can be transported over the M protection LSP whenever the protection LSPs is not used to carry a normal traffic.

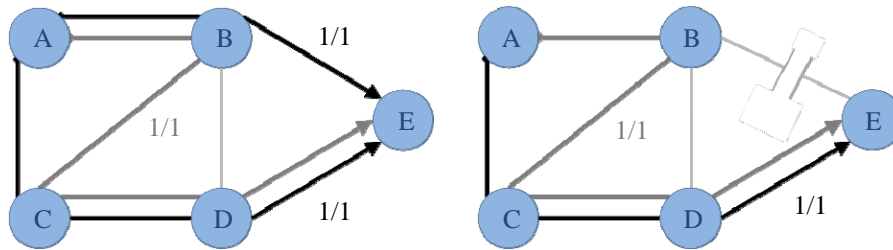


Figure 2.5 – $N:M$ Protection before (left) and after (right) failure of link (B,E) .

Figure 2.5 illustrates the behavior of $N:M$ Protection with $N = 2$ and $M = 1$. In the nominal state, three LSPs are set from node A to node E with the same capacity of 1 unit of traffic. Paths drawn in black $\{A-B-E\}$ and $\{A-C-D-E\}$ are working LSPs, while the path drawn in dark gray $\{A-B-C-D-E\}$ is a protection LSP and the egress node E selects the traffic from one of these two working paths. Once link (B,E) fails, if the traffic was taken from path $\{A-B-E\}$ then E must change its selected path to $\{A-C-D-E\}$ or $\{A-B-C-D-E\}$. However, if the traffic was taken from another path, nothing is changed.

2.2.1.3 Path Diversity - PD

Path Diversity (Pióro and Medhi 2004a), or Demand-wise Shared Protection (Koster et al. 2005), is a protection scheme in which the traffic of a demand is routed in several paths not necessarily vertex-disjoint. They are "dedicated" to this demand and a part of the reserved bandwidth is intended for backup. The only requirement is that, whatever the failure link, the remaining capacity on the non-disrupted paths has to be sufficient to route all the traffic corresponding to the demand. Note that the complete establishment of the restoration LSP (i.e., activation) occurs only after failure detection and/or notification of the working LSPs and requires some additional restoration signaling. Therefore, this mechanism protects against working LSPs failure(s) but requires activation of the restoration LSPs after failure occurrence. After the ingress node has activated the restoration LSPs, the latter can carry the normal traffic. Notice that when each working LSP is recoverable by exactly one restoration LSP, one refers also to 1+1 Protection.

In Figure 2.6, the demand is routed from node A to node E . Paths $\{A-B-E\}$ and $\{A-B-D-E\}$ are working LSPs, while paths $\{A-C-D-E\}$ and $\{A-C-B-E\}$ are protection LSPs. When link (B,E) fails, 0.5 unit of bandwidth is missing to fully route the traffic of demand (A,E) . Hence, protection path $\{A-C-D-E\}$ is activated and the two paths can route all the traffic.

2.2.2 Restoration Strategies

Rerouting is another way to restore the traffic. It ensures restoration using the unused capacity available in the network to route the disturbed traffic. The recovery level can be very high (until 100%). In the same time, the

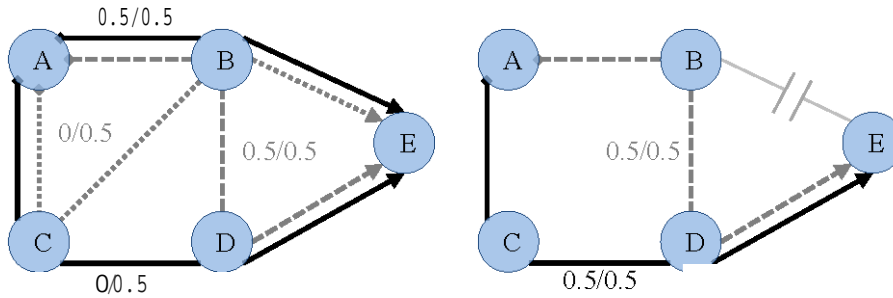


Figure 2.6- PD before (left) and after (right) failure of link (B ,E).

residual capacities can be shared between different backup paths, which makes this strategy more cost efficient than protection strategies. The main drawback of this rerouting strategy is that the recovery time in case of failure and the number of paths used for every demand is really higher than ones obtained by protections strategies.

2.2.2.1. Local Rerouting- LR

When a link failure occurs, the rerouting is done between the extremities of the broken link. In other words, a new demand is created between the extremities of the broken link with a value equal to the nominal flow on this link. Concerning non-disturbed paths, no modification will be done.

In Figure 2.7, we observe that when link (B,E) fails, the primary LSP of demand (A,E) remains the same from node A to node B. The perturbed traffic is rerouted on path {B-D-E}, from the upstream node of the failed link to its downstream node, which in our example is the egress node of the demand.

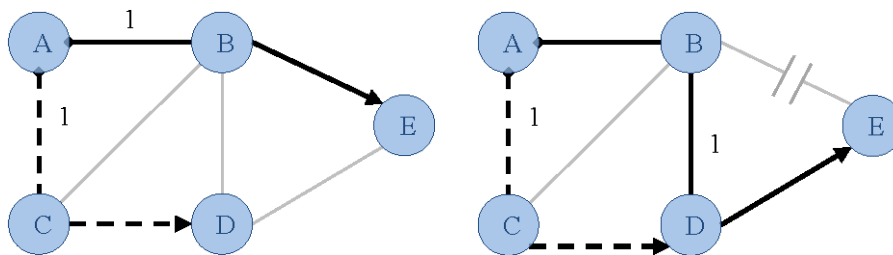


Figure 2.7- Local Rerouting before (left) and after (right) failure of link (B ,E).

2.2.2.2 Global Rerouting - GR

This strategy allows us to reroute all the demands, even those which are not perturbed by the failure. This strategy is highly complex to manage and almost not conceivable for large networks. However, this strategy is the most cost efficient and is used as a lower bound for the other strategies.

Figure 2.8 illustrates the GR strategy. In the nominal state, a LSP {A-B-E} is set for demand (A,E) and another LSP {A-C-D} is set for demand (A,D). When link (B,E) fails, all LSPs are removed. New LSPs are created

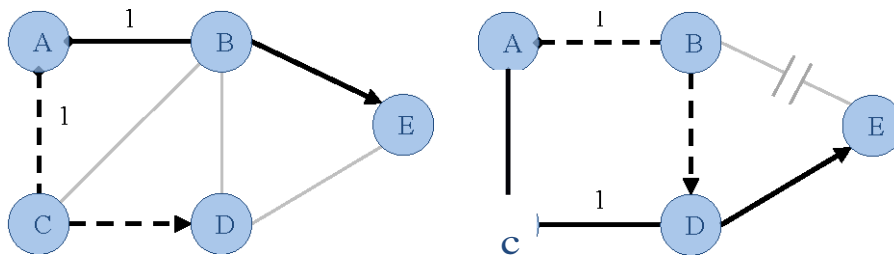


Figure 2.8- Global Rerouting before (left) and after (right) failure of link (B,E).

to route the traffic using path $\{A-C-D-E\}$ for demand (A,E) and $\{A-B-D\}$ for demand (A,D).

2.2.2.3 End-to-end rerouting with stub release- RR+

This strategy can reroute the disturbed paths using the capacities released by the other disturbed paths. When a link fails, the traffic of the demands using this link in the nominal state is broken and rerouted on other paths. The capacities used by disturbed routing paths over remaining operational links, are released and can be used for rerouting.

In the following example, Figure 2.9, demand (A,E) is routed through path $\{A-B-E\}$. When link (B,E) fails, this path is perturbed and the traffic can be rerouted to another LSP, here path $\{A-B-D-E\}$ using capacities available on links (B,D) and (D,E) and those released by disturbed path $\{A-B-E\}$ and link (AB). Note that, contrary to GR (Figure 2.8), path $\{A-C-D\}$ cannot be rerouted as it is not perturbed. One drawback of the method is in obtaining long routing paths and in some cases including cycles (Nace et al. 2013). This happens because the strategy releases all disturbed flows and reroute them so longer they are, more often they will be affected, released and rerouted (Nace et al. 2011). To the extreme end, it could be possible to release all paths (all are disturbed) and reroute all from scratch, which comes to be GR.

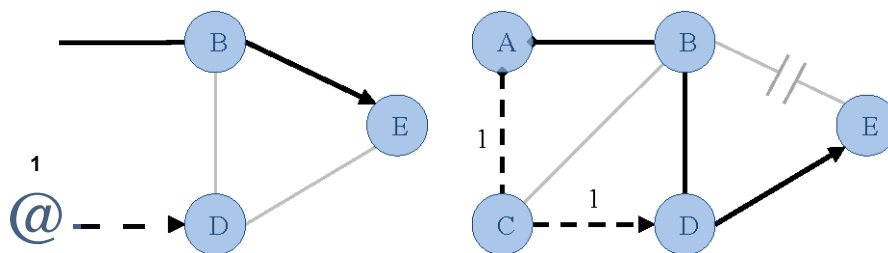


Figure 2.9- RR+ before (left) and after (right) failure of link (B,E).

2.2.2.4 End-to-end rerouting without stub release- RR-

This strategy reroutes the lost traffic without using capacities released by the disturbed paths i.e. the capacities are dedicated to the routing or to the rerouting but not to both of them.

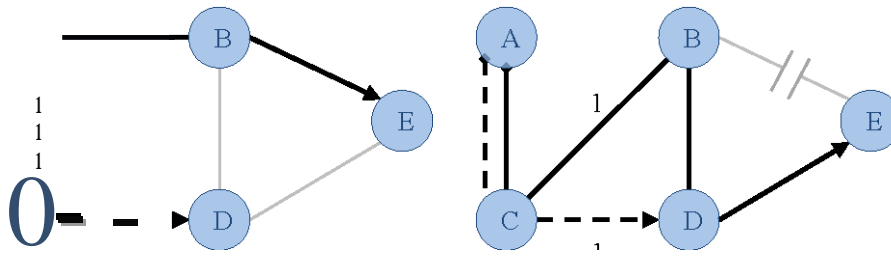


Figure 2.10- RR- before (left) and after (right) failure of link (B,E).

This strategy is easier to implement than the RR+ strategy but it is more costly due to the split between nominal and restoration available capacities. In Figure 2.10, we consider that 2 units of bandwidth are available on link (A,C) and only 1 unit on the other links. After failure of link (B,E), the traffic cannot be rerouted using link (A,B) as there is no room left to route the traffic (remember that rerouting is done without using capacities released by the disturbed paths). Hence, demand (A,E) can only be rerouted on path {A-C-B-D-E}.

2.2.2.5 Shared Robust Rerouting- ShRR

Shared Robust Rerouting (ShRR) (Fundo et al. 2013) is a hybrid end-to-end rerouting method based on the following ideas. First, like for the PD strategy, routing paths allow routing a traffic demand beyond the required demand, i.e. higher than 100% of the demand. This guarantees a high level of robustness when traffic varies. Furthermore the amount of traffic to be rerouted is generally less than for the other rerouting strategies: for instance, it may be that for a given affected demand, the non-disrupted routing paths allow full routing of the demand and no rerouting is needed. The second feature is concerned with the fact that the non-disrupted demands are allowed to release a part of their bandwidth in order to accommodate the rerouting of the disrupted demands. Last, as for the other end-to-end restoration strategies, the rerouting linked to a demand can use paths out of the set of nominal paths. Given all the above, the ShRR strategy combines features from the following recovery techniques: PD, RR+ and GR. In terms of cost-effectiveness this strategy is somewhere between GR and RR+. Note that ShRR is the base of our work on *Elastic Flow Rerouting*.

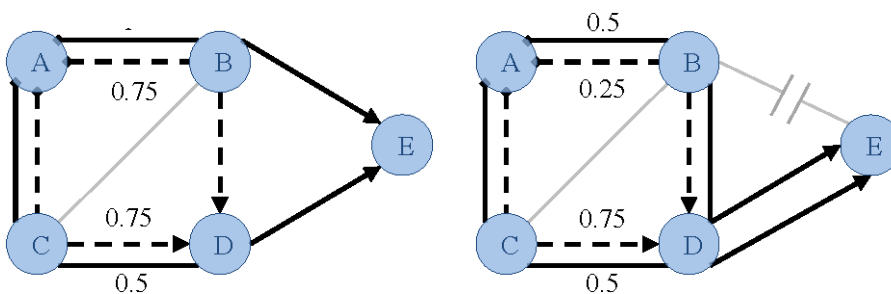


Figure 2.11 – ShRR before (left) and after (right) failure of link (B,E).

ShRR is illustrated by Figure 2.11. Demand (A,E) is routed through

a two LSPs $\{A-B-E\}$ and $\{A-C-D-E\}$, and demand (A,D) is routed using two paths $\{A-B-D\}$ and $\{A-C-D\}$. When link (B,E) fails, the bandwidth associated to path $\{A-B-D\}$ is reduced by 0.5 units to make room for new (rerouting) paths on links (A,B) and (B,D) . Hence, demand (A,E) can be rerouted using path $\{A-B-D-E\}$.

2.2.3 MPLS feature - MPLS Fast Rerouting

MPLS Fast Reroute (Pan et al. 2005) (also called MPLS local restoration) is a hybrid strategy between local rerouting and end-to-end rerouting. MPLS Fast Reroute works as follows. A LSP is set for a demand. When a link of this LSP fails, the upstream node of the failed link informs the source node of the demand and reroute the traffic directly to the destination. This is a temporary action which enable fast recovery of the traffic. Once the ingress node is aware of the failure, it switches the traffic to a protection LSP from the source to the destination nodes of the demand.

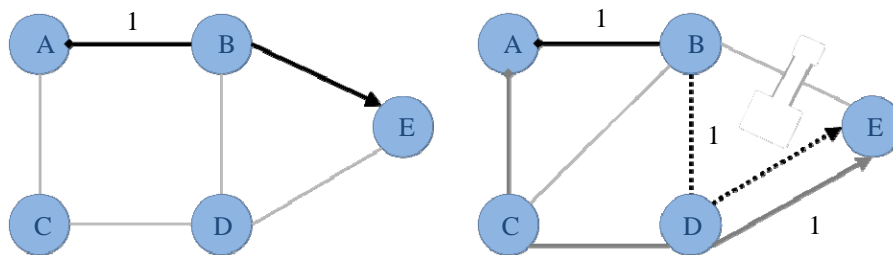


Figure 2.12 – *MPLS Fast Reroute before (left) and after (right) failure of link (B,E).*

Figure 2.12 shows how MPLS Fast Reroute works. A path is established to route the traffic from node A to node E . When link (B,E) fails, the traffic is temporary rerouted from node B to the demand destination E . A few milliseconds later, the ingress node A reroutes all the traffic on path $\{A-C-D-E\}$.

2.3 Strategies for partial link failures

In this section, we present mechanisms that are designed to manage partial link failures or that can be adapted to manage them.

2.3.1 Probabilistic Routing - PR

Claßen et al. (2011) propose a method that we call Probabilistic Routing which has been designed for wireless microwave networks and could be seen as an extension of Path Diversity to manage partial link failures. It ensures that the flows can carry the required traffic with high probability. Indeed, when designing the network to be used with PR, the capacity of each link is computed so that, in case of failure i.e. change of modulation, the link can carry out the traffic. If we consider that a probability is given to each link and each possible modulation scheme. Hence, as the available capacity on the link depends on the modulation scheme, a probability is given to each failure scenarios. Taking into account these probabilities,

a nominal capacity (corresponding to the best modulation scheme) is assigned to each link while capacities in case of failure only depend on the change of modulation. Contrary to PD, the traffic routed for a demand in the nominal case is not higher than the requirements but the capacity of the links is highly over-dimensioned, as it can be observed in Figure 2.13.

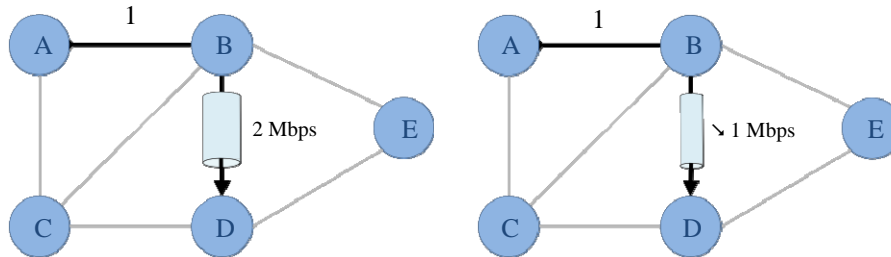


Figure 2.13 – PR before (left) and after (right) failure of link (B,E).

2.3.2 Elastic Traffic Flows - ETF

Elastic Traffic Flows (Tomaszewski 2014) has been designed for wireless optical networks that carry elastic packet flows. A bandwidth is associated to each demand, and a set of paths carry out the required traffic in the nominal state. In case of failure, path flows can be decreased by a percentage (up to a certain threshold) with respect to the max-min fairness sharing.

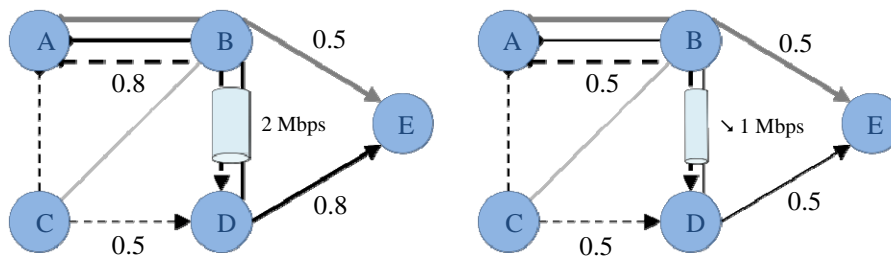


Figure 2.14 – ETF before (left) and after (right) failure of link (B,E).

In Figure 2.14, two LSPs are established for each demand: paths {A-B-E} and {A-B-D-E} for demand (A,E), and paths {A-B-D} and {A-C-D} for demand (A,D). When the capacity of link (B,D) is reduced by 50%, there is not enough room for paths {A-B-D-E} and {A-B-D}. Hence, each of flows going through (B,D) share the capacity according to max-min fairness and get 50% each. Note that Elastic Traffic Flows has been presented after FT and EFR strategies.

2.3.3 Global Rerouting - GR

Global Rerouting has not been designed for partial link failures. However, as all nominal paths are released after a failure, partial failure can be easily managed. Note that this strategy remains the most cost efficient for the management of partial link failures. Figure 2.15 illustrates the behavior of Global Rerouting when suffering from a partial link failure. In the nominal

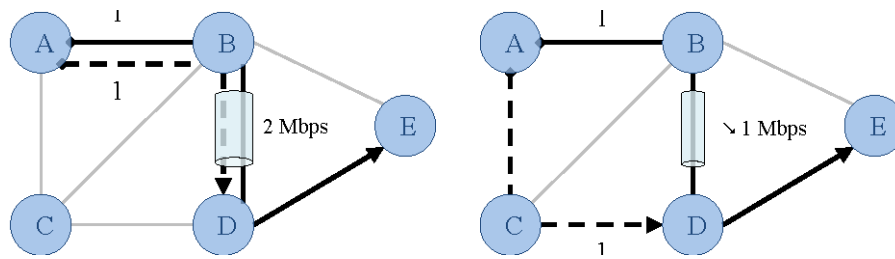


Figure 2.15- GR before (left) and after (right) failure of link (B,E).

state, demand (A,E) is routed on path {A-B-D-E} and demand (A,D) is routed on path {A-B-D}. When link (B,D) suffers a degradation of 50% of its nominal capacity (equals to 2 Mbps), there is not enough capacity for the two paths. Hence, the two paths are released and the traffic is rerouted through paths {A-B-D-E} and {A-C-D} respectively for demands (A,E) and (A,D). The bandwidth of the paths routed on link (B,D) is now equal to its reduced capacity of 1 Mbps.

2.3-4 MPLS feature - MPLS TE Auto-bandwidth Adjustment

MPLS TE Auto-Bandwidth Adjustment (Dhody et al. 2015) allows an LSP to automatically and dynamically adjust its reserved bandwidth over time, i.e. without network operator intervention. The bandwidth adjustment uses the make-before-break adaptive signaling method so that there is no interruption to traffic flow.

The new bandwidth reservation is determined by sampling the actual traffic flowing through the LSP. If the traffic flowing through the LSP is lower than the configured or current bandwidth of the LSP, the extra bandwidth is being reserved needlessly and being wasted. Conversely, if the actual traffic flowing through the LSP is higher than the configured or current bandwidth of the LSP, it can potentially cause congestion or packet loss. Contrary to the presented strategies, Auto-bandwidth is not computed off-line, the LSP bandwidth can be set to some arbitrary value (even zero) during initial setup time, and it will be periodically adjusted over time based on the actual bandwidth requirement.

The traffic rate is repeatedly sampled at each interval and the sampled traffic rates are accumulated over the interval period. The ingress node reports the traffic information. Auto-Bandwidth will adjust the bandwidth of the LSP to the highest sampled traffic rate amongst the set of samples taken over the adjustment-interval. Note that the highest sampled traffic rate could be higher or lower than the current LSP bandwidth.

Auto-Bandwidth Adjustment is a mechanism that deals with capacity changes (i.e. partial failures). RSVP tunnels (used by MPLS TE) are periodically refreshed to recover from network capacity changes. If the capacity of a link has been reduced, the refresh process will fail, the LSP will be broken and the traffic will be rerouted to other LSPs. Instead of removing the LSP, Auto-Bandwidth feature will decrease the bandwidth of the LSP. As the traffic is too high, it will then try to enlarge the LSP.

Because of the failure, this action will fail and the traffic will be rerouted through other LSPs.

Network Dimensioning Problem

Contents	
3.1	Notation 23
3.2	Study of the nominal state 25
3.2.1	Dual problem and dual separation 26
3.2.2	Path generation 27
3.3	Study of Global Rerouting 27
3.3.1	Dual problem and dual separation 29
3.3.2	Path generation 29
3.4	Mathematical formulation of the restoration methods 30
3.4.1	End-to-end rerouting with stub release - RR^+ 30
3.4.2	End-to-end rerouting without stub release - RR^- 32
3.4.3	Local Rerouting - LR 32
3.4.4	Path Diversity - PD 33
3.4.5	Shared Robust Rerouting - ShRR 34
3.5	Numerical results for all mechanisms 35
3.5.1	Dimensioning cost 35
3.5.2	Efficiency of Path Generation 35

In this chapter, we present the dimensioning problem of some recovery mechanisms of the literature. We first introduce some notations. Then, we study the modelisation of the nominal state, and continue with dimensioning problem for networks resilient to total link failures using restoration presented in Chapter 2.

3.1 Notation

The considered network is modeled with a graph $G = (V, E)$, undirected or directed, composed of the set of nodes V and the set of links E . In the sequel, we will always assume directed graphs, unless explicitly stated otherwise. Thus, each link $e \in E$ represents a directed pair (v, w) of nodes

$v, w \in V$, and is assigned a non-negative unit capacity cost ξ_e which is a parameter, and the maximum capacity y_e^0 which is an optimization variable. The cost of the network is given by the quantity $C = \sum_{e \in E} \xi_e y_e^0$.

The traffic demands are represented by the set D . Each demand $d \in D$ is characterized by a directed pair $(o(d), t(d))$, composed of the originating node $o(d)$ and the terminating node $t(d)$, and a reference value h_d^0 (a parameter) of the traffic volume that has to be carried from $o(d)$ to $t(d)$. Here $\delta^+(v) = \{(v, w) \in V\}$ and $\delta^-(v) = \{(u, v) \in V\}$ are the outgoing arcs and ingoing arcs of v , respectively. Demand volumes and link capacities are expressed in the same units.

Each demand d has a specified set of admissible paths P_d (called the path-list) composed of selected elementary paths from $o(d)$ to $t(d)$ in graph G . Recall that an elementary path does not traverse any node more than once. Paths in P_d , used to realize the demand (traffic) volumes, are assigned flows x_{dp}^0 , $p \in P_d$, which are optimization variables. Each value x_{dp}^0 specifies the reference capacity (expressed in the same units as link capacity and demand volume) reserved on the tunnel realized by path $p \in P_d$. The set of all admissible paths is denoted by $P := \bigcup_{d \in D} P_d$. The maximum path-lists, i.e., path-lists P_d containing all elementary paths from $o(d)$ to $t(d)$, will be denoted by \hat{P}_d , $d \in D$, with $\hat{P} := \bigcup_{d \in D} \hat{P}_d$.

Since we assume only elementary paths, we will sometimes identify the paths with the sets of links they traverse: $p \subseteq E$, $p \in P$. The given sets of admissible paths define the link-path incidence coefficients δ_{edp} , $e \in E$, $d \in D$, $p \in P_d$, where $\delta_{edp} = 1$ if path $p \in P_d$ traverses link $e \in E$, i.e., if $e \in p$, and $\delta_{edp} = 0$ otherwise. It is important to note that the sets of admissible paths P_d , $d \in D$, are parameters in the problem formulations considered in the sequel, although in general we assume that all possible elementary paths can potentially be used if this is required to achieve the optimum.

Note that we can represent flows, not as an end-to-end path-flow, but as a flow routed through an edge. These flows are defined, for each demand, by the vector $f^d = (f_e^d, e \in E, d \in D)$.

The basic feature of the considered network model is that the links are subject to failures. In the nominal state S_0 (failure-less situation), link capacities are defined by the vector $y^0 = (y_e^0, e \in E)$. In a failure scenario, a subset of links have no capacity (failed links), and the remaining links are totally operational. We thus distinguish a set of failure scenarios S . Each state $s \in S$ is characterized by the two following fixed vectors: the link availability coefficient vector $\alpha^s := (\alpha_e^s, e \in E)$, and the traffic demand satisfaction coefficient vector $\beta := (\beta_d^s, d \in D)$. Basically, we assume $\alpha_e^s = 0$ for all edges e which failed in failure state S and $\alpha_e^s = 1$ for all other links. The capacity of link $e \in E$ available in state s is equal to $y_e^s := \alpha_e^s y_e^0$ while the volume of demand $d \in D$ to be carried in s is given by $h_d^s := \beta_d^s h_d^0$ (the fixed parameter β_d^s is called the **demand coefficient**). Recall that $h^0 := (h_d^0, d \in D)$ is a given demand volume reference vector.

The set of links that are not available in state $s \in S$ will be denoted by $E_s := \{e \in E : \alpha_e^s < 1\}$, and, symmetrically, the set of states in which link $e \in E$ is not available, by $S_e := \{s \in S : \alpha_e^s < 1\}$. Note that $E_s = \emptyset$ for each $s \in S_0$.

3.2 Study of the nominal state

Now we introduce the basic problem related to the nominal routing. The network dimensioning problem is to find a feasible flow minimizing the overall cost of the network i.e. the sum of the cost of the links of the network. The problem can be represented by the following edge-node linear programming (LP) formulation involving variables f_e^d ($d \in D, e \in E$), and y_e^0 ($e \in E$):

$$\min C = \sum_{e \in E} \xi_e y_e^0 \quad (3.1a)$$

$$\sum_{d \in D} f_e^d \leq y_e^0, \quad e \in E \quad (3.1b)$$

$$\sum_{e \in \delta^+(o(d))} f_e^d + \sum_{e \in \delta^-(o(d))} f_e^d \geq h_d^0, \quad d \in D \quad (3.1c)$$

$$\sum_{e \in \delta^+(v)} f_e^d - \sum_{e \in \delta^-(v)} f_e^d = 0, \quad d \in D, v \in V \setminus \{o(d), t(d)\} \quad (3.1d)$$

$$f, y \text{ continuous and nonnegative} \quad (3.1e)$$

Objective (3.1a) minimizes the total cost of links. The first capacity constraint (3.1b) makes sure that the reference link loads do not exceed the maximum link capacities. Next, the demand constraint (3.1c) assures that the net out-flow of the demand sources are sufficient to realize the volume of each demand $d \in D$. Equations (3.1d) are called flow conservation constraints (or Kirchoff constraint), since they guarantee that the in-flow equals the out-flow for all nodes except $o(d)$ and $t(d)$. Note that this LP can be solved polynomially.

We now give a path formulation for the nominal routing problem. The problem assumes that the required demand volume h_d^0 ($d \in D$) is realized within the bandwidth reserved by the reference flows x_{dp}^0 , $p \in P_d$, so that the links capacities y_e^0 , $e \in E$, available in state s are not exceeded. For given admissible path-lists P_d , $d \in D$, the considered problem is denoted by $P(P)$ (where $P = \bigcup_{d \in D} P_d$) and represented by the following link-path linear programming (LP) formulation involving variables x_{dp}^0 ($d \in D, p \in P_d$), and y_e^0 ($e \in E$):

$$\min C = \sum_{e \in E} \xi_e y_e^0 \quad (3.2a)$$

$$[\pi_e^0 \geq 0] \quad y_e^0 - \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \geq 0, \quad e \in E \quad (3.2b)$$

$$[\lambda_d^0 \geq 0] \quad \sum_{p \in P_d} x_{dp}^0 \geq h_d^0, \quad d \in D, s \in S \quad (3.2c)$$

$$x^0, y^0 \text{ continuous and nonnegative} \quad (3.2d)$$

Objective (3.2a) minimizes the total cost of links. The first capacity constraint (3.2b) makes sure that the reference link loads do not exceed the maximum link capacities. Next, the demand constraint (3.2c) assures that the flows are sufficient to realize the volume of each demand $d \in D$. The optimal value of the cost (3.2a) of $P(P)$ will be denoted by $C(P)$.

In (3.2), the quantities in the square brackets on the left-hand side are dual variables. Dual variable π_e^0 corresponds to the increase of the objective function (3.2a) for one unit of bandwidth reserved on edge $e \in E$ without being used by any paths. In the same way, λ_d^0 represents the increase of the objective function for one unit of traffic increase for demand $d \in D$. Let us introduce reduced costs. While dual variables are associated to constraints in the LP, reduced costs are associated to the variables. The reduced cost associated to an edge $e \in E$ takes 0 if the edge is used, i.e. at least one path is routed through this edge, c_e otherwise. The reduced cost corresponding to a path $p \in P_d$ represent the increase of the objective function if the path is used, and takes value 0 if it is already used. These dual variables and reduced costs will be of high interest when studying dual separation in subsection 3.2.1.

Observe, that our assumption of admitting only elementary paths on the path-lists made in 3.1 is not limiting. If we admitted non-elementary paths in the overall path-list P , then eliminating loops in any feasible solution of $P(P)$ would never increase $C(P)$. For the full sets of admissible paths, the corresponding instance of problem (3.2) is referred to as $P(\hat{P})$. Contrary to (3.1), this formulation would require adding all paths to the problem and the number of variables would become very large.

In order to avoid an exponentially high number of variables, we will use Path Generation (PG). In this context, $P(P)$ with $P \subset \hat{P}$ a subset of all admissible paths, represents the restricted master problem. Note that a feasible solution $x_{dp}^0, d \in D, p \in P_d$ for $P(P)$ can be expanded to a feasible solution $x_{dp}^0, d \in D, p \in \hat{P}_d$ for $P(\hat{P})$ (setting $x_{dp}^0 = 0$ for $p \in \hat{P}_d \setminus P_d$).

3.2.1 Dual problem and dual separation

Let us consider the problem dual to problem $P(P)$ formulated in (3.2) for a given set of admissible paths P . The dual, denoted by $D(P)$, is as follows:

$$\max W = \sum_{d \in D} h_d^0 \lambda_d^0 \quad (3.3a)$$

$$\pi_e^0 \leq \xi_e, \quad e \in E \quad (3.3b)$$

$$\lambda_{dp}^0 \leq \sum_{e \in E} \delta_{edp} \pi_e^0, \quad d \in D, p \in P_d \quad (3.3c)$$

$$\text{all variables } \lambda, \pi \text{ continuous and nonnegative.} \quad (3.3d)$$

Let $\Pi(P)$ denote the dual polyhedron of all feasible solutions (π, λ) of $D(P)$, where $\pi = (\pi_e^0, e \in E)$, $\lambda = (\lambda_d^0, d \in D)$. We treat the dual solutions (π, λ) as vectors in the \mathbb{R}^N space of dimension $N = |E| + |D|$, and observe that $\Pi(P)$ is a fully dimensional subset of \mathbb{R}^N .

Let $W(P)$ denote the optimal value of the objective function (3.3a) of $D(P)$. Clearly, $W(P)$ is also equal to the optimal value of objective (3.3a) maximized over $(\pi, \lambda) \in T(P)$. Due to the strong duality property (Lasdon 1970, Minoux 1986), $W(P) = C(P)$. Moreover, $P \subseteq \hat{P}$ implies that polyhedron $\Pi(P)$ contains polyhedron $\Pi(\hat{P})$, so that $W(P) \geq W(\hat{P})$ and hence $W(P)$ is actually an upper bound of the optimal objective $C(\hat{P})$ of $P(\hat{P})$.

Suppose π, λ form an optimal solution of the dual problem $D(P)$ defined by (3.3). Checking whether the above inequalities are satisfied for all paths $q \in \hat{P}_d$ is called pricing subproblem. This can be done efficiently by computing paths w.r.t. the weight vector $\pi_e^0, e \in E$ for each commodity $d \in D$. If a path is at least λ_d , then we must have:

$$\sum_{e \in E} \delta_{edp} \pi_e^0 \geq \lambda_{dp}^0, \quad d \in D, p \in P_d \quad (3.4)$$

Otherwise, we find a new path q that has the positive reduced cost and we can add this path (column) to the new master problem $P(P \cup q)$.

3.2.2 Path generation

In fact, in path generation it is advantageous (to speed up the PG algorithm) to find, for each $d \in D$, not only a path (if any) in \hat{P}_d that just separates the current dual solution π, λ , but rather a path $q \in \hat{P}_d$ for which the dual constraints (4.2c) are maximally violated by the considered dual solution π, λ . This is especially true when finding such a path is not substantially more complex than finding an arbitrary path that violates the dual constraints. In our setting, the (negative) measure of violation of the dual constraints corresponding to path q is equal to $|q| = \sum_{e \in E} \delta_{edp} \pi_e^0 - \lambda_d^0$ provided it is negative. Maximizing the measure of violation is equivalent to minimizing $\sum_{e \in E} \delta_{edp} \pi_e^0$. This can be done efficiently by computing the shortest path w.r.t. the weight vector $\pi_e^0, e \in E$ for each commodity $d \in D$. Note that finding a shortest path in graph $G(V, E)$ with nonnegative weights $\pi_e^0 \geq 0, e \in E$ is polynomial (e.g., by Dijkstra's algorithm).

The PG algorithm is then described in Algorithm 1.

Algorithm 1: PG algorithm for the nominal routing

Input: Graph $G = (V, E)$, a set of demands D , a vector of traffic requirements h_d^0 and an admissible paths-list P .

Output: Subset of paths $P \subseteq \hat{P}$ leading to optimal cost $C(P)$.

- 1: Repeat
 - 2: Solve $P(P)$
 - 3: Let π, λ be the dual variables for $D(P)$
 - 4: For each demand $d \in D$
 - 5: Find a shortest path w.r.t. weights π_e^0
 - 6: If length of shortest path q is less than λ_d^0 add q to P
 - 7: End If
 - 8: End for
 - 9: Until no path has been added
-

3.3 Study of Global Rerouting

As we did in the previous section, let us introduce the edge-node LP formulation of GR involving variables $f_{d,e}^0 (d \in D, e \in E)$, $f_{d,e}^s (d \in D, s \in S, e \in E \setminus s)$ and $y_e^0 (e \in E)$:

$$\min C = \sum_{e \in E} \xi_e y_e^0 \quad (3.5a)$$

$$\sum_{d \in D} f_{d,e}^0 \leq y_e^0, \quad e \in E \quad (3.5b)$$

$$\sum_{e \in \delta^+(o(d))} f_{d,e}^0 + \sum_{e \in \delta^-(o(d))} f_{d,e}^0 \geq h_d^0, \quad d \in D \quad (3.5c)$$

$$\sum_{e \in \delta^+(v)} f_{d,e}^0 - \sum_{e \in \delta^-(v)} f_{d,e}^0 = 0, \quad d \in D, v \in V \setminus \{o(d), t(d)\} \quad (3.5d)$$

$$\sum_{d \in D} f_{d,e}^s \leq y_e^0, \quad s \in S, e \in E \quad (3.5e)$$

$$\sum_{e \in \delta^+(o(d))} f_{d,e}^s + \sum_{e \in \delta^-(o(d))} f_{d,e}^s \geq h_d^0, \quad s \in S, d \in D \quad (3.5f)$$

$$\sum_{e \in \delta^+(v)} f_{d,e}^s - \sum_{e \in \delta^-(v)} f_{d,e}^s = 0, \quad s \in S, d \in D, v \in V \setminus \{o(d), t(d)\} \quad (3.5g)$$

$$f^0, f^s, y \text{ continuous and nonnegative} \quad (3.5h)$$

Objective (3.5a) minimizes the total cost of links. Capacity constraints (3.5b) and (3.5e) make sure that the reference link loads do not exceed the maximum link capacities in the nominal and failure states. Next, the demand constraints (3.5c) (resp. (3.5f)) assure that the out-flow of the demand sources are sufficient to realize the volume of each demand $d \in D$ in the nominal (resp. failure) state. Equations (3.5d) and (3.5g) are called flow conservation constraints, since they guarantee that the in-flow equals the out-flow for all nodes except $o(d)$ and $t(d)$. Note that this LP can be solved polynomially.

We now give a path formulation for the problem. For given admissible path-lists $P_d, d \in D$, the considered problem is denoted by $P(P)$ (where $P = \bigcup_{d \in D} P_d$) and represented by the following link-path linear programming (LP) formulation involving variables x_{dp}^0 ($d \in D, p \in P_d$), and y_e^0 ($e \in E$):

$$\min C = \sum_{e \in E} \xi_e y_e^0 \quad (3.6a)$$

$$[\pi_e^0 \geq 0] \quad y_e^0 - \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \geq 0, \quad e \in E \quad (3.6b)$$

$$[\lambda_d^0 \geq 0] \quad \sum_{p \in P_d} x_{dp}^0 \geq h_d^0, \quad d \in D \quad (3.6c)$$

$$[\pi_e^s \geq 0] \quad y_e^0 - \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^s \geq 0, \quad s \in S, e \in E \quad (3.6d)$$

$$[\lambda_d^s \geq 0] \quad \sum_{p \in P_d} x_{dp}^s \geq h_d^s, \quad d \in D, s \in S \quad (3.6e)$$

$$x^0, y^0 \text{ continuous and nonnegative} \quad (3.6f)$$

(where the quantities in the square brackets on the left-hand side are dual variables).

Objective (3.6a) minimizes the total cost of links. The first capacity constraint (3.6b) makes sure that the reference link loads do not exceed

the maximum link capacities. Next, the demand constraint (3.6c) assures that the flows are sufficient to realize the volume of each demand $d \in D$. Capacity constraint (3.6d) makes sure that the link loads do not exceed the maximum link capacities in any failure scenario s . Finally, constraint (3.6e) assures that the flows are sufficient to realize the possibly reduce volume of each demand. The optimal value of the cost (3.6a) of $P(P)$ will be denoted by $C(P)$.

3.3.1 Dual problem and dual separation

Let us consider the problem dual to problem $P(P)$ formulated in (3.6) for a given set of admissible paths P . The dual, denoted by $D(P)$, is as follows:

$$\max W = \sum_{d \in D} h_d^0 \lambda_d^0 + \sum_{s \in S} \sum_{d \in D} h_d^s \lambda_d^s \quad (3.7a)$$

$$\pi_e^0 + \sum_{s \in S} \pi_e^s \leq \xi_e, \quad e \in E \quad (3.7b)$$

$$\lambda_{dp}^0 \leq |p|^0, \quad d \in D, p \in P_d \quad (3.7c)$$

$$\lambda_{dp}^s \leq |p|^s, \quad s \in S, d \in D, p \in P_d^s \quad (3.7d)$$

$$\text{all variables } \lambda, \pi^0, \pi^s \text{ continuous and nonnegative.} \quad (3.7e)$$

Above $|p|^0 = \sum_{e \in E} \delta_{edp} \pi_e^0$ and $|p|^s = \sum_{e \in E} \delta_{edp} \pi_e^s$, $s \in S$ denote the state-dependent (dual) lengths of path p .

Below, $D(P)$ will denote the problem defined by (3.7) for a given set of admissible paths $P = \bigcup_{d \in D} P_d$, $\Pi(P)$ —the polyhedron (in \mathbb{R}_+^n of an appropriate dimension n) defined by constraints (4.2b)-(4.2e), and $W(P)$ —the optimal value of the objective function of $D(P)$. Certainly, $W(P)$ is equal to the optimal value $C(P)$ considered for the admissible path-set P , and $W(\hat{P})$ (recall that \hat{P} is the maximum set of admissible paths) is the solution of GR we are looking for. Certainly, $W(P) \geq W(\hat{P})$ for $P \subseteq \hat{P}$.

Suppose π, λ form an optimal solution of the dual problem $D(P)$ defined by (3.7). We consider the following two cases that fully characterize the set of all paths $q \in \mathcal{P}_d \not\subseteq P_d$ that separate the current optimal dual solution.

Case 1 There exists a path q such that $|q|^0 < \lambda^0$ which, by equation (3.7c), makes the inequalities in (3.7) infeasible.

Case 2 Suppose that $|q|^0 \geq \lambda^0$ for a path q , and for some $s \in S$, $|q|^s < \lambda^s$. Equation (3.7d) does not allow any $|q|^s - \lambda^s$ to be negative. Hence, (3.7) is infeasible.

3.3.2 Path generation

Because of the specific form of the LP, we can use Benders decomposition (Benders 1962). Hence, the path generation process requires a subproblem for each demand $d \in D$, for the nominal state and for each failure state $s \in S$. Subproblems related to the nominal state are the same than in 3.2.1. Failure state related subproblems can efficiently be solved by computing the shortest path w.r.t. the weight vector $\pi_e^s, e \in E, s \in S$ and $\pi_e^0, s \in S, e \in$

$E \setminus s$ for each commodity $d \in D$. Note that finding a shortest path in graph $G(V, E)$ with nonnegative weights $\pi_e^0 \geq 0, e \in E$ or $\pi_e^s \geq 0, s \in S, e \in E \setminus s$ is polynomial (e.g., by Dijkstra's algorithm). The PG algorithm is then described in Algorithm 2.

Algorithm 2: PG algorithm for Global Rerouting

Input: Graph $G = (V, E)$, a set of demands D , a vector of traffic requirements $h_d^s, s \in S \cup S_0$ and an admissible paths-list P .

Output: Subset of paths $P \subseteq \hat{P}$ leading to optimal cost $C(P)$.

- 1: Repeat
 - 2: Solve $P(P)$
 - 3: Let π, λ be the dual variables for $D(P)$
 - 4: For each demand $d \in D$
 - 5: Find a shortest path w.r.t. weights π_e^0
 - 6: If length of shortest path q is less than λ_d^0 , add q to (GR)
 - 7: End If
 - 8: For each failure state $s \in S$
 - 9: For each demand $d \in D$
 - 10: Find a shortest path w.r.t. weights π_e^s
 - 11: If length of shortest path q is less than λ_d^s , add q to P
 - 12: End If
 - 13: End for
 - 14: End for
 - 15: End for
 - 16: Until no path has been added
-

3.4 Mathematical formulation of the restoration methods

In this section, we study the dimensioning problem for five other recovery methods of the literature: end-to-end rerouting with stub release (RR^+), end-to-end rerouting without stub release (RR^-), local rerouting (LR), path diversity (PD) and shared robust rerouting (ShRR).

3.4.1 End-to-end rerouting with stub release - RR^+

The arc-path formulation of the problem dimensioning problem for RR^+ , denoted $P_{RR^+}(P)$, is as follows:

$$\min C = \sum_{e \in E} \xi_e y_e^0 \quad (3.8a)$$

$$[\pi_e^0 \geq 0] \quad y_e^0 - \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \geq 0, \quad e \in E \quad (3.8b)$$

$$[\lambda_d^0 \geq 0] \quad \sum_{p \in P_d} x_{dp}^0 \geq h_d^0, \quad d \in D \quad (3.8c)$$

$$[\pi_e^s \geq 0] \quad y_e^0 - \sum_{d \in D} \sum_{p \in P_d, s \notin p} \delta_{edp} x_{dp}^0 - \sum_{d \in D} \sum_{p \in P_d^s} \delta_{edp} x_{dp}^s \geq 0, \quad s \in S, e \in E \setminus s \quad (3.8d)$$

$$[\lambda_d^s \geq 0] \quad \sum_{p \in P_d, s \notin p} x_{dp}^0 + \sum_{p \in P_d^s} x_{dp}^s \geq h_d^s, \quad d \in D, s \in S \quad (3.8e)$$

$$x^0, x^s, y^0 \text{ continuous and nonnegative} \quad (3.8f)$$

Constraints (3.8b) (respectively (3.8d)) express capacity constraints in the nominal (respectively failure) state. Constraints (3.8c) ensure the satisfaction of traffic in the nominal state and constraints (3.8e) ensure the lowest traffic satisfaction rate for the failure scenario s . Finally, the non-negativity of variables is expressed by constraints (3.8f) and dual variables $(\pi_e^0, \pi_e^s, \lambda_d^0, \lambda_d^s)$ are given in the left-hand terms.

Let us consider the problem dual to problem $P_{RR^+}(P)$ formulated in (3.8) for a given set of admissible paths P . The dual is as follows:

$$\max W = \sum_{d \in D} h_d^0 \lambda_d^0 \quad (3.9a)$$

$$\pi_e^0 + \sum_{s \in S} \pi_e^s \leq \xi_e, \quad e \in E \quad (3.9b)$$

$$\lambda_d^0 + \sum_{s \in S, s \notin p} \lambda_d^s \leq \sum_{e \in E} \delta_{edp} \pi_e^0 + \sum_{s \in S, s \notin p, e \in E} \delta_{edp} \pi_e^s, \quad d \in D, p \in P_d \quad (3.9c)$$

$$\lambda_d^s \leq \sum_{e \in E} \delta_{edp} \pi_e^s, \quad s \in S, d \in D, p \in P_d^s \quad (3.9d)$$

$$\text{all variables } \lambda^0, \lambda^s, \pi^0, \pi^s \text{ continuous and nonnegative.} \quad (3.9e)$$

Using notations introduced in section 3.3 (i.e. $|q|^0$ and $|q|^s$), we can extract from 3.9 that a path $q \in \hat{\mathcal{P}}_d \setminus P_d$ will be added to 3.8 if it meets one of the following cases.

Case 1 $|q|^s < \lambda_d^s$. This can be computed using Dijkstra's algorithm with nonnegative weights $\pi_e^s \geq 0, s \in S, e \in E \setminus s$.

Case 2 $|q|^0 + \sum_{s \in S, s \notin q} |q|^s < \lambda_d^0 + \sum_{s \in S, s \notin q} \lambda_d^s$. This inequality can easily be transformed to $|q|^0 + \sum_{s \in S, s \notin q} |q|^s + \sum_{s \in S, s \in q} \lambda_d^s < \lambda_d^0 + \sum_{s \in S} \lambda_d^s$. Note that the second term of the above equation ($\sum_{s \in S, s \notin q} |q|^s$) makes the shortest reduced cost problem difficult. It is similar to the quadratic shortest path problem, a NP -hard problem. Notice, for instance, that the property of the optimality of any sub-path of an optimal path does not apply to this case.

3.4.2 End-to-end rerouting without stub release - RR⁻

RR⁻ reroutes the disturbed demands from the source to the destination without using the capacities of the disturbed paths. In this strategy, capacities are used for the nominal state or for the failure state, but not both. The arc-path formulation of the dimensioning problem for end-to-end rerouting without stub release, denoted $P_{RR^-}(P)$, is as follows:

$$\min C = \sum_{e \in E} \xi_e y_e^0 \quad (3.10a)$$

$$y_e^0 - \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \geq 0, \quad e \in E \quad (3.10b)$$

$$\sum_{p \in P_d} x_{dp}^0 \geq h_d^0, \quad d \in D \quad (3.10c)$$

$$y_e^0 - \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 - \sum_{d \in D} \sum_{p \in P_d^s} \delta_{edp} x_{dp}^s \geq 0, \quad s \in S, e \in E \setminus s \quad (3.10d)$$

$$\sum_{p \in P_d, s \neq p} x_{dp}^0 + \sum_{p \in P_d^s} x_{dp}^s \geq h_d^s, \quad d \in D, s \in S \quad (3.10e)$$

$$x^0, x^s, y^0 \text{ continuous and nonnegative} \quad (3.10f)$$

Constraints (3.10b) (respectively (3.10d)) express capacity constraints in the nominal (respectively failure) state. Constraints (3.10c) ensure the satisfaction of traffic in the nominal state and constraints (3.10e) ensure the lowest traffic satisfaction rate for the failure scenario s . Finally, the non-negativity of variables is expressed by constraints (3.10f). Note that in constraints (3.10d) all paths are considered, while in constraints (3.10e) only non-disturbed paths are taken into account. Note that path generation for RR⁻ is polynomial (Lutton et al. 2000).

3.4.3 Local Rerouting - LR

The rerouting process is local if, in case of a link failure, we do not reroute the disturbed demands from the source to the destination but we only reroute the traffic between the extremities of the failure link. Let us introduce some more notations:

- Let P^l be the set of paths of graph $G(V, E \setminus l)$ between the extremities of link $l \in E$.
- Let x_p^l be the proportion of cumulated traffic routed on path $p \in P^l$ between the extremities of failed link $l \in E$.

The formulation of the dimensioning problem for local rerouting, denoted $P_{LR}(L)$, is as follows:

$$\min C = \sum_{e \in E} \xi_e y_e^0 \quad (3.11a)$$

$$y_e^0 - \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \geq 0, \quad e \in E \quad (3.11b)$$

$$\sum_{p \in P_d} x_{dp}^0 \geq h_d^0, \quad d \in D \quad (3.11c)$$

$$y_e^0 - \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 - \sum_{p \in P^I} \delta_{ep} x_p^I \geq 0, \quad I \in E, e \in E \setminus \{I\} \quad (3.11d)$$

$$\sum_{p \in P^I} x_p^I \geq \sum_{p \in P_d, I \in p} x_{dp}^0, \quad d \in D, I \in E \quad (3.11e)$$

$$x^0, x^s, y^0 \text{ continuous and nonnegative} \quad (3.11f)$$

Constraints (3.11b) (respectively (3.11d)) express capacity constraints in the nominal (respectively failure) state. Constraints (3.11c) ensure the satisfaction of traffic in the nominal state and constraints (3.11e) ensure that all the lost traffic from the extremities of the failed link I is rerouted. Finally, the non-negativity of variables is expressed by constraints (3.11f). Note that, as RR^- , path generation for LR is polynomial.

3.4.4 Path Diversity - PD

Contrary to the above strategies, PD is a protection strategy. The only requirement is that in case of failure, enough bandwidth is remaining to route the traffic of every demand. The formulation of the dimensioning problem for path diversity, denoted $P_{PD}(P)$, is as follows:

$$\min C = \sum_{e \in E} \xi_e y_e^0 \quad (3.12a)$$

$$y_e^0 - \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \geq 0, \quad e \in E \quad (3.12b)$$

$$\sum_{p \in P_d} x_{dp}^0 \geq h_d^0, \quad d \in D \quad (3.12c)$$

$$y_e^0 - \sum_{d \in D} \sum_{p \in P_d, s \notin p} \delta_{edp} x_{dp}^0 \geq 0, \quad s \in S, e \in E \setminus s \quad (3.12d)$$

$$\sum_{p \in P_d, s \notin p} x_{dp}^0 \geq h_d^s, \quad d \in D, s \in S \quad (3.12e)$$

$$x^0, x^s, y^0 \text{ continuous and nonnegative} \quad (3.12f)$$

Constraints (3.12b) (respectively (3.12d)) express capacity constraints in the nominal (respectively failure) state. Constraints (3.12c) ensure the satisfaction of traffic in the nominal state and constraints (3.12e) ensure the lowest traffic satisfaction rate for the failure scenario s (recall that we only consider single total link failures). Finally, the non-negativity of variables is expressed by constraints (3.12f).

Notice that PD is polynomial in case of single link failure scenarios, i.e. when every link can fail but one at a time. However, it becomes NP -hard in the general case of failure scenarios that admit simultaneous failures of multiple links (Tomaszewski et al. 2010).

3.4.5 Shared Robust Rerouting - ShRR

Contrary to the other mechanisms, ShRR makes a distinction between affected and unaffected demands after a failure. A traffic demand d is said to be affected by a failure state s if its surviving nominal path-flows (thinned or not) are not sufficient to carry the traffic volume assumed for this state, i.e., when $\sum_{p \in \bar{P}_d^s} x_{dp}^s + \sum_{p \in P_d^s} x_{dp}^0 < h_d^s$. Otherwise, the demand is unaffected. The status (affected/unaffected) of demand $d \in D$ in failure state $s \in S$ is represented, by a binary variable $T_d^s, d \in D, s \in S$: $T_d^s = 1$ if d is an affected demand in s and $T_d^s = 0$ if d is not affected in s . In failure state $s \in S$, unaffected demands $d \in D^s$ can only thin their unaffected path-flows (concerning only paths in P_d^s) and affected demands $d \in \bar{D}^s$ can reroute the perturbed path-flows. The formulation of the dimensioning problem for path diversity, denoted $P_{ShRR}(P)$, is as follows:

$$\min C = \sum_{e \in E} \xi_e y_e^0 \quad (3.13a)$$

$$y_e^0 - \sum_{d \in D} \sum_{p \in P_d} \bar{\delta}_{edp} x_{dp}^0 \geq 0, \quad e \in E \quad (3.13b)$$

$$\sum_{p \in P_d} x_{dp}^0 \geq h_d^0, \quad d \in D, s \in S \quad (3.13c)$$

$$y_e^0 - \sum_{d \in D} \sum_{p \in P_d^s} \bar{\delta}_{edp} x_{dp}^0 - \sum_{d \in \bar{D}^s} \sum_{p \in P_d^s} \bar{\delta}_{edp} u_{dp}^s + \sum_{d \in \bar{D}^s} \sum_{p \in P_d^s} \bar{\delta}_{edp} x_{dp}^s \geq 0, \quad s \in S, e \in E \neq s \quad (3.13d)$$

$$\sum_{p \in P_d^s} (x_{dp}^0 - u_{dp}^s) + \sum_{p \in \bar{P}_d^s} x_{dp}^s \geq h_d^s, \quad d \in D, s \in S \quad (3.13e)$$

$$\sum_{p \in P_d^s} x_{dp}^0 + M(T_d^s - 1) < h_d^s, \quad s \in S \quad (3.13f)$$

$$\sum_{p \in P_d^s} x_{dp}^0 + h_d^s T_d^s \geq h_d^s, \quad s \in S \quad (3.13g)$$

$$x_{dp}^0 - u_{dp}^s \geq 0, \quad d \in D, s \in S, p \in P_d^s \quad (3.13h)$$

$$u_{dp}^s + M(T_d^s - 1) \leq 0, \quad d \in D, s \in S, p \in P_d^s \quad (3.13i)$$

$$x^0, x^s, u^s, y^0 \text{ continuous and nonnegative} \quad (3.13j)$$

Constraints (3.13b) (respectively (3.13d)) express capacity constraints in the nominal (respectively failure) state. Constraints (3.13f) and (3.13g) classify demands in affected and unaffected. Constraints (3.13c) ensure the satisfaction of traffic in the nominal state and constraints (3.13e) ensure the lowest traffic satisfaction rate in any link failure state. Then, constraints (3.13h) guarantee that decreasing the traffic volume over non-disrupted demand paths by more than the corresponding nominal traffic is not permitted and constraints (3.13i) ensure that only unaffected demands can release bandwidth. Finally, the non-negativity of variables is expressed by constraints (3.13j). Note that ShRR has not been solved to optimality using classic path generation algorithms because of the distinction between affected and unaffected demands.

3.5 Numerical results for all mechanisms

In this section, we present results of computational study illustrating the performances of the mechanisms presented in 3.4 and Global Rerouting. We also study the efficiency of the path generation algorithm compare to the other formulation of the same problems.

The undirected network instances used in our tests and listed in table 3.1 are taken from SNDlib (Orlowski et al. 2010).

Table 3.1 – Network description

Network	Nodes	Links	Demands
pdh	11	34	24
polska	12	18	66
nobel-us	14	21	91
nobel-germany	17	26	121

3.5.1 Dimensioning cost

In the following, we compare the Global Rerouting strategy (GR) and the presented strategies (Local Rerouting (LR), Restricted Rerouting with stub release (RR^+), Restricted Rerouting without stub release (RR^-) and Path Diversity (PD)) applied to the total single link failure case in terms of cost effectiveness.

For the experiments reported in this section we assume a uniform demand satisfaction ratio β , i.e. $h_d^s = \beta h_d^0$ for all $d \in D, s \in S$. In the comparisons, let C_{GR} denote the optimal value of the objective function (i.e., minimum cost of the link capacity) for GR, and $C_{LR}, C_{RR^+}, C_{RR^-}, C_{PD}$ the respective values for the remaining strategies. Certainly, the computed link capacities (and thus the link capacity cost) ensure the routing of all traffic demands in the nominal state and the guaranteed traffic restoration (specified by β) in all failure states.

As we already mentioned, the cost C_{GR} indicated by GR is not greater than any of the remaining cost, as GR assumes the least restricted flow restoration mechanisms. Hence, we can define the relative cost increase (in percent) for each strategy S (where S stands for LR, RR, WoR or PD) with respect to GR as $G_S = \frac{C_S - C_{GR}}{C_{GR}} * 100\%$. In the following, this quantity will be called relative gap. Note that, by definition, $G_{GR} = 0$.

Figures 3.1-3.3 show that Path Diversity is by far the less cost-efficient strategy with up to 87% gap. Note that PD seems not to be sensitive to variations of β , regarding to the gap with GR. LR and RR^- have slightly the same cost with a gap varying between 1% and 35% for all networks and all values of β . Finally, RR seems to be the most cost-efficient strategy for single total link failures (after GR) with a gap less than 2% for total traffic restoration, i.e. $\beta = 1$.

3.5.2 Efficiency of Path Generation

In sections 3.2, 3.3 and 3.4, a Path Generation algorithm intended to accelerate the CPU time was presented. From the LP presented above, we can

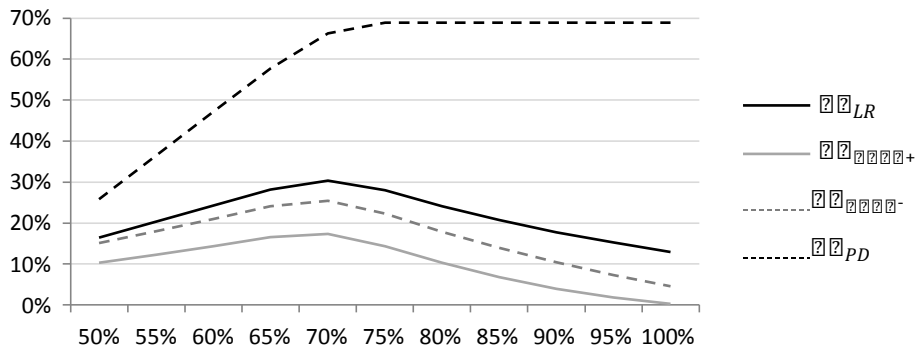


Figure 3.1 – Relative gap (in %) for all strategies for network polska.

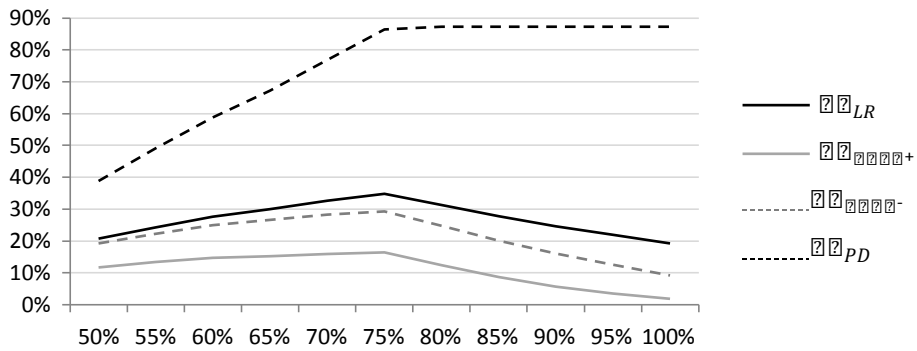


Figure 3.2 – Relative gap (in %) for all strategies for network nobel-us.

establish that the number of variables is the same for GR, RR⁺ and RR⁻ with $|E| + (1 + |S|) * \sum_{d \in D} |P_d|$ variables. The number of variables for LR is equal to $|E| + \sum_{d \in D} |P_d| + \sum_{l \in S} \sum_{d \in D} |P_d^l|$ while for PD it is limited to $|E| + \sum_{d \in D} |P_d|$. To have a clear sight of these numbers, let us take a look at them for the 4 networks.

Table 3.2 – Number of optimization variables for all networks, for all strategies.

Network	GR, RR ⁺ , RR ⁻	LR	PD
pdh	65 885 819	1 882 485	1 882 485
polksa	46 701	37 455	2 475
nobel-us	156 507	128 437	7 134
nobel-germany	324 215	224 875	12 033

Hence, the need for PG becomes clear as the number of variables might be very high depending on the network instance. We can see from above that the total number of paths for a highly dense network such as *pdh* (with only 11 nodes) is up to 2 million while this number is highly reduced for a sparse network with more nodes such as *polksa*. Notice that the network used in this section are small networks. Let us study the efficiency of the arc-path formulation with and without path generation for the GR and RR⁺ strategies. As the compact formulation of GR is presented, it is added to the following results.

Tables 3.3 and 3.4 clearly shows that Path Generation is very interesting to solve multicommodity flow problems when possible. Both tables

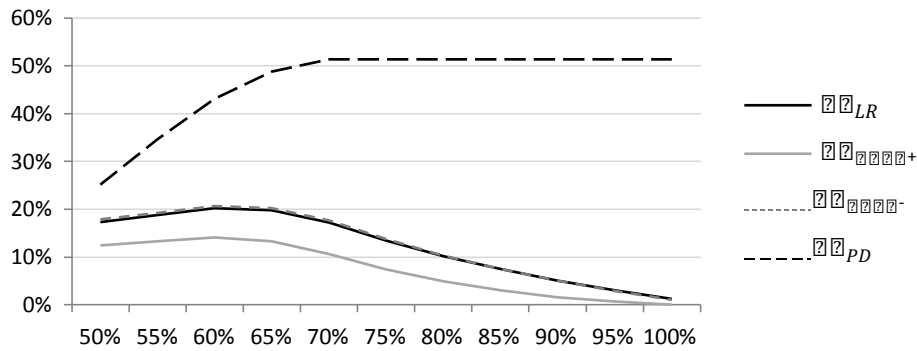


Figure 3.3 – Relative gap (in %) for all strategies for network nobel-germany.

shows that using PG improves the computation time for these networks. It is especially clear for GR. More importantly, the number of paths used, and so the number of variables, is drastically reduced (more than 90% less paths) with up to 99.95% of reduction for network *pdh* for GR. Path Generation will be used for the strategies presented in the following chapters. Note that the pricing problem for RR^+ is NP -hard and this can be seen for network *pdh*, where the computation time using PG is up to 23 hours while the computation of the compact formulation is 17 times faster.

Table 3.3 – Study of Path Generation for GR.

Network	arc-node	arc-path		arc-path + PG	
	time (s)	time (s)	nb paths	time (s)	nb paths
pdh	9	442	1 882 451	7	98
polska	4	1	2 457	1	184
nobel-us	38	6	7 113	5	245
nobel-germany	18	16	12 007	19	439

Table 3.4 – Study of Path Generation for RR^+ .

Network	arc-path		arc-path + PG	
	time (s)	nb paths	time (s)	nb paths
pdh	4 874	1 882 451	83 607	721
polska	10	2 457	2	403
nobel-us	74	7 113	6	706
nobel-germany	140	12 007	43	837

Part II

Flow Thinning

This part is dedicated to the *Flow-Thinning* strategy (FTS). Chapter 4 introduces the principle of the strategy and gives illustration through examples. Then, we present the basic FT optimization problem (FTOP) followed by a discussion on its computational complexity. FTOP is NP-hard at least for two-links failure scenarios. Thus, we study the separation problem and propose an original Path Generation algorithm using a binary LP. Next, in Chapter 5, we study the efficiency of FTS. We first compare the dimensioning cost of the network using FTS with other strategies such as GR. This study is followed by a study of a lower bound formulation of the problem. In the last chapter (Chapter 6), we briefly present an affine version of FTS that reduces the number of failure states for which the strategy needs to be optimized. Next, we deal with implementation issues of FTS and its affine variant. Finally, we present an extension of FTS that allows for thickening and argue why this method cannot be implemented in the current form.

Flow Thinning - Introduction and Optimization Problem

Contents

4.1	Introduction to Flow Thinning	42
4.2	FTOP—the basic FT optimization problem	44
4.2.1	Notations	44
4.2.2	Formulation of FTOP	46
4.2.3	Complexity of FTOP and global rerouting	47
4.3	Dual problem and dual separation	48
4.3.1	Dual problem formulation	48
4.3.2	Dual separation	49
4.4	Path generation	51
4.4.1	Pricing problem	52
4.4.2	Binary formulation of PP	53
4.4.3	PG algorithm	54
4.5	Computational complexity of PP, DSP, and special cases	55
4.6	Single-link failures	57
4.6.1	Single-link failures – compact FTOP formulation	57
4.6.2	Single-link failures - proof of Lemma 3	59
4.7	Single-node failures	60
4.7.1	Single-node failures – the pricing problem	60
4.7.2	Single-node failures – compact FTOP formulation	63

Flow thinning (FTS) is a protection strategy ¹, designed to manage multiple partial link failures. In this chapter, we first introduce this strategy with some illustrations. Then, we present the mathematical formulation of the dimensioning problem using FTS, denoted FTOP. Before studying the path generation algorithm, we present the dual separation problem. Next, we analyze the complexity of the pricing problem. Finally, we study two special cases: single partial link failures and single partial node failures.

¹We categorize FTS in the group of protection strategies because the main feature of FTS is not creating new paths which is proper to restoration methods

4.1 Introduction to Flow Thinning

The flow-thinning protection strategy (FTS, Pióro et al. (2013b)), designed for Multiple Partial Link Failures (MPLF), is inspired by the idea of flow adjustment proposed in *ShRR* (Fundo et al. 2013) and *path diversity* (Orlowski and Pióro 2012). The main idea behind FTS is as follows:

- Following the idea of *path diversity*, we assume that each demand is in general routed over several paths, not necessarily disjoint, with over-dimensioned path-flows to ensure an assumed level of survivability.
- Unlike conventional end-to-end restoration strategies, FTS guarantees survivability solely by releasing a fraction of flow on the affected routing paths as *ShRR*. Consequently, no flow is rerouted or increased. It is important that this thinning of path-flows is done in an appropriate failure-state dependent way.

To give a more clear idea of how FTS works, we wish to emphasize a few points. First, note that no distinction is made between capacity devoted to flow routing in the normal state and those devoted to protection, and all available link capacity is used jointly as a common pool of resources to realize demand flows in failure states. Hence, the selected routing paths carry the flows dimensioned so that the total flow realized on the demand's paths can in general be greater than the nominal traffic. The most important feature of FTS is handling partial failures without any flow rerouting/restoration at all. In other words, no nominal paths increase their flows in failure states and no paths besides the nominal paths are used for handling failure states. Therefore, the proposed approach results in using a sort of limited dynamic routing, adapting to the network states. To summarize, for each demand there is a fixed set of nominal routing paths carrying nominal flows. In a failure state in general only a part of the total nominal demand flow will be realized on these paths, depending on the available capacity and a given demand restoration ratio. Consequently, each affected nominal routing path can only release a fraction of its flow according to the failure that has occurred, and no new (re)routing paths are allowed. Note that FTS becomes equivalent to PD for total link failure situations.

Example 4.1 Consider a network with 4 nodes A , B , C and D and 5 undirected links (A, B) , (A, C) , (B, C) , (B, D) and (C, D) . The unit capacity cost of all five links is equal to 1. We consider two demands d_1 and d_2 (each with demand volume equal to 1) between nodes A and B for d_1 , and A and D for d_2 . We consider partial single-link failure scenarios in which each link can fail, but once at a time. A failure consists in a reduction of 50% of the reference capacity. We show below what is the result of the minimum link cost network when FTS is used to recover from partial single link failures. The obtained solution gives a capacity equal to $\frac{3}{2}$ for link (A, B) , $\frac{5}{4}$ for link (A, C) , 1 for link (C, D) , $\frac{1}{2}$ for link (B, D) and $\frac{1}{4}$ for link (B, C) , which gives a global cost equal to $\frac{9}{2}$. In the nominal state, as depicted in Figure 4.1,

Demand d_1 is realized on 2 paths: $p_1 = \{(A, B)\}$, $p_2 = \{(A, C), (C, B)\}$ and demand d_2 is routed on 2 paths: $p_3 = \{(A, B), (B, D)\}$, $p_4 = \{(A, C), (C, D)\}$.

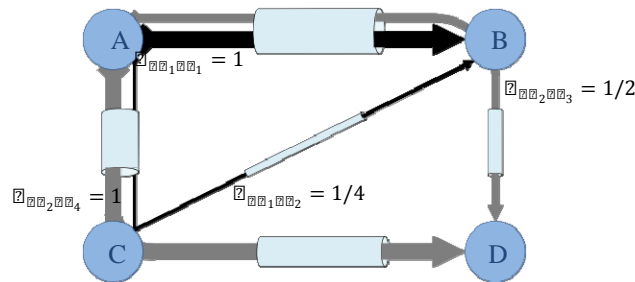


Figure 4.1 – FTS - Nominal State.

Figure 4.2 represents the failure of link (C, D). In this scenario, the capacity of link (C, D) decreases down to $\frac{1}{2}$. Hence, the basic solution is to decrease the bandwidth of path p_4 (the only path routed through (C, D)) by 50%. Hence, capacity constraints are respected and all traffic requirements are met.

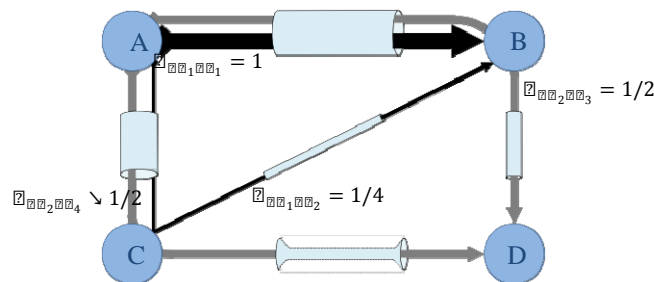


Figure 4.2 – FTS - Failure of link (C, D).

Another failure scenario is presented on Figure 4.3, concerning the failure of link (A, B). This failure scenario is slightly more complex as two paths p_1 and p_3 are routed through this link and due to the failure, the capacity of (A, B) is now equal to $\frac{3}{4}$ and the sum of the bandwidth of the paths routed through (A, B) is equal to $\frac{6}{4} > \frac{3}{4}$. FTS will manage this failure by releasing all bandwidth of path p_3 , and $\frac{1}{4}$ of bandwidth of path p_1 . Thus, the sum of the bandwidth of the paths routed through (A, B) is equal to the bandwidth of path p_1 i.e. $\frac{3}{4}$ which is equal to the capacity of link (A, B). Furthermore, all traffic requirements are met.

Note that, as FTS is a protection strategy, the total bandwidth associated to the demands is always sufficient to route the traffic. However, before FTS manages the failure, strong congestion may appear on the paths routed through the disturbed link.

This example considered failures of 50% of the reference capacities. Figure 4.4 represents the evolution of the optimal dimensioning cost of the network varying the failure coefficient from 0% (nominal state) to 100% (total link failures).

We notice that the optimal cost of $C_{FT} = 7$ obtained for total link failures (failure coefficient of 100%) corresponds to path diversity.

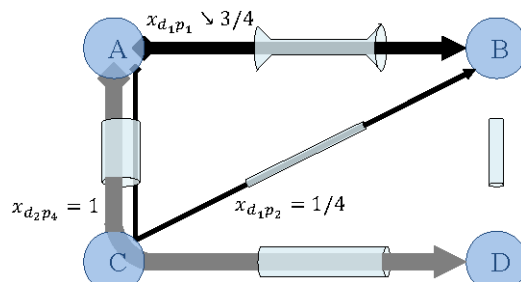


Figure 4-3- FTS- Failure of link (A, B).

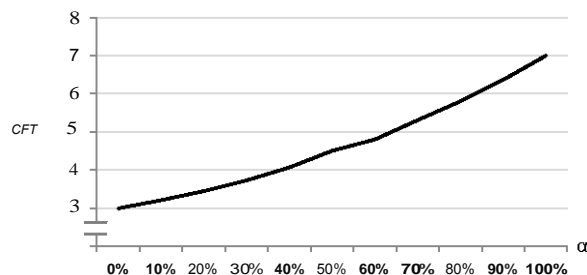


Figure 4-4 – Evolution of the optimal cast of the network using FTS depending on the failure coefficient.

4.2 FTOP-THE BASIC FT OPTIMIZATION PROBLEM

The basic problem considered in this section is referred to as *flow-thinning optimization problem* (FTOP) and is as follows. We minimize the cost of link capacity assuming that in the normal state of network operation all demand volumes are realized by means of (normal) path-flows. When the network is subject to a failure from a given set of failure states (we assume that a failure state consists of multiple partial failures of links) then the demand volumes, possibly reduced, are realized for the duration of the failure state by appropriate thinning of the normal flows. The detailed formulation of FTOP will be given in Subsection (4.2.2).

4.2.:1 Notations

FTOP requires the use of the following notations:

V	set of routers
E	set of links, directed or undirected, represented by a pair $\{v, w\}$ of some nodes $v, w \in V$
D	set of demands associated with an unordered pair of nodes $\{o(d), t(d)\}$. For the sake of simplicity we call $o(d)$ a source node and $t(d)$ a termination node.
S	set of failure scenario equal to all possible link failure. So represents the nominal state.
	set of links incoming to node $v \in V$
	set of links outgoing from node $v \in V$
	non-negative unit capacity cost of link $e \in E$, seen as a parameter

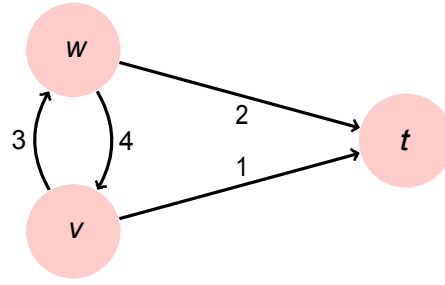


Figure 4.5 – A simple network example

h_d^0	volume that has to be sent between $o(d)$ and $t(d)$ for demand $d \in D$, seen as a parameter
β^s	demand reduction coefficient vector in state $s \in S$, $\beta^s := (\beta_d^s, d \in D)$
h_d^s	traffic volume of demand d to be carried in state s
y^e	capacity reservation for edge $e \in E$, which is an optimization variable
α^s	link availability coefficient vector in state $s \in S$, $\alpha^s := (\alpha_e^s, e \in E)$ with α_e^s between 0 and 1.
y_e^0	capacity available in the nominal state for edge $e \in E$.
y_e^s	capacity (possibly decreased) available in failure state $s \in S$ for edge $e \in E$. This capacity is equal to $y_e^s := \alpha_e^s y_e^0$
P_d	set of nominal paths (called the path-list), parameters in the EFR-P formulation
P	set of admissible paths, $P := \bigcup_{d \in D} P_d$. Remind that we consider only elementary paths.
P_d^s	set of paths that are unaffected in failure state $s \in S$
\hat{P}_d	maximum path-lists containing all the elementary paths from $o(d)$ to $t(d)$, $\hat{P}_d := \bigcup_{d \in D} \hat{P}_d$
δ_{edp}	link-path incidence coefficients $(\delta_{edp}, e \in E, d \in D, p \in P_d)$. It takes value $\delta_{edp} = 1$ if path $p \in P_d$ traverses link $e \in E$, and $\delta_{edp} = 0$ otherwise.
x_{dp}^0	nominal path-flow associated to path $p \in P_d$ for demand $d \in D$
x_{dp}^s	nominal path-flows thinned associated to path $p \in P_d$ for demand $d \in D$ for state $s \in S$. As only thinned nominal flows are allowed, $x_{dp}^s \leq x_{dp}^0$, $s \in S, d \in D, p \in P_d$.
E_s	set of links that are not fully available in state $s \in S$, $E_s := \{e \in E : \alpha_e^s < 1\}$
S_e	set of states in which link $e \in E$ is not fully available $S_e := \{s \in S : \alpha_e^s < 1\}$

An illustration of the above notations is given in Example 4.2.

Example 4.2 Consider the network depicted in Figure 4.5. It is a 3-node, 4-link directed network with $V = \{v, w, t\}$ and $E = \{1, 2, 3, 4\}$. The unit link costs are all equal to 1: $\xi_e = 1, e \in E$. There are two demands ($D = \{1, 2\}$), demand $d = 1$ from node v to node t , and demand $d = 2$ from node w to node t , with the nominal traffic volumes $h_1^0 = h_2^0 = 1$. Each demand has two admissible paths: $P_1 = \{p_{11}, p_{12}\}$ where $p_{11} = \{1\}, p_{12} = \{3, 2\}$, and $P_2 = \{p_{21}, p_{22}\}$ where $p_{21} = \{2\}, p_{22} = \{4, 1\}$. We assume that links 3 and 4 are always fully available while links 1 and 2 can fail, but only one at a time. When link $e = 1$ or $e = 2$ fails, it loses half of its capacity. Thus, there are three availability states $S = \{1, 2, 3\}$ and we assume

that in each state the nominal traffic volumes are to be realized so that the demand coefficients are all equal to 1 ($\beta^s = 1, d \in D, s \in S$). State $s = 3$ is a maximum state ($S_0 = \{3\}$) with all links^d fully available: $\alpha_e^3 = 1, e \in E$. In state $s = 1$, link $e = 1$ fails: $\alpha_1^1 = \frac{1}{2}, \alpha_e^1 = 1, e = 2, 3, 4$. Symmetrically, in state $s = 2$, link $e = 2$ fails: $\alpha_2^2 = 1/2, \alpha_e^2 = 1, e = 1, 3, 4$.

A feasible solution is as follows. The nominal capacity of the four tunnels is defined as

$$x_{11}^0 = 1, x_{12}^0 = 1/3, x_{21}^0 = 1, x_{22}^0 = 1/3$$

and hence the (minimal) nominal capacity of the links that realizes the nominal flows is equal to $y_1^0 = y_2^0 = 4/3, y_3^0 = y_4^0 = 1/3$. The cost of the network is then equal to $C = 10/3$. The path flows (i.e., the capacity of the tunnels) in the three considered availability states is as follows:

$$x_{11}^1 = 2/3, x_{12}^1 = 1/3, x_{21}^1 = 1, x_{22}^1 = 0$$

$$x_{11}^2 = 1, x_{12}^2 = 0, x_{21}^2 = 2/3, x_{22}^2 = 1/3$$

$$x_{11}^3 = 1, x_{12}^3 = 0, x_{21}^3 = 1, x_{22}^3 = 0.$$

Note that the nominal capacity of the tunnels could carry the traffic volume $h = 4/3$ simultaneously for both demands, i.e., the volume greater than the nominal volume $h^0 = 1$.

4.2.2 Formulation of FTOP

In essence, FTOP consists in minimizing the cost of nominal link capacity that supports a set of nominal tunnel capacities (path-flows) that are subject to thinning in each of the considered network states. The FTOP instance corresponding to a given set of admissible path-lists $P_d, d \in D$ is denoted by $P_{FT}(P)$ (where $P = \bigcup_{d \in D} P_d$) and represented by the following link-path linear programming (LP) formulation involving variables $x_{dp}^0 (d \in D, p \in P_d), x_{dp}^s (d \in D, p \in P_d, s \in S)$ and $y_e^0 (e \in E)$:

$$P_{FT}(P): \quad C(P) = \min \sum_{e \in E} \xi_e y_e^0 \quad (4.1a)$$

$$[\pi_e^0 \geq 0] \quad \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \leq y_e^0, \quad e \in E \quad (4.1b)$$

$$[\lambda_d^s \geq 0] \quad \sum_{p \in P_d} x_{dp}^s \geq h_d^s, \quad d \in D, s \in S \quad (4.1c)$$

$$[\pi_e^s \geq 0] \quad \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^s \leq \alpha_e^s y_e^0, \quad e \in E, s \in S_e \quad (4.1d)$$

$$[\sigma_{dp}^s \geq 0] \quad x_{dp}^s \leq x_{dp}^0, \quad d \in D, p \in P_d, s \in S \quad (4.1e)$$

$$x^0, x^s, y^0 \geq 0 \text{ and continuous} \quad (4.1f)$$

(where the quantities in the square brackets on the left-hand side are dual variables to be used later).

Objective (4.1a) minimizes the overall cost of links. The first capacity constraint (4.1b) makes sure that the nominal link loads do not exceed the nominal link capacities. Next, the demand constraint (4.1c) assures that in each state $s \in S$, the thinned nominal flows are sufficient to realize

the volume of each demand $d \in D$ assumed for this state. Then, the second capacity constraint (4.1d) assures that in each state $s \in S$, the available capacity of each link $e \in E_s$ is not exceeded. Finally, constraint (4.1e) assures thinning. Note that constraint (4.1d) is, for each link $e \in E$, written down only for $s \in S_e$, and not for all $s \in S$ because for any state $s \in S \not\subseteq S_e$ (in which the considered link e is fully available, $\alpha_e^s = 1$) such a constraint would be redundant—it is implied by (4.1b), as the link load $\sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^s$ does not exceed its nominal load $\sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0$ due to constraint (4.1e). The optimal value of the cost (4.1a) of $P_{FT}(P)$ is denoted by $C(P)$. Note also that formulation of FTOP does not explicitly involve dependence of β^s on α^s ($s \in S$)—this is done while preprocessing h^s .

Observe that our assumption of admitting only elementary paths on the path-lists made in Section 4.2.1 is not limiting. If we admitted non-elementary paths in the overall path-list P , then eliminating loops in any feasible solution of $P_{FT}(P)$ would never increase $C(P)$. This is true because if path p^t is an elementary version of path $p \in P_d$, then adding p^t to P_d , putting $x_{dp^t}^0 := x_{dp}^0$ and $x_{dp^t}^s := x_{dp}^s$, $s \in S$, and finally setting x_{dp}^0 and x_{dp}^s , $s \in S$, to 0 would lead to a feasible solution of (4.1) which does not require the capacity reserved on the links in $p \not\subseteq p^t$. The last statement also implies that non-elementary paths will not appear in optimal solutions of (4.1) when $\xi_e > 0$, $e \in E$. An illustration of the above considerations is given in Example 4.3.

Example 4.3 *For the network depicted in Figure 4.5, the solution given in Example 4.2 is feasible and optimal for 4.1. Note that the flows x^0, x^1, x^2 of Example 4.2 are unique at optimality while x^3 are not. For example, the flows $x^3 = x^0$ are also optimal. If we change the unit costs of links 3 and 4 to M , where M is a large number ($M \gg 1$), then the optimal solution of FTOP will change. The optimal cost will increase to $C = 4$ and the optimal tunnel capacities will be as follows:*

$$\begin{aligned} x_{11}^0 &= 2, x_{12}^0 = 0, x_{21}^0 = 2, x_{22}^0 = 0 \\ y_1^0 &= 2, y_2^0 = 2, y_3^0 = 0, y_4^0 = 0 \\ x_{11}^1 &= 1, x_{12}^1 = 0, x_{21}^1 = 1, x_{22}^1 = 0 \\ x_{11}^2 &= 1, x_{12}^2 = 0, x_{21}^2 = 1, x_{22}^2 = 1 \\ 1 &\leq x_{11}^3 \leq 2, x_{12}^3 = 0, 1 \leq x_{21}^3 \leq 2, x_{22}^3 = 0. \end{aligned}$$

4.2.3 Complexity of FTOP and global rerouting

In the sequel, by FTOP we will always mean the instances of problem (4.1) with the full sets of admissible paths \hat{P} , i.e., $P(\hat{P})$. (Instances of $P(P)$ with restricted sets of admissible paths $P \subsetneq \hat{P}$ will be denoted by $FTOP^<$.) We note that problem FTOP is NP -hard (already for a polynomial number of states) since it contains, as a special case, the corresponding PD optimization problem whose NP -hardness was proven by (Tomaszewski et al. 2010) for multiple total link failure scenarios. More precisely, the PD optimization problem (and thus FTOP) is NP -hard already for undirected graphs with $|V| = L + 1$ nodes, $|E| = 2L$ links, one demand ($|D| = 1$), and $|S| = L$ failure states (each such state consists of simultaneous failures of

a certain subset of links). Consequently, FTOP cannot be formulated as a compact linear program (unless $P = NP$). Yet, whenever admissible paths in P are predefined and $|P|$ is polynomial with the size of the network, then problem $P(P)$ becomes polynomial as (4.1) becomes compact (provided $|S|$ is polynomial).

Difficulty of FTOP stems from the thinning constraint (4.1e) which introduces dependence between the nominal flows and the state-dependent flows. In fact, when constraints (4.1e), together with the associated nominal path-flow variables x^0 and constraints (4.1b), are deleted from formulation (4.1), then the resulting optimization problem becomes polynomial and corresponds to the so called global rerouting (GR) flow restoration strategy, also referred to as unrestricted flow reconfiguration or dynamic routing, see (Pióro and Medhi 2004b), (Orlowski and Pióro 2012). In GR, the flows x^s realizing the demands in different states are independent of each other and are only coupled by the capacity constraints (4.1d). This can be interpreted as if for each state $s \in S$ the flows x^s were established from scratch in the currently available capacity $\alpha_e^s y_e$, $e \in E$. We note that the resulting GR optimization problem (4.1a), (4.1c), (4.1d) is polynomial also for the full set of admissible paths \hat{P} , as, because of the independence of flow variables x^s , $s \in S$, it can be easily formulated as a compact linear program using the node-link notation. GR is the most flexible protection mechanism we can think of and therefore it provides a lower bound for the network cost (4.1a). Yet, for the network considered in Example 4.2, the optimal FTOP solution happens to be optimal for GR as well.

A natural extension of FTOP that allows for limited tunnel thickening is discussed in Chapter 6.

4.3 Dual problem and dual separation

In this section we will formulate the problem dual to the primal problem $P_{FT}(P)$ defined by (4.1) and discuss the separation problem corresponding to the dual polyhedron.

4.3.1 Dual problem formulation

Problem $D_{FT}(P)$ dual to $P_{FT}(P)$ for a given set of admissible paths P is as follows (for derivation see Pióro and Medhi (2004b)):

$$D_{FT}(P): \quad W(P) = \max \sum_{d \in D} \sum_{s \in S} h_d^s \lambda_d^s \quad (4.2a)$$

$$\pi_e^0 + \sum_{s \in S_e} \alpha_e^s \pi_e^s \leq \xi_e, \quad e \in E \quad (4.2b)$$

$$\sum_{s \in S} \sigma_{dp}^s \leq \sum_{e \in E} \delta_{edp} \pi_e^0, \quad d \in D, p \in P_d \quad (4.2c)$$

$$\lambda_d^s \leq \sigma_{dp}^s + \sum_{e \in E_s} \delta_{edp} \pi_e^s, \quad s \in S, d \in D, p \in P_d \quad (4.2d)$$

$$\pi, \lambda, \sigma \geq 0 \text{ and continuous.} \quad (4.2e)$$

Let $\Pi(P)$ denote the dual polyhedron of all feasible solutions (π, λ, σ) of $D_{FT}(P)$, where $\pi = (\pi_e^0, \pi_e^s, e \in E, s \in S_e)$, $\lambda = (\lambda_d^s, d \in D, s \in S)$,

$\sigma = (\sigma_{dp}^s, d \in D, p \in P_d, s \in S)$. We treat the dual solutions (π, λ, σ) as vectors in the \mathbb{R}^N space of dimension $N = |E| + \sum_{e \in E} |S_e| + |D| \cdot |S| + (\sum_{d \in D} |P_d|) \cdot |S|$, and observe that $\Pi(P)$ is a fully dimensional subset of \mathbb{R}^N . The projection of polyhedron $\Pi(P)$ onto the (π, λ) space \mathbb{R}^n , where $n = |E| + \sum_{e \in E} |S_e| + |D| \cdot |S|$, is defined as $\Omega(P) := \{(\pi, \lambda) \in \mathbb{R}^n : \exists \sigma \in \mathbb{R}^{N-n}, (\pi, \lambda, \sigma) \in \Pi(P)\}$. Polyhedron $\Omega(P)$ is also fully dimensional.

Assume that the linear program $P_{FT}(P)$, where $P \subseteq \hat{r}$, formulated in (4.1) is feasible. Then, due to the strong duality property (Lasdon 1970, Minoux 1986), the optimal objective $W(P)$ of the corresponding dual problem $D(P)$ formulated in (4.2) is equal to the optimal primal objective $C(P)$ of $P_{FT}(P)$, i.e., $W(P) = C(P)$. Moreover, $P \subseteq \hat{r}$ implies that polyhedron $\Pi(P)$ contains polyhedron $\Pi(\hat{P})$, $\Pi(P) \supseteq \Pi(\hat{P})$, so that $W(P) \geq W(\hat{P})$ and hence $W(P)$ is actually an upper bound on the optimal objective $C(\hat{r})$ of $P_{FT}(\hat{r})$, i.e., of FTOP.

Certainly, problem $D_{DP}(\hat{r})$ represented by formulation (4.2) is NP -hard since it is an LP dual of an NP -hard LP problem $P_{PD}(\hat{r})$ represented by formulation (4.1).

4.3.2 Dual separation

In the sequel, the dual polyhedrons of FTOP, i.e., $\Pi(\hat{P})$ and $\Omega(\hat{P})$, will be simply denoted by Π and Ω , respectively. The *dual separation problem* for FTOP is the separation problem for polyhedron Ω formulated as follows (see Grotschel et al. (1988), Nemhauser and Wolsey (1988)):

Dual separation problem (DSP): Given an arbitrary vector $(\pi, \lambda) \in \mathbb{R}^n$, determine whether $(\pi, \lambda) \in \Omega$, and if not, find a hyperplane in \mathbb{R}^n that separates (π, λ) from the polyhedron Ω .

Let us first notice that if $(\pi, \lambda) \notin \mathbb{R}_+^n$ then it is separated from Ω trivially, by one of the nonnegativity inequalities in (4.2e). Hence, we can assume that $(\pi, \lambda) \in \mathbb{R}^n$. Then, it is easy to check whether π fulfills all constraints in (4.2b). If it does not, then (π, λ) is separated from Ω by one of the unfulfilled inequalities in (4.2b), i.e., by one of the hyperplanes $\pi_e^0 + \sum_{s \in S_e} \alpha_e^s \pi_e^s = \xi_e$.

So now we can assume that $(\pi, \lambda) \in \mathbb{R}_+^n$ and π is feasible with respect to (4.2b). Under this assumption $(\pi, \lambda) \notin \Omega$ if, and only if, there exists a demand $d \in D$ and a path $q \in \hat{r}_d$ such that for any $\sigma_{dq}^s \geq 0, s \in S$, (π, λ) does not fulfill constraints (4.2c) or (4.2d) for path q . Before formulating a condition for such a path q (in Proposition 1), we introduce the following useful definitions.

For a given path $q \in \hat{r}_d$, the quantity $|q|^0 := \sum_{e \in E} \delta_{edq} \pi_e^0$ is called the nominal dual length of q , while the quantities $|q|^s := \sum_{e \in E_s} \delta_{edq} \pi_e^s, s \in S$, are called the state-dependent dual lengths of path q , where as before, $\delta_{edq}, e \in E$, are the link-path incidence coefficients characterizing path q . Note that, by definition, $E_s \cap q = \emptyset$ implies $|q|^s = 0$, and in particular $|q|^s = 0$ for all $q \in \hat{r}_d$ and $s \in S_0$. With these two notions we can rewrite

constraints (4.2c) and (4.2d) in a more concise way:

$$\sum_{s \in S} \sigma_{dq}^s \leq |q|^0 \quad (4.3a)$$

$$\lambda_d^s \leq \sigma_{dq}^s + |q|^s, \quad s \in S \quad (4.3b)$$

$$\sigma_{dq}^s \geq 0, \quad s \in S. \quad (4.3c)$$

Now we can define the *total dual length* of path $q \in \hat{r}_d$, a crucial quantity specified for a given vector $(\pi, \lambda) \in \mathbb{R}_+^n$:

$$q) := |q|^0 + \sum_{s \in S_q(\pi, \lambda)} (|q|^s - \lambda_d^s), \quad (4.4)$$

where

$$S_q(\pi, \lambda) := \{s \in S : |q|^s < \lambda_d^s\}. \quad (4.5)$$

Proposition 1 Consider a given vector $(\pi, \lambda) \in \mathbb{R}_+^n$ with π fulfilling constraints (4.2b). Then $(\pi, \lambda) \notin \Omega$ if, and only if, there exists a demand $d \in D$ and a path $q \in \hat{r}_d$ such that

$$q) < 0. \quad (4.6)$$

If this is the case, the inequality reverse to (4.6), that is,

$$\sum_{e \in E} \bar{\delta}_{edq} \pi_e^0 + \sum_{s \in S_q(\pi, \lambda)} \left(\sum_{e \in E_s} \bar{\delta}_{edq} \pi_e^s - \lambda_d^s \right) \geq 0, \quad (4.7)$$

is not fulfilled by (π, λ) and thus the hyperplane $\sum_{e \in E} \bar{\delta}_{edq} \pi_e^0 + \sum_{s \in S_q(\pi, \lambda)} (\sum_{e \in E_s} \bar{\delta}_{edq} \pi_e^s - \lambda_d^s) = 0$ separates (π, λ) from Ω .

Proof 4.1 The proof of Proposition 1 is as follows. Consider vector $(\pi, \lambda) \in \mathbb{R}_+^n$ with π fulfilling constraints (4.2b) and a given path $q \in \hat{r}_d$. We will show that system of inequalities (4.3) is infeasible if, and only if, inequality (4.6) holds. Clearly, this implies that (4.7) separates (π, λ) from Ω . Suppose $q) \geq 0$, that is, $|q|^0 \geq \sum_{s \in S_q(\pi, \lambda)} (\lambda_d^s - |q|^s)$. Then, obviously,

$$\sigma_{dq}^s := \begin{cases} \lambda_d^s - |q|^s, & \text{if } s \in S_q(\pi, \lambda) \\ 0, & \text{if } s \in S \setminus S_q(\pi, \lambda) \end{cases} \quad (4.8)$$

$(\sigma_{dq}^s, \lambda_d^s)$ is a feasible solution of the system of inequalities (4.3). Now suppose $q) < 0$, that is,

$$\sum_{s \in S_q(\pi, \lambda)} (\lambda_d^s - |q|^s) > |q|^0. \quad (4.9)$$

Then, assuming feasibility of (4.3), due to (4.5), (4.3b), (4.3c), (4.3a), and (4.9), respectively, the following sequence of inequalities must hold:

$$0 \leq \sum_{s \in S_q(\pi, \lambda)} (\lambda_d^s - |q|^s) \leq \sum_{s \in S_q(\pi, \lambda)} \sigma_{dq}^s \leq \sum_{s \in S} \sigma_{dq}^s \leq |q|^0 < \sum_{s \in S_q(\pi, \lambda)} (\lambda_d^s - |q|^s). \quad (4.10)$$

This is a contradiction since in (4.10) the second term is strictly less than the last term but they are identical. Thus, system (4.3) is infeasible.

Proposition 1 states that $(\pi, \lambda) \in \Omega$ if, and only if, (i) $(\pi, \lambda) \in \mathbb{R}_+^n$, (ii) π fulfills (4.2b), and (iii) for each $d \in D$, $q \in \hat{\mathcal{P}}_d$, the total dual length of q with respect to (π, λ) is nonnegative, i.e., $q \geq 0$. The last condition, however, cannot be used as an inequality characterizing polyhedron Ω since its form (4.7) depends on a particular vector (π, λ) through $S_q(\pi, \lambda)$. Still, it directly implies such inequalities.

The explicit characterization of the projected polyhedron Ω is as follows:

$$\pi_e^0 + \sum_{s \in S_e} \alpha_e^s \pi_e^s \leq \xi_e, \quad e \in E \quad (4.11a)$$

$$\sum_{e \in E} \delta_{edp} \pi_e^0 \geq \sum_{s \in S^t} (\lambda_d^s - \sum_{e \in E_s} \delta_{edp} \pi_e^s), \quad d \in D, p \in \hat{P}_d, S^t \subseteq S \quad (4.11b)$$

$$\pi \geq 0, \lambda \geq 0. \quad (4.11c)$$

The above characterization follows from the fact that for a fixed path $q \in \hat{\mathcal{P}}_d$, the quantity on the right hand side of (4.11b) computed for any given $(\pi, \lambda) \in \mathbb{R}_+^n$ attains its maximum for $S^t = S_q(\pi, \lambda)$. This means that for the given path, the proper inequality, i.e., inequality (4.7), dominates the remaining inequalities in (4.11b) for each $(\pi, \lambda) \in \mathbb{R}_+^n$. Clearly, characterization (4.11) can be easily obtained with Fourier-Motzkin elimination (see (Motzkin 1951)).

Finally, we observe that by equivalence of optimization and separation in linear programming, see (Grotschel et al. 1988, Nemhauser and Wolsey 1988), DSP is NP -hard since problem $D(\hat{\mathcal{P}})$ specified in (4.2) is NP -hard as a problem dual to FTOP which is NP -hard (already for polynomially bounded number of states $|S|$, see Section 4.2.3). This fact is reflected by the difficulty of finding a path q violating inequality (4.7). We will come back to the complexity issues related to FTOP and DSP in Section 4.5.

4.4 Path generation

The link-path LP formulation (4.1) of FTOP is non-compact because of the presence of exponentially many path-flow variables x (and, as a matter of fact, of exponentially many constraints (4.1e)) corresponding to all possible elementary paths included in the maximum admissible path-list \hat{P} . The formulation is potentially non-compact also because of possibly exponential number of states in the set S (since the number of all link availability coefficient vectors is in general exponential, equal to $K^{|E|}$ when each link has K levels of availability). To avoid such an additional intrinsic hardness of the problem, we assume that $|S|$ is polynomially bounded by the number of links.

Thus, in practice, formulation (4.1) of FTOP cannot be solved directly, as we are virtually not able to include all elementary routing paths in the path-lists. In order to consider all paths in $\hat{\mathcal{P}}$, we need to apply path generation (PG), see (Ahuja et al. 1993)—a classical technique in multicommodity flow networks related to column generation in linear programming, cf. (Lasdon 1970, Minoux 1986). With PG, starting from some initial path-lists P_d , $d \in D$, we iteratively generate new paths, one per demand, and add those paths that may improve the solution to the path-lists—in effect, we

are adding the corresponding path-flow variables to the problem formulation. As discussed below, this is done by solving an appropriate pricing problem using, as parameters, optimal dual variables, i.e., an optimal solution (λ^*, π^*) of the current problem $D_{FT}(P)$ dual to $P_{FT}(P)$.

The application of PG to FTOP is explained below.

4.4.1 Pricing problem

PG is based on the so called *pricing problem* (PP). In essence, PP is similar to DSP (in the sequel, by DSP we will mean its version implied by Proposition 1), but with two main differences:

- the point to be separated from polyhedron Ω is not just any vector $(\pi, \lambda) \in \mathbb{R}_+^n$ with π fulfilling (4.2b), but one of the optimal solutions $D_{FT}(P)$ for some $P \subseteq \hat{P}$
- in PP we need to find, for each $d \in D$, not just a path (if any) in \hat{P}_d which separates the current optimal dual solution (π, λ) of $D(P)$, but rather a path $q \in \hat{P}_d \not\subseteq P_d$ for which the constraints of Ω corresponding to the given d and q are *most violated* by (π, λ) . In the sequel, the problem of finding such a path, i.e., PP for this setting, will be denoted by $PP(P, \pi, \lambda, d)$.

In fact, the second property does not necessarily have to be obeyed in PG since any path $q \in \hat{P}_d$ that violates at least one of the constraints (4.7) provides a cut that decreases, in a valid way, the current polyhedron $\Omega(P)$. Still, the use of the most violated cuts implied by the second property can substantially speed up the path generation process, especially when finding the most violated cut is not substantially more time consuming than finding an arbitrary cut.

Consider a given dual optimal solution (π, λ) of $D_{FT}(P)$ and a given path $q \in \hat{P}_d$ for some $d \in D$. Let $D(q)$ denote the degree of violation of constraints (4.7) (for the considered d and q) by vector (π, λ) .

Proposition 2 $D_{FT}(q) = \max\{-q, 0\}$.

Proof 4.2 *The proof of Proposition 2 is as follows. Since (π, λ) is a solution of $D(P)$, $(\pi, \lambda) \in \mathbb{R}_+^n$ and hence, as already noticed when introducing characterization (4.11) of the projected dual polyhedron Ω , the equality*

$$\sum_{s \in S_q(\pi, \lambda)} (\lambda_d^s - \sum_{e \in E_s} \delta_{edq} \pi_e^s) = \max \left\{ \sum_{s \in S^t} (\lambda_d^s - \sum_{e \in E_s} \delta_{edq} \pi_e^s) : S^t \subseteq S \right\} \quad (4.12)$$

holds. Assume that for some $S^t \subseteq S$, the corresponding constraint (4.11b) is broken, i.e., $\sum_{e \in E} \delta_{edq} \pi_e^0 < \sum_{s \in S^t} (\lambda_d^s - \sum_{e \in E_s} \delta_{edq} \pi_e^s)$. Then, due to (4.12), also the constraint $\sum_{e \in E} \delta_{edq} \pi_e^0 < \sum_{s \in S_q(\pi, \lambda)} (\lambda_d^s - \sum_{e \in E_s} \delta_{edq} \pi_e^s)$ is broken, and, for the same reason, the degree of violation of the latter constraint, i.e., $D(q) = \sum_{s \in S_q(\pi, \lambda)} (\lambda_d^s - \sum_{e \in E_s} \delta_{edq} \pi_e^s) - \sum_{e \in E} \delta_{edq} \pi_e^0 = -q$ is the largest among all the constraints (4.11b) corresponding to path q .

Certainly, Proposition 2 is not surprising as it implies that the pricing problem $PP(P, \pi, \lambda, d)$ is equivalent to

$$\text{minimize } q) \text{ over } q \in \hat{P}_d, \quad (4.13)$$

i.e., to finding the most violated separating inequality (4.7). Clearly, when the solution q of (4.13) is negative, then for a given $d \in D$ $q \notin P_d$ and $D_{FT}(q) = -q$ is the largest degree of violation of constraints (4.7) by (π, λ) corresponding to a given $d \in D$.

Finally, let us note that the dual separation problem and the pricing problem are similar in that in both DSP and PP we need to find a path with a negative total dual length q .

4.4.2 Binary formulation of PP

The pricing problem formulated in (4.13) is difficult because of the particular form of the total dual length q . Nevertheless, the problem can be stated as a binary programming problem by means of formulation (4.14) given below. In the formulation, binary variables $u_e, e \in E$, specify the path $q \in \hat{P}_d$ we are looking for: $q = \{e \in E : u_e = 1\}$. Binary variables $z^s, s \in S$, in turn, identify the set $S_q(\pi, \lambda)$ corresponding to the so defined path q : $S_q(\pi, \lambda) = \{s \in S : z^s = 1\}$. Besides, $\delta^+(v)$ and $\delta^-(v)$ denote the sets of all links outgoing from, and all links incoming to node $v \in V$, respectively. Also, $o(d)$ is the originating node of the considered demand $d \in D$ and $t(d)$ is its terminating node.

$$\min L = \sum_{e \in E} (\pi_e^0 + \sum_{s \in S} \pi_e^s z^s) u_e - \sum_{s \in S} \lambda_d^s z^s \quad (4.14a)$$

$$\sum_{e \in \delta^+(o(d))} u_e - \sum_{e \in \delta^-(o(d))} u_e = 1 \quad (4.14b)$$

$$\sum_{e \in \delta^+(v)} u_e - \sum_{e \in \delta^-(v)} u_e = 0, \quad v \in V \setminus \{o(d), t(d)\} \quad (4.14c)$$

$$\lambda_d^s z^s \geq \lambda_d^s - \sum_{e \in E_s} \pi_e^s u_e, \quad s \in S \quad (4.14d)$$

$$u_e \in \{0, 1\}, e \in E; \quad z^s \in \{0, 1\}, s \in S. \quad (4.14e)$$

Constraints (4.14b) and (4.14c) assure that those variables u_e that are equal to 1 form a path from o to t . Constraints (4.14d) force each variable $z^s, s \in S$, to be equal to 1 when the length, with respect to π^s , of the path q defined by variables u is sharply smaller than λ_d^s . Assume (for a while) that $z^s = 0$ when the length, with respect to π^s , of path q is greater than or equal to λ_d^s . Then it is clear that the objective function computes the total dual length of path q , i.e., $q = |q|^0 + \sum_{s \in S_q(\pi, \lambda)} (|q|^s - \lambda_d^s)$ (see definition (4.4)). But this assumption will be fulfilled by any optimal solution of (4.14)—this is easily seen from the equivalent form of the objective function

$$L = \sum_{e \in E} \pi_e^0 u_e + \sum_{s \in S} (\sum_{e \in E_s} \pi_e^s u_e - \lambda_d^s) z^s, \quad (4.15)$$

which, when minimized, will set z^s to 0 whenever $|q|^s = \sum_{e \in E_s} \pi_e^s u_e > \lambda_d^s$

Hence, an optimal solution u, z, L of (4.14) defines an optimal path $q := \{e \in E : u_e = 1\}$, with $L = |q|^0 + \sum_{s \in S_q(\pi, \lambda)} (|q|^s - \lambda_d^s)$ equal to its total dual length q .

To get rid of bi-linearities $z^s \cdot u_e$ in the objective function, we can introduce binary variables $Z_e^s, e \in E, s \in S$, rewrite objective as

$$L = \sum_{e \in E} \pi_e^0 u_e + \sum_{e \in E} \sum_{s \in S} \pi_e^s Z_e^s - \sum_{s \in S} \lambda_d^s z^s, \quad (4.16)$$

and add the constraints

$$Z_e^s \leq u_e, Z_e^s \leq z^s, Z_e^s \geq 0, Z_e^s \geq u_e + z^s - 1, e \in E, s \in S. \quad (4.17)$$

These constraints force the equalities $Z_e^s = z^s u_e, e \in E, s \in S$, so that $Z_e^s = 1$ if, and only if, $e \in S_e$ and $s \in S_q(\pi, \lambda)$; otherwise $Z_e^s = 0$. Hence, variables Z can be assumed continuous.

4.4.3 PG algorithm

The algorithm for solving FTOP by path generation is as follows.

Algorithm 3: PG algorithm for Global Rerouting

Input: Graph $G = (V, E)$, a set of demands D , a vector of traffic requirements $h_d^s, s \in S \cup S_0$.

Output: Subset of paths $P \subseteq \hat{P}$ leading to optimal cost $C(P)$.

- 1: Define initial admissible path-lists $P_d, d \in D$, and $P := \bigcup_{d \in D} P_d$
 - 2: Repeat
 - 3: Solve the dual problem $D(P)$ given by (4.2)
 - 4: Extract optimal dual variables π, λ
 - 5: For each demand $d \in D$
 - 6: Solve $PP(P, \pi, \lambda, d)$ formulated in (4.14)
 - 7: If optimal L is negative Then
 - 8: Add the resulting path q to the path-list P_d
 - 9: End If
 - 10: End for
 - 11: Until no path has been added
-

When the algorithm stops, after M iterations, say, it results in a sequence of dual polyhedrons $\Pi(P^0) \supset \Pi(P^1) \supset \dots \supset \Pi(P^M)$, where the inclusions are proper. The corresponding optimal dual objective values are decreasing, $W(P^0) \geq W(P^1) \geq \dots \geq W(P^M)$, and $W(P^M) = W(\hat{P}) = C(\hat{P})$.

Assume that the set S_0 of maximum availability states is not empty. If there exists a path $q \in \hat{r}_d$ with $|q|^0 < \sum_{s \in S_0} \lambda_d^s$, then $|q| \leq |q|^0 - \sum_{s \in S_0} \lambda_d^s < 0$ (because $|q|^s = 0$ for all $s \in S_0$). Hence, for any such path q the degree of violation $D(q)$ is greater than or equal to $\sum_{s \in S_0} \lambda_d^s - |q|^0 > 0$. Thus, while the pricing problem (finding a path q with minimum $|q|$) is in general NP -hard (see Section 4.5), finding a path q with minimum $|q|^0$ is obviously polynomial. Hence, we may speed up the PG process: in Step 2 for each $d \in D$ we find a path $q \in \hat{r}_d$ with minimum $|q|^0$, and if $|q|^0 < \sum_{s \in S_0} \lambda_d^s$, then we add path q to the problem and skip solving $PP(P, \pi, \lambda, d)$ for the current d . Certainly, the overall time efficiency of this modification is in general hard to assess since there is a tradeoff between the value of $D(q)$ (the larger the better) and the time spent in Step 2. An illustration of the above considerations is given in Example 4.4).

Example 4.4 Consider for the network described in Example 4.2 and depicted in Figure 4.5 with the two sets of admissible paths reduced to $P = P_1 \cup P_2$ where $P_1 = \{p_{11}\}$ and $P_2 = \{p_{21}\}$. One of the optimal solutions of the resulting instance $P(P)$ is as follows:

$$x_{11}^0 = x_{21}^0 = 2, x_{11}^s = x_{12}^s = 1, s \in S, y_1^0 = y_2^0 = 2, y_3^0 = y_4^0 = 0, C = 4.$$

The corresponding optimal solution of the dual problem $D(P)$ (see (4.2)) is in turn given by:

$$\pi_1^1 = \pi_2^2 = 2, \pi_3^0 = \pi_4^0 = 1, \lambda_1^1 = \lambda_2^2 = 2, W = 4,$$

and all other dual variables $\pi_1^0, \pi_2^0, \lambda_1^2, \lambda_2^1, \lambda_1^3, \lambda_1^3, \sigma_d^s, s \in S, d \in D$, equal to 0. Hence, applying PG, we find out that paths p_{12} and p_{22} have their total dual lengths (defined by (4.4)) equal to $p_{12} = p_{22} = -1$. Since these values are negative, both paths are added to the problem so that the set of admissible paths P becomes set equal to \hat{r} , i.e., the set of all (loop-less) paths in the network graph for the considered two demands. An optimal solution of the resulting instance $P(\hat{r})$ of FTOP is given in Example 4.2.

4.5 Computational complexity of PP, DSP, and special cases

Although not shown guaranteed, the iterative PG process described in Section 4.4.3 will typically terminate in a polynomial number of steps, as it is closely related to the revised simplex method (Lasdon 1970, Minoux 1986), an exponential algorithm which exhibits this desirable property in practical applications. Indeed, consider a path $q \in \hat{r}_d \not\subseteq P_d$ that solves $PP(P, \pi, \lambda, d)$ for $d \in D$. The variables x_{dq}^0 and $x_{dq}^s, s \in S$, are by assumption non-basic as they are not seen in formulation (4.1) of $P(P)$, and the maximum degree of constraint violation $D(q)$ is equal to the sum of the reduced cost of variable x_{dq}^0 and of the reduced costs of those variables among $x_{dq}^s, s \in S$, that have the negative reduced cost. Thus, the computational complexity of the PG algorithm is in practice determined by the complexity of the pricing problem PP.

Let us consider the special case of state scenarios containing only partial failures of single links. Hence, $S = S_0 \cup S_1$, where S_0 is a set of maximum states (recall that in a maximum state, all links maintain full availability), and $S_1 := \bigcup_{e \in E} S(e)$ is a set of single-link partial availability states. Each set $S(e)$ represents a set of states in which only link e has the reduced capacity, i.e., $s \in S(e)$ implies $\alpha_e^s < 1$ and $\alpha_{e^t}^s = 1, e^t \in E \setminus \{e\}$. Note that if link e is assumed to be perfectly reliable then the set $S(e)$ is empty. It turns out that in the considered case the pricing problem is polynomial.

Lemma 3 Consider a given single-link failure scenario S and a fixed demand $d \in D$. The total dual length of any path $q \in \hat{r}_d$ is given by the formula

$$q) = \sum_{e \in q} \omega(e) - \sum_{s \in S} \lambda_{dq}^s \quad (4.18)$$

where the (non-negative) link weights $\omega(e)$, $e \in E$ are defined by

$$\omega(e) := \pi_e^0 + \sum_{s \in S(e)} \min\{\pi_e^s, \lambda_q^s\}. \quad (4.19)$$

The proof of this lemma is presented in 4.6.2

Since the term $\sum_{s \in S} \lambda_q^s$ in (4.18) is constant, solving PP is equivalent to finding a shortest path with respect to the link weights $\omega(e)$, $e \in E$, e.g., by the Dijkstra algorithm. Hence, PP is polynomial (provided that the number of states in S is polynomial—note that this is the case for a finite number K of link availability levels).

The same argument shows that DSP (the dual separation problem), and hence the dual to FTOP, and thus FTOP, are polynomial as well. Moreover, the way dual is solved by the ellipsoidal method of Khachian (1979) (implied by Proposition 1, i.e., by finding successive paths that separate the current dual solution) gives a polynomial set of paths required to solve FTOP. Besides, since FTOP is polynomial, we may expect that it can be solved with a compact node-link linear program—such an LP formulation is given in the following section (Section 4.6.1).

The above weight construction applies to undirected networks as well. Also, the compact FTOP formulation (4.37) can be easily modified for the undirected links case. Besides, as shown in Section 4.7, the single-node failure scenarios can be treated by polynomial algorithms in a way analogous to the single-link case.

As already mentioned at the end of Section 4.3.2, the dual separation problem is NP -hard for the general case of a state scenario with polynomially bounded size. Moreover, DSP is NP -hard already for partial double-link failures, i.e., when any two links can fail simultaneously. This was demonstrated by Coudert et al. (2007) (see also Orłowski and Pióro (2012)) for a PD problem formulation analogous to (4.1). Thus the pricing problem (4.13) is most likely NP -hard in those cases as well, as it is very similar to DSP.

4.6 Single-link failures

4.6.1 Single-link failures – compact FTOP formulation

A compact node-link FTOP formulation for a given single-link failure scenario $S = S_0 \cup S_1$, where $S_1 = \bigcup_{e \in E} S(e)$, is as follows.

$$\min C = \sum_{e \in E} \xi_e y_e^0 \quad (4.20a)$$

$$\sum_{e \in \delta^+(o(d))} x_{ed}^0 - \sum_{e \in \delta^-(o(d))} x_{ed}^0 = X_d^0, \quad d \in D \quad (4.20b)$$

$$\sum_{e \in \delta^+(v)} x_{ed}^0 - \sum_{e \in \delta^-(v)} x_{ed}^0 = 0, \quad d \in D, v \in V \setminus \{o(d), t(d)\} \quad (4.20c)$$

$$\sum_{d \in D} x_{ed}^0 \leq y_e^0, \quad e \in E \quad (4.20d)$$

$$X_d^0 \geq h_d^s, \quad d \in D, s \in S_0 \quad (4.20e)$$

$$X_d^0 - z_{ed}^s \geq h_d^s, \quad d \in D, e \in E, s \in S(e) \quad (4.20f)$$

$$\sum_{d \in D} (x_{ed}^0 - z_{ed}^s) \leq a_e^s y_e^0, \quad e \in E, s \in S(e) \quad (4.20g)$$

$$z_{ed}^s \leq x_{ed}^0, \quad d \in D, e \in E, s \in S(e) \quad (4.20h)$$

$$\text{all variables } x^0, X^0, z, y^0 \text{ continuous and nonnegative.} \quad (4.20i)$$

In the formulation, variables $x_{ed}^0, e \in E$, are the link-flows of demand $d \in D$ (realizing its nominal path-flows), while variable X_d^0 expresses the overall flow of demand d from $o(d)$ to $t(d)$ (realized by its nominal path-flows). Variables $z_{ed}^s, e \in E, d \in D, s \in S(e)$, in turn, specify the amount of link-flow by which the nominal link-flow x_{ed}^0 is reduced in state $s \in S(e)$. Constraints (4.20b) and (4.20c) are conservation equations for the nominal link-flows. Capacity constraint (4.20d) assures that the nominal link loads do not exceed the maximum link capacities, and constraint (4.20g) assures that the capacity available on link $e \in E$ is not exceeded by its loads in its failure states. Demand constraints (4.20e) and (4.20f) take care about satisfaction of the assumed demand volumes in the maximum states and the single-link failure states, respectively. Finally, constraint (4.20h) bounds the possible link-flow reductions.

In the case of a failure of a single link, say of link $e^t \in E$ in one of its failure states $s \in S(e^t)$, any given pattern of nominal path-flows realizing the nominal link-flows $x_{ed}^0, e \in E$, for a given demand $d \in D$, can be properly thinned according to the values of $z_{e^t d}^s, d \in D$. Let P_d and L_d be the sets of elementary $o(d) - t(d)$ paths and loops, respectively, assigned non-zero path/loop-flows x_{dp}^0 when realizing the nominal link-flows $x_{ed}^0, e \in E$, of a given demand $d \in D$ (such sets of paths and loops exist, see Theorem 8.8 in Korte and Vygen (2012)). Because $\sum_{p \in P_d} x_{dp}^0 = X_d^0$, due to constraints (4.20f) and (4.20g), we can apply thinning for any given failure state $s \in S(e^t)$ only to the nominal path-flows in P_d (and not to the loops in L_d) containing the failing link e^t . Let then $Q_{e^t d} \subseteq P_d$ denote the set of paths that contain the considered link e^t , so that $\sum_{p \in Q_{e^t d}} x_{dp}^0 = x_{e^t d}^0$. We can consistently thin down the nominal path-flows through link e^t in the

considered state $s \in S(e^t)$ according to the formula

$$x_{dp}^s := x_{dp}^0 - \frac{z_{e^t d}^s}{x_{e^t d}^0} \cdot x_{dp}^0, \quad p \in Q_{e^t d}. \quad (4.21)$$

Note that possible loops in the optimal solutions can be eliminated as their capacity is not used by the state-dependent path-flows x_{dp}^s , $p \in P_d$, $s \in S$, realizing the demand volume h_d^s defined by (4.21). Since, as discussed Section 4.2.2, FTOP (4.1) does not require non-elementary paths (and thus loops in the link-flows), formulation (4.20) is correct. In fact, when all link unit capacity costs ξ_e , $e \in E$, are strictly positive, the loops will not appear in the optimal solutions of (4.20).

It is interesting to note that for the single-link failure scenarios the LBP formulation (5.1) remains to be only a lower bound for the exact formulation (4.20). In fact, when an optimal solution $x^0, x^s, s \in S, y^0$ of (5.1) does not contain loops in the flow x^0 then the optimal values of (5.1) determine the corresponding feasible solution $x^0, z^s, s \in S, y^0$ of (4.20) with the same x^0, y^0 and $z_{ed}^s := x_{ed}^0 - x_{ed}^s$, $d \in D$, $e \in E$, $s \in S(e)$. In this case, as LBP is a lower bound for FTOP, its optimum is at the same time the optimum for FTOP. However, if the link-flow x^0 contains loops, then the solution C^t of LBP can be strictly smaller than the optimal solution C of FTOP, as illustrated in Example 4.5 below. This is because no counterpart of constraint (4.20f) is present in (5.1) and hence we may thin the loops without thinning the realized overall demand volume X_σ^0 .

Example 4.5 Consider the network depicted in Figure 4.6. There is one demand (from o to t) and two states, $s = 1$ and $s = 2$, with link 1 and link 2 failing, respectively. In both states the failing link has the same availability coefficient α , and the demand volume to be realized is equal to h . The unit link costs are as follows: $\xi_1 = \xi_2 = \xi_3 = 1, \xi_4 = M$, where $M \gg 1$.

In the considered cost setting with $\xi_4 \gg 1$, optimal solutions of (4.20) (and hence of FTOP) are induced by the nominal path/loop-flows of the form: $x_1^0 = f$ (on the upper path (o, v, t)), $x_2^0 = f$ (on the lower path (o, v, t)), $x_3^0 = 0$ (on the upper loop (o, v, o)), $x_4^0 = 0$ (on the lower loop (o, v, o)). There are two cases to consider.

Case 1: $0 \leq \alpha < \frac{1}{M}$. The optimal solution is given by $f = \frac{h}{1+\alpha}$ and $y^0 = (\frac{h}{1+\alpha}, \frac{h}{1+\alpha}, 0, \frac{2Mh}{1+\alpha})$. Hence, $C = \frac{2h(M+1)}{1+\alpha} \cdot 0$

Case 2: $\frac{1}{M} \leq \alpha \leq 1$. One of the optimal solutions, a symmetric one, is given by $f = \frac{h}{2}$ and $y^0 = (\frac{h}{2\alpha}, \frac{h}{2\alpha}, 0, h)$. Hence, $C = \frac{h}{\alpha} + Mh$. \square

The optimal solution of LBP (5.1), in turn, is induced by the two nominal path-flows equal to $\frac{h}{2}$, and by the two nominal loop-flows equal to $(\frac{1}{1+\alpha} - \frac{1}{2})h = \frac{(1-\alpha)h}{2(1+\alpha)}$. Then the corresponding nominal link-flows are equal to $x^0 = (\frac{h}{1+\alpha}, \frac{h}{1+\alpha}, (1-\alpha)h, \frac{(3-\alpha)h}{1+\alpha})$, and the resulting cost is equal to $C = \frac{\alpha}{1+\alpha} + Mh$. In

the availability state $s = 1$, the link-flows become equal to $x^1 = (\frac{\alpha h}{1+\alpha}, \frac{h}{1+\alpha}, 0, h)$, and in $s = 2$, the link-flows are symmetric and equal to $x^2 = (\frac{h}{1+\alpha}, \frac{\alpha h}{1+\alpha}, 0, h)$.

Certainly, for $M \gg 1$, the LBP cost C^t will be strictly smaller than the FTOP cost C .

Example EC.4 shows that loops can be necessary in the LBP optimum so in general they cannot be removed without an impact on the optimal

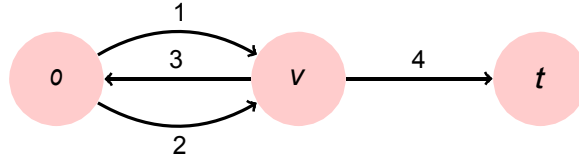


Figure 4.6 – A network example with a loop in the optimum link-flow

solution. Intuitively, the inclusion of loops in LBP allows for (implicit) “forced thinning” of the flows from origin to destination that otherwise would not traverse the failed links. We observe that in the considered example the loops (o, v, o) could be eliminated by putting $x_{ed}^0 = 0, d \in D, e \in \delta^-(o(d)) \cup \delta^+(t(d))$, and $x_{ed}^s = 0, s \in S, d \in D, e \in \delta^-(o(d)) \cup \delta^+(t(d))$, in (5.1). Yet, this does not help much as in the general case loops will appear anyhow, only they will not involve the demands’ origins and destinations.

In the case of undirected links the situation is analogous to the directed case. The standard counterpart of formulation (4.20) for the undirected links (with two oppositely directed link-flows corresponding to each undirected link) remains correct, and the LBP formulation modified for undirected links is still not equivalent to the modified formulation (4.20).

4.6.2 Single-link failures - proof of Lemma 3

First, we observe that the total dual length of path $q \in \hat{\mathcal{P}}_d$ defined by (4.4) can be (in fact also in the general case of S) equivalently expressed as

$$|q| = (|q|^0 + \sum_{s \in S} \forall q /^s) - \sum_{s \in S} \lambda_d^s, \quad (4.22)$$

where the *modified dual length* $\forall q /^s$ of path q in state $s \in S$ is defined as

$$\forall q /^s := \begin{cases} |q|^s, & \text{if } |q|^s \leq \lambda_d^s \\ \lambda_d^s, & \text{otherwise.} \end{cases} \quad (4.23)$$

Since for the considered states $s \in S$, the state-dependent dual length of a path q can take only two values:

$$|q|^s = \begin{cases} 0, & \text{if } s \in S_0, \text{ or } s \in S(e) \text{ and } e \notin q \\ \pi_e^s, & \text{if } s \in S(e) \text{ and } e \in q \end{cases} \quad (4.24)$$

the modified dual path lengths are equal to:

$$\forall q /^s = \begin{cases} 0, & \text{if } s \in S_0, \text{ or } s \in S(e) \text{ and } e \notin q \\ \pi_e^s, & \text{if } s \in S(e) \text{ and } e \in q \text{ and } \pi_e^s \leq \lambda_d^s \\ \lambda_d^s, & \text{if } s \in S(e) \text{ and } e \in q \text{ and } \pi_e^s > \lambda_d^s. \end{cases} \quad (4.25)$$

Now we make use of the link weights $\omega(e), e \in E$, specified by (4.19) in the formulation of Lemma 3:

$$\omega(e) = \pi_e^0 + \sum_{s \in S(e)} \max\{\pi_e^s, \lambda_d^s\}. \quad (4.26)$$

By (4.26) and (4.25) we have

$$\sum_{e \in q} \omega(e) = |q|^0 + \sum_{e \in q} \sum_{s \in S(e)} \forall q /^s = |q|^0 + \sum_{s \in S} \forall q /^s, \quad (4.27)$$

where the quantity on the left-hand side is equal the length of path q with respect to the link weights defined by (4.26) (and (4.19)). Hence, by (4.22) we finally obtain the equality

$$\sum_{e \in q} \omega(e) = |q| + \sum_{s \in S} \lambda_s^s, \quad (4.28)$$

which is equivalent to (4.18).

4.7 Single-node failures

A single-node failures scenario consists of the states with limited availability of the links incident to one single node. Thus, a single-node failure scenario S is composed of a set of maximum states and a set of single-node failure states: $S = S_0 \cup S_2$, where S_0 is a set of maximum states, and $S_2 := \bigcup_{v \in V} S(v)$ is a set of single-node failure states. Each set $S(v)$, if not empty, represents a set of states in which only node v fails. That is, $s \in S(v)$ implies that $\alpha_e^s \leq 1$, $e \in \delta(v)$, $\alpha_e^s = 1$, $e \in E \setminus \delta(v)$, and $\alpha_e^{s(v)} < 1$ for at least one $e \in \delta(v)$ (where $\delta(v) := \delta^-(v) \cup \delta^+(v)$). If node v does not fail at all, then the set $S(v)$ is empty.

4.7.1 Single-node failures – the pricing problem

In the single-node failure scenario the pricing problem (4.13) is also polynomial. To apply shortest path computation to solve PP, we consider, for each $d \in D$, a transformed graph $\hat{G}(d) = (\hat{V}(d), \hat{E}(d))$, constructed from the original network graph $G = (V, E)$ by exchanging, in a way, the roles of nodes and links. The transformation is illustrated in Figures 4.7 and 4.8 for a demand d from node o to node t . The original graph G is depicted on the left-hand side while the transformed graph $\hat{G}(d)$ is shown on the right-hand side.

Graph $\hat{G}(d)$ is defined by the following conditions:

$$\hat{V}(d) := \{\hat{v}(e) : e \in E\} \cup \{o(d), t(d)\}; \quad (\hat{v}(e), \hat{v}(f)) \in \hat{E}(d)$$

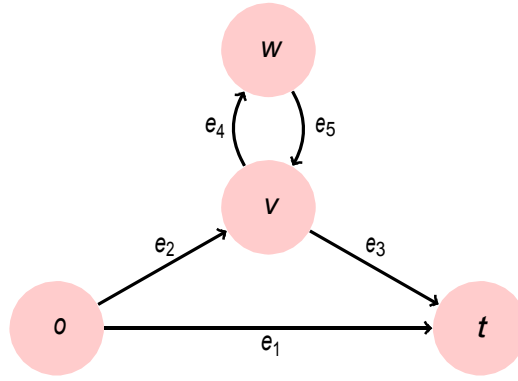
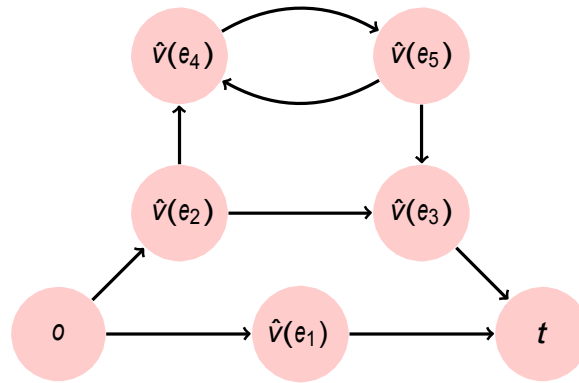
$$\Leftrightarrow \exists v \in V \setminus \{o(d), t(d)\}, e \in \delta^-(v) \wedge f \in \delta^+(v);$$

$$(o(d), \hat{v}(f)) \in \hat{E}(d), f \in \delta^+(o(d)); \quad (\hat{v}(e), t(d)) \in \hat{E}(d), e \in \delta^-(t(d)).$$

In graph $\hat{G}(d)$, all links of the form $(\hat{v}(e), \hat{v}(f))$, where $e \in \delta^-(v)$, $f \in \delta^+(v)$ for a given (original) node $v \in V \setminus \{o(d), t(d)\}$, represent all possible two-link passages through this node in the (original) graph G . On top of that, all links of the form $(o(d), \hat{v}(f))$ reproduce $\delta^+(o(d))$ in the transformed graph, while all links $(\hat{v}(e), t(d))$ reproduce $\delta^-(t(d))$. Finally, paths $q \in \hat{\mathcal{P}}_d$ in G are transformed to their counterparts \hat{q} in $\hat{G}(d)$, and vice versa, in a natural way.

Lemma 4 Consider a given single-node failure scenario S and a fixed demand $d \in D$. The total dual length of any path $q \in \hat{\mathcal{P}}_d$ is given by the formula

$$|q| = \sum_{\hat{e} \in \hat{q}} \omega(\hat{e}) - \sum_{s \in S} \lambda_s^s, \quad (4.29)$$

Figure 4.7 – Original graph G Figure 4.8 – Transformed graph $\hat{G}(d)$

where the (non-negative) link weights $\omega(\hat{e})$ are defined for each link \hat{e} in $\hat{E}(d)$ as follows:

$$\omega(\hat{v}(e), \hat{v}(f)) := \pi_f^0 + \sum_{s \in S(v)} \min\{\pi_e^s + \pi_f^s, \lambda_d^s\},$$

$$v \in V \setminus \{o(d), t(d)\}, e \in \delta^-(v), f \in \delta^+(v) \quad (4.30a)$$

$$\omega(o(d), \hat{v}(f)) := \pi_f^0 + \sum_{s \in S(o(d))} \min\{\pi_f^s, \lambda_d^s\}, f \in \delta^+(o(d)) \quad (4.30b)$$

$$\omega(\hat{v}(e), t(d)) := \sum_{s \in S(t(d))} \min\{\pi_e^s, \lambda_d^s\}, e \in \delta^-(t(d)). \quad (4.30c)$$

As in the proof of Lemma 3 (see (4.22)), the total dual length of path $q \in \hat{F}_d$ defined by (4.4) can be expressed as:

$$q) = (|q|^0 + \sum_{s \in S} \#q^s) - \sum_{s \in S} \lambda_d^s. \quad (4.31)$$

In the considered case, for a given path $q \in \hat{F}_d$, its dual lengths in the considered states $s \in S$ can take one of the three values:

$$|q|^s = \begin{cases} 0, & \text{if } s \in S_0, \text{ or } s \in S(v) \text{ and } q \cap \delta(v) = \emptyset \\ \pi_e^s, & \text{if } s \in S(v) \text{ and } q \cap \delta(v) = \{e\} \\ \pi_{e^t}^s + \pi_{e^{tt}}^s, & \text{if } s \in S(v) \text{ and } q \cap \delta(v) = \{e^t, e^{tt}\}. \end{cases} \quad (4.32)$$

Thus, the modified dual lengths, defined in (4.23), are as follows:

$$\forall q \neq \emptyset = \begin{cases} 0, & \text{if } s \in S_0, \text{ or } s \in S(v) \text{ and } q \cap \delta(v) = \emptyset \\ \pi_e^s, & \text{if } s \in S(v) \text{ and } q \cap \delta(v) = \{e\} \text{ and } \pi_e^{s(v)} \leq \lambda_d^{s(v)} \\ \lambda_d^s, & \text{if } s \in S(v) \text{ and } q \cap \delta(v) = \{e\} \text{ and } \pi_e^{s(v)} > \lambda_d^{s(v)} \\ \pi_{e^t}^s + \pi_{e^{tt}}^s, & \text{if } s \in S(v) \text{ and } q \cap \delta(v) = \{e^t, e^{tt}\} \text{ and } \pi_{e^t}^{s(v)} + \pi_{e^{tt}}^{s(v)} \leq \lambda_d^{s(v)} \\ \lambda_d^s, & \text{if } s \in S(v) \text{ and } q \cap \delta(v) = \{e^t, e^{tt}\} \text{ and } \pi_{e^t}^{s(v)} + \pi_{e^{tt}}^{s(v)} > \lambda_d^{s(v)}. \end{cases} \quad (4.33)$$

Let \hat{q} be a path from $o(d)$ to $t(d)$ in graph \hat{G} represented by the sequence of nodes in \hat{v} , $\hat{q} = (o(d), \hat{v}(e_1), \hat{v}(e_2), \dots, \hat{v}(e_n), t(d))$. Then the corresponding sequence of links from E , $q := (e_1, e_2, \dots, e_n)$, is a path from $o(d)$ to $t(d)$ in graph G , i.e., $q \in P_d$. Conversely, if we specify path $q = (e_1, e_2, \dots, e_n)$ from $o(d)$ to $t(d)$ in graph G , then $\hat{q}(o(d), \hat{v}(e_1), \hat{v}(e_2), \dots, \hat{v}(e_n), t(d))$ will be a path from $o(d)$ to $t(d)$ in graph \hat{G} . This establishes a one-to-one correspondence of $o(d) - t(d)$ paths \hat{q} and q in graphs \hat{G} and G , respectively.

Now we make use of the link weights $\omega(\hat{e})$, $\hat{e} \in \hat{E}$, specified by (4.30) in the formulation of Lemma 4:

$$\omega(\hat{v}(e), \hat{v}(f)) := \pi_f^0 + \sum_{s \in S(v)} \min\{\pi_e^s + \pi_f^s, \lambda_d^s\}, \quad v \in V \setminus \{o(d), t(d)\}, \quad e \in \delta^-(v), \quad f \in \delta^+(v) \quad (4.34a)$$

$$\omega(o(d), \hat{v}(f)) := \pi_f^0 + \sum_{s \in S(o(d))} \min\{\pi_f^s, \lambda_d^s\}, \quad f \in \delta^+(o(d)) \quad (4.34b)$$

$$\omega(\hat{v}(e), t(d)) := \sum_{s \in S(t(d))} \min\{\pi_e^s, \lambda_d^s\}, \quad e \in \delta^-(t(d)) \quad (4.34c)$$

and consider a path $\hat{q} = (o(d), \hat{v}(e_1), \hat{v}(e_2), \dots, \hat{v}(e_n), t(d))$. We observe that the length $\sum_{\hat{e} \in \hat{q}} \omega(\hat{e})$ of path \hat{q} with respect to the link weights (4.34) obeys the equality

$$\sum_{\hat{e} \in \hat{q}} \omega(\hat{e}) = |q|^0 + \sum_{s \in S} \forall q \neq \emptyset^s \quad (4.35)$$

where $q = (e_1, e_2, \dots, e_n)$ is the path in graph G corresponding to \hat{q} . Hence,

$$\sum_{\hat{e} \in \hat{q}} \omega(\hat{e}) = |q| + \sum_{s \in S} \lambda_d^s, \quad (4.36)$$

which is equivalent to (4.29).

In consequence, and due to the one-to-one correspondence of the paths in the original graph and the transformed graph, the pricing problem in the considered case of single-node failures can be solved by a shortest path algorithm applied to graph \hat{G} .

The above construction can be applied, after a slight modification, to undirected graphs. In the undirected case we first define the transformed graph as an undirect graph, similarly as in the directed case. Then we transform it to a bi-directed graph, with two oppositely directed arcs corresponding to each undirected link. The two arcs in each such pair are assigned non-symmetric arc weights, differing in the π_f^0 value that is determined by the direction of the arc (see definition (4.30a)).

In this case we just skip the directions of links in the definition of the new graph and adopt the following definition:

$$\begin{aligned} V &:= \{\hat{v}(e) : e \in E\} \cup \{o(d), t(d)\} \\ \exists v \in V, e, f \in \delta(v) &\Rightarrow \{\hat{v}(e), \hat{v}(f)\} \in \hat{E}(d) \\ \{o(d), \hat{v}(f)\} &\in \hat{E}(d), f \in \delta(o(d)) \\ \{\hat{v}(e), t(d)\} &\in \hat{E}(d), e \in \delta(t(d)). \end{aligned}$$

Since the term $\sum_{s \in S} \lambda_d^s$ in (4.18) is constant, solving PP is equivalent to finding a shortest path in the transformed graph with respect to the link weights $\omega(\hat{e}), \hat{e} \in \hat{E}$. Hence, PP is polynomial, and so is FTOP. A compact LP formulation for the considered case of FTOP is given in Section 4.7.2.

4.7.2 Single-node failures – compact FTOP formulation

A compact node-link FTOP formulation for a given single-node failure scenario $S = S_0 \cup S_2$, where $S_2 = \bigcup_{v \in V} S(v)$, is given in 4.37. Below, $D(v) := \{d \in D : v \notin \{o(d), t(d)\}\}$ denotes the set of all demands for which node v is a transit node. Similarly $D^-(v) := \{d \in D : v = t(d)\}$ and $D^+(v) := \{d \in D : v = o(d)\}$ denote the sets of demands with v as the destination node and the originating node, respectively. Transit flow variables $w_{e^-e^+d}^v, v \in V \setminus \{o(d), t(d)\}, e^- \in \delta^-(v), e^+ \in \delta^+(v)$, specify the amount of nominal flow realizing demand d traversing its transit node v via links e^- and e^+ . Next, variables $z_{e^-e^+d}^s, v \in V \setminus \{o(d), t(d)\}, e^- \in \delta^-(v), e^+ \in \delta^+(v), s \in S(v)$, specify the amount of flow by which the portion of the nominal flow X_d^0 of demand d traversing its transit node v via links e^- and e^+ is reduced when node v fails in state $s \in S(v)$. Further, variables $Z_{ed}^s, d \in D, e \in \delta^-(t(d)), s \in S(t(d))$, specify the amount of flow by which the portion of the nominal flow X_d^0 of demand d entering its destination node $t(d)$ via link e is reduced when node $t(d)$ fails in state $s \in S(t(d))$. Finally, variables $Z_{ed}^s, d \in D, e \in \delta^+(o(d)), s \in S(o(d))$, specify the amount of flow by which the portion of the nominal flow X_d^0 of demand d leaving its originating node $o(d)$ via link e is reduced when node $o(d)$ fails in state $s \in S(o(d))$.

The values of transit variables $w_{e^-e^+d}^v$ can be used to define the overall flow on the paths $p \in \hat{P}_d$ traversing a transit node v of a demand $d \in D$ containing two particular links e^- and e^+ , where $e^- \in \delta^-(v)$ and $e^+ \in \delta^+(v)$. When node v fails, the flows on these paths can be consistently thinned using the value of $z_{e^-e^+d}^s$ instead of $w_{e^-e^+d}^v$ in a formula analogous to (4.21). A similar observation applies to variables Z .

For a fixed $d \in D$, an elementary path-flow pattern $x_{dp}^0 > 0, p \in P_d$ (for some $P_d \subseteq \hat{P}_d$), that realizes the link-flows $x_{ed}^0, e \in E$, and is consistent with the values of variables w can be found recursively as follows. Initially, we set P_d equal to \emptyset , and find a (elementary) path $q = (v_0, e_1, v_1, e_2, \dots, e_{n-1}, v_{n-1}, e_n, v_n)$ in \hat{P}_d , where $v_0 = o(d)$ and $v_n = t(d)$, such that the values of all variables $x_{e_1d}^0, w_{e_1e_2d}^{v_1}, w_{e_2e_3d}^{v_2}, \dots, w_{e_{n-1}e_nd}^{v_{n-1}}, x_{e_nd}^0$ are strictly greater than 0. Then we define $x_{dq}^0 := \min\{x_{e_1d}^0, w_{e_1e_2d}^{v_1}, w_{e_2e_3d}^{v_2}, \dots, w_{e_{n-1}e_nd}^{v_{n-1}}, x_{e_nd}^0\}$, subtract the flow x_{dq}^0 from variables $x_{e_1d}^0, w_{e_1e_2d}^{v_1}, w_{e_2e_3d}^{v_2}, \dots, w_{e_{n-1}e_nd}^{v_{n-1}}, x_{e_nd}^0$, add the so found path q

to P_d , and repeat the procedure until no such path q for d can be found. This observation justifies the correctness of formulation (4.37).

$$\min C = \sum_{e \in E} \xi_e y_e^0 \quad (4.37a)$$

$$\sum_{e \in \delta^+(o(d))} x_{ed}^0 - \sum_{e \in \delta^-(o(d))} x_{ed}^0 = X_d^0, \quad d \in D \quad (4.37b)$$

$$\sum_{e \in \delta^+(v)} x_{ed}^0 - \sum_{e \in \delta^-(v)} x_{ed}^0 = 0, \quad d \in D, v \in V \setminus \{o(d), t(d)\} \quad (4.37c)$$

$$\sum_{d \in D} x_{ed}^0 \leq y_e^0, \quad e \in E \quad (4.37d)$$

$$X_d^0 \geq h_d^s, \quad d \in D, s \in S_o \quad (4.37e)$$

$$\sum_{e^- \in \delta^-(v)} w_{e^-ed}^v = x_{ed}^0, \quad d \in D, v \in V \setminus \{o(d), t(d)\}, e \in \delta^+(v) \quad (4.37f)$$

$$\sum_{e^+ \in \delta^+(v)} w_{ee^+d}^v = x_{ed}^0, \quad d \in D, v \in V \setminus \{o(d), t(d)\}, e \in \delta^-(v) \quad (4.37g)$$

$$z_{e^-e^+d}^s \leq w_{e^-e^+d}^v, \quad d \in D, v \in V \setminus \{o(d), t(d)\}, \\ e^- \in \delta^-(v), e^+ \in \delta^+(v), s \in S(v) \quad (4.37h)$$

$$X_d^0 - \sum_{e^- \in \delta^-(v)} \sum_{e^+ \in \delta^+(v)} z_{e^-e^+d}^s \geq h_d^s, \quad d \in D, \\ v \in V \setminus \{o(d), t(d)\}, s \in S(v) \quad (4.37i)$$

$$Z_{ed}^s \leq x_{ed}^0, \quad d \in D, e \in \delta^+(o(d)), s \in S(o(d)) \quad (4.37j)$$

$$X_d^0 - \sum_{e \in \delta^+(o(d))} Z_{ed}^s \geq h_d^s, \quad d \in D, s \in S(o(d)) \quad (4.37k)$$

$$Z_{ed}^s \leq x_{ed}^0, \quad d \in D, e \in \delta^-(t(d)), s \in S(t(d)) \quad (4.37l)$$

$$X_d^0 - \sum_{e \in \delta^-(t(d))} Z_{ed}^s \geq h_d^s, \quad d \in D, s \in S(t(d)) \quad (4.37m)$$

$$\sum_{d \in D} x_{ed}^0 - \sum_{d \in D(v)} \sum_{e^- \in \delta^-(v)} z_{e^-ed}^s - \sum_{d \in D^+(v)} Z_{ed}^s \leq \alpha_e^s y_e^0, \\ v \in V, e \in \delta^+(v), s \in S(v) \quad (4.37n)$$

$$\sum_{d \in D} x_{ed}^0 - \sum_{d \in D(v)} \sum_{e^+ \in \delta^+(v)} z_{ee^+d}^s - \sum_{d \in D^-(v)} Z_{ed}^s \leq \alpha_e^s y_e^0, \\ v \in V, e \in \delta^-(v), s \in S(v) \quad (4.37o)$$

$$\text{all variables } x^0, X^0, w, z, Z, y^0 \text{ continuous and nonnegative.} \quad (4.37p)$$

It turns out that the LBP formulation becomes exact when it has loop-less optimal solutions and each node can fail at most once (i.e., $|S(v)| \leq 1, v \in V$). This is because only then any feasible loop-less solution of (5.1) determines a feasible solution of (4.37) with the same cost. (Note that this is true for an arbitrary single-link failure scenario.) For single-node failure scenarios admitting $|S(v)| > 1$ for some $v \in V$, the above observation is not true anymore because of constraint (4.37h) which implies that for each value of $w_{e^-e^+d}^v$ (for v, e^-, e^+, d fixed) must be in this case an upper bound for the set of values $z_{e^-e^+d}^s, s \in S(v)$ with more than one element. This is illustrated by the following example.

Example 4.6 Consider the network depicted in Figure 4.9 with 4 links $e = 1, 2, 3, 4$. The unit capacity cost of all four links is equal to 1. The single demand (with demand

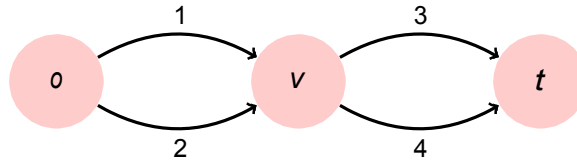


Figure 4.9 – A network example

volume h) between nodes o and t can be realized on 4 paths: $p_1 = \{1, 3\}$, $p_2 = \{1, 4\}$, $p_3 = \{2, 3\}$, $p_4 = \{2, 4\}$. We consider a single-node failure scenario with 4 states $s = 1, 2, 3, 4$ characterized by the sets of failing links $E_1 = \{2, 4\}$, $E_2 = \{2, 3\}$, $E_3 = \{1, 4\}$, $E_4 = \{1, 3\}$. (Note that the four availability states can be considered as four different failures of node v .) The availability coefficients of all the failing links in all the states are equal to α . The demand has the nominal volume and the volumes in the states all equal to h . Note that in each state $s = 1, 2, 3, 4$ only path p_s remains unaffected.

It is clear that in terms of link capacity the optimal solution of the lower bound formulation (5.1) is unique and symmetric. Hence, a simple cut-based argumentation implies that the common optimal value y^0 of the link capacity must fulfil the equality $y^0 + \alpha y^0 = h$, that is, $y^0 = \frac{h}{1+\alpha}$. The common value x^0 for the optimal nominal link-flows is equal to y^0 . In each state, the optimal link-flows on the links with reduced capacity are equal to $\frac{h\alpha}{1+\alpha}$, while on the fully available links—to $\frac{h}{1+\alpha}$. Hence, the optimal LBP objective function value is equal to $C^t = \frac{4h}{1+\alpha}$.

The optimal solution for FTOP is obtained by considering the two following cases.

Case 1: $0 \leq \alpha < \frac{1}{2}$. The optimal FTOP solution is as follows: $f = \frac{h}{1+4\alpha}$, $y^0 = \frac{2h}{1+4\alpha}$, $C = \frac{8h}{1+4\alpha}$. All four nominal path-flows are equal to f . In each state s , the nominal flow f on the unaffected path p_s is maintained, the flow on the path with both links affected is deleted (thinned to 0), and the flows on the two remaining paths are thinned from f to $\frac{2h\alpha}{1+4\alpha}$. \square

Case 2: $\frac{1}{2} \leq \alpha \leq 1$. The optimal FTOP solution is as follows: $f = \frac{h}{2(1+\alpha)}$, $y^0 = \frac{h}{1+\alpha}$, $C = \frac{4h}{1+\alpha}$. In state s , the nominal flow f on the unaffected path p_k is maintained, the flows on the two paths with only one link affected are also maintained, and the flow on the path with both links affected is thinned to $\frac{h(2\alpha-1)}{2(1+\alpha)}$. \square

Note that in Case 2 the optimal solutions of LBP and of FTOP have the same cost, i.e., $C^t = C$. In Case 1, however, $C^t < C$ and $\frac{C-C^t}{C} = \frac{1-2\alpha}{2(1+\alpha)}$. For $\alpha = 0$, the difference of the FTOP cost and the LBP cost constitutes 50% of the FTOP cost. For $\alpha = \frac{1}{4}$, this value decreases to 20%.

We note that for the single-node failures, the relation between the exact formulation (4.37) and the lower bound formulation (5.1) in the undirected case is the same as for the directed case.

Efficiency of Flow Thinning Optimization

Contents

5.1	Cost efficiency of flow thinning	68
5.1.1	Single partial link failure scenarios	69
5.1.2	Single partial node failure scenarios	70
5.1.3	Double partial failure scenarios	72
5.2	Computational efficiency of flow thinning optimization	73
5.2.1	Path generation	73
5.2.2	Evolution of the PG iterations	74
5.2.3	Running times	76
5.3	A lower bound formulation	77
5.3.1	LBP – Lower Bound Problem	77
5.3.2	Comparing LBP and FTOP	77

This chapter is devoted to numerical part. The purpose of our computational experiments is threefold. The first objective is to demonstrate the traffic efficiency of *flow thinning*. Our results from section 5.1 illustrate the efficiency of FTS by comparing the corresponding minimum capacity allocation costs for several strategies. Second, we assess the computational difficulty of FTOP. Hence, in section 5.2 we present the results of two sets of experiments which evaluate the computational efficiency of the path generation algorithm. Finally, since FTOP is *NP*-hard, we introduce a polynomial problem that provides a lower bound for its cost $C(\hat{P})$ referred to as LBP. The results of LBP will be compared with FTOP, formulated in 4.1.

In this chapter, we present the results of our numerical study carried out for four realistic undirected network examples taken from the library of network instances SNDlib (Orlowski et al. 2010): *pdh*, *polska*, *nobel-us*, and *nobel-germany*. Figure 5.1 represents the number of vertices, edges and demands of each networks, as well as the maximum and minimum degree of the corresponding graphs ($\Delta(G)$, $\delta(G)$). The demand matrices for those networks specified are taken as the nominal demand volumes in

our models. The study was carried out on a computer equipped with an Intel(R) Xeon(R) CPU E5-2670 @ 2.60GHz processor, and 132 GB of RAM, using CPLEX 12.5 (CPLEX 2013).

<i>network</i>	$ V $	$ E $	$ D $	$\Delta(G)$	$\delta(G)$
<i>pdh</i>	11	34	24	9	7
<i>polska</i>	12	18	66	5	2
<i>nobel-us</i>	14	21	91	6	2
<i>nobel-germany</i>	17	26	121	4	2

Table 5.1 – Network description.

5.1 Cost efficiency of flow thinning

In this section, we compare the cost effectiveness of *flow thinning* and *global rerouting*. We first start by a short note on the typology of failures occurring frequently. Let us consider the example presented in Figure 5.1 representing rainfall in north-west of Poland a day of July 2015¹.

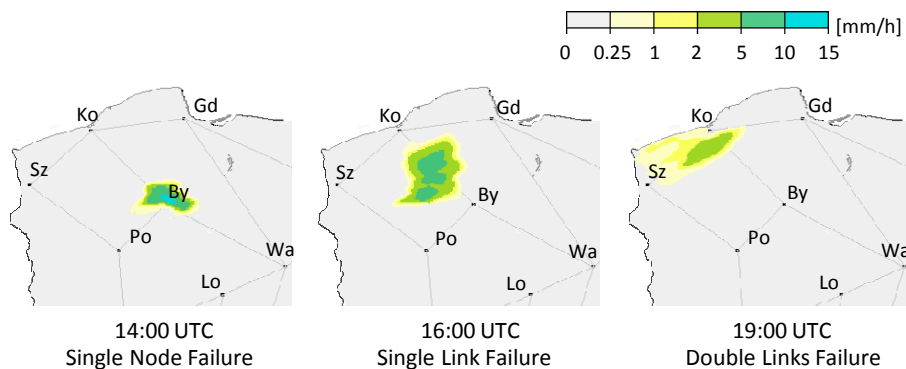


Figure 5.1 – Rainfall radar over Poland for network *polska*.

We can notice the following type of failure scenarios illustrated in the figure:

- single link failures
- single node failures
- double link failures (in close proximity) ...

In the sequel, these three failure scenarios will be studied and we will also present results for double link failures where the two links are randomly located (realistic for large scale network).

Let the quantity C_{GR} denote the optimal cost for reliable networks under GR (recall that GR implies the cheapest networks and is thus taken as the reference value). The cost increases with respect to GR (in percents) for FTS is given by $Gap_{FTS} := \frac{C_{FTS} - C_{GR}}{C_{GR}} \times 100$, where C_{FTS} is the cost for the flow thinning strategy. Certainly, the computed link capacities (and thus the link capacity cost) ensure routing of all traffic demands in the

¹Data from WeatherOnline Limited - July 2015 (<http://www.weatheronline.co.uk/>)

nominal state and the guaranteed traffic restoration (specified by β) in all failure states. In the reported experiments, we always use the same traffic reduction ratio $\beta = 1$ for all demands in all failure states, i.e., $h_d^s = h_d^0$ for all $d \in D$ and $s \in S$.

5.1.1 Single partial link failure scenarios

In Figures 5.2-5.5, for all four network instances, we present Gap_{FTS} as a function of the availability coefficient α assumed for the failure states. We here consider only single partial link failures, i.e. in a failure state one link can loose a percentage of its nominal capacity ($1 - \alpha\%$) and all remaining links reach their nominal capacities. We consider 21 scenarios, varying α from 0% (nominal state) to 100% (total failure) with a step of 5%. Note that the dotted lines represents availability coefficient for which the relative gaps meet levels of 10%, 20%, 30%, 40% and 50%. These levels are used to compare the network instances.

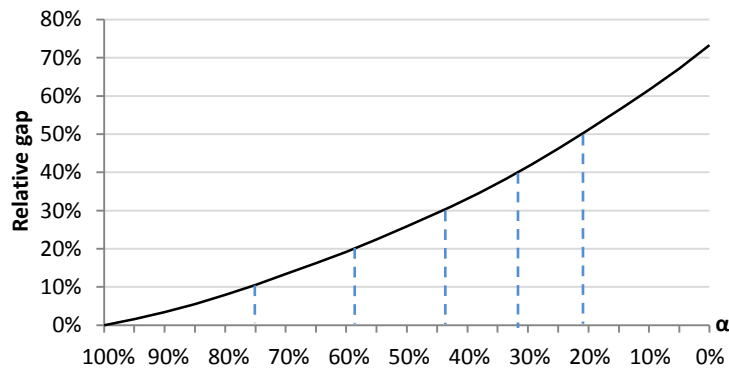


Figure 5.2 – Relative gap between GR and FTS depending on the availability coefficient α for network *pdh* (link failures).

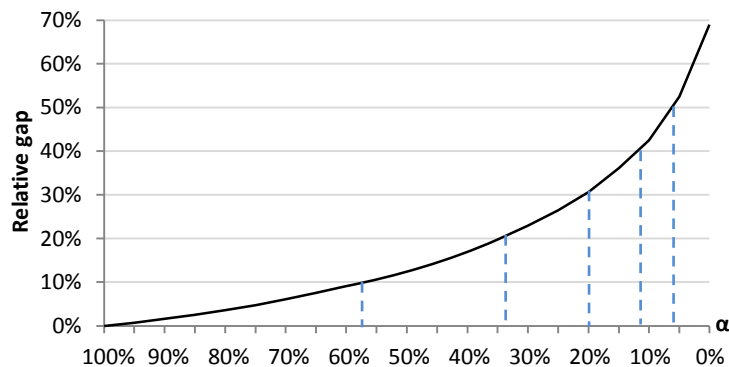


Figure 5.3 – Relative gap between GR and FTS depending on the availability coefficient α for network *polska* (link failures).

First, we note that for $\alpha = 0\%$ FTS comes to be as *path diversity*. Considering single total link failure scenarios, PD can be considered as an upper bound of the cost of the network for all single partial link failure scenarios that can occur.

Second, let us analyze the performance of FTS. As shown in Figures 3.1-3.3 (see Chapter 3), the gap between the dimensioning cost of *global*

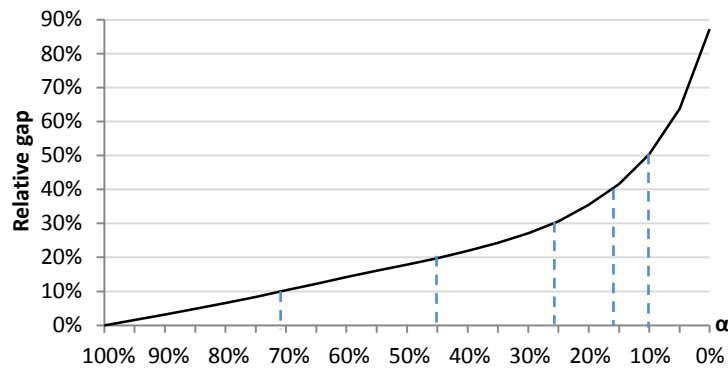


Figure 5.4 – Relative gap between GR and FTS depending on the availability coefficient α for network nobel-us (link failures).

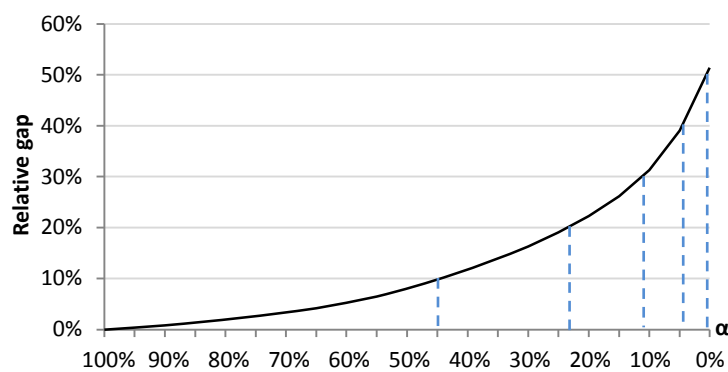


Figure 5.5 – Relative gap between GR and FTS depending on the availability coefficient α for network nobel-germany (link failures).

rerouting and *local rerouting* is up to 30% for total failures. Hence, as *local rerouting* is used in *MPLS Fast Reroute* (described in section 2.2.3), let us consider a relative gap of 30% as acceptable for a recovery mechanism. Let us track now when this value is achieved for two types of networks, meshed and sparse, respectively network *pdh* versus *polska*, *nobel-us* and *nobel-germany*. In one hand, the relative gap of 30% is met for $\alpha = 0.45$ in *pdh*. Hence, FTS is of reasonable cost for light to moderate perturbations. On the other hand, considering sparse network, the relative gap of 30% is met for $\alpha = 0.25$ up to $\alpha = 0.1$. Thus, FTS can be considered cost-efficient even for strong perturbations in sparse networks. However, in both cases, strong perturbations (including total failure) cannot be managed by FTS with reasonable cost.

5.1.2 Single partial node failure scenarios

In Figures 5.6-5.8, for all four network instances, we present Gap_{FTS} as a function of the availability coefficient α assumed for the failure states. We here consider only single partial node failures, i.e. in a failure state every link outgoing from or incoming to the failed node suffers a degradation of their capacities of $1 - \alpha\%$ and all remaining links reach their nominal capacities. We consider 20 scenarios, varying α from 0% (nominal state) to 95%. Note that in this context, we cannot present results on total node failures as we require the traffic to be fully rerouted in case of failure. Note

that even though the failure scenarios may appear stronger than single link failures due to the fact that many links fail in the same time, the computation time of FTOP is significantly decreased due to the reduction of the number of failure scenarios $|S|$ and of the number of variables ($|V| \ll |E|$ for all network instances).

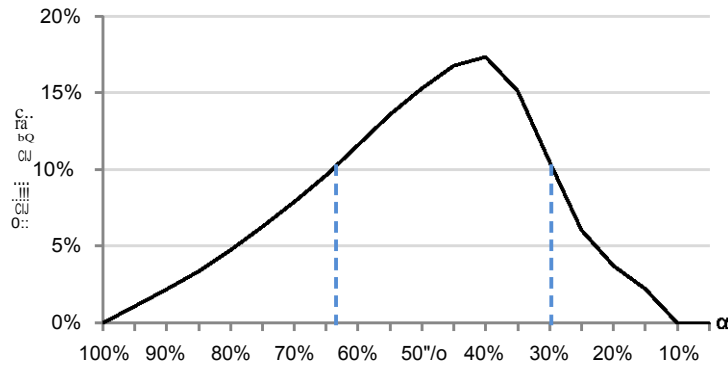


Figure 5.6– Relative gap between GR and FTS depending on the availability coefficient α for network polska (node failures).

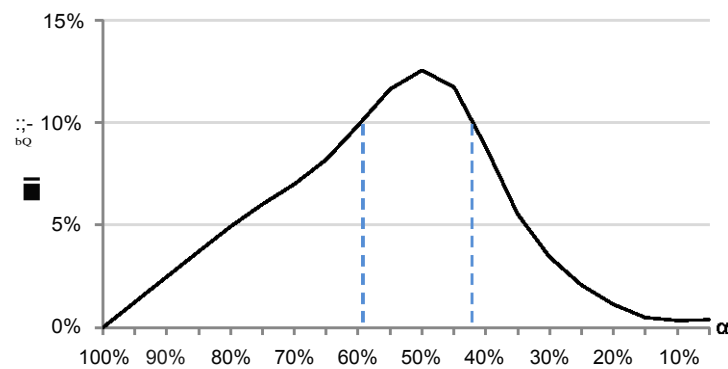


Figure 5.7 – Relative gap between GR and FTS depending on the availability coefficient α for network nobel-us (node failures).

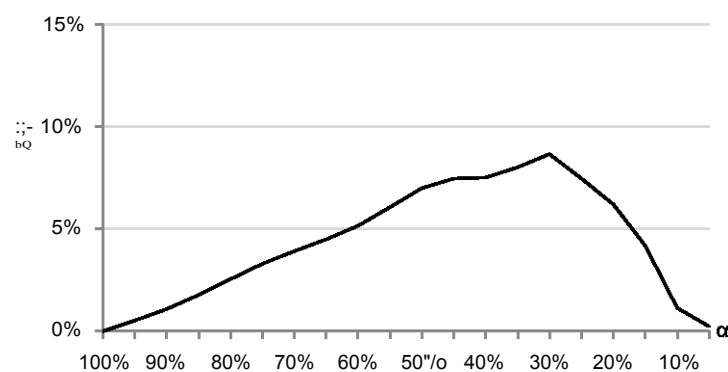


Figure 5.8 – Relative gap between GR and FTS depending on the availability coefficient α for network nobel-germany (node failures).

Considering single node failure scenarios, flow thinning is clearly a cost-efficient strategy with a relative gap to GR inferior to 20% for all α . Note that the relative gap for network *pdh* is equal to 0 for all values of α .

Contrary to single link failure scenarios, this kind of scenarios can be managed by FTS with limited cost ($Gap_{FTS} < 5\%$) for light perturbations but higher cost for relatively major perturbations. Note that the cost of FTS for moderate perturbations remains reasonable ($< 17\%$ for all networks). This is mostly due to the fact that we require to recover all demand traffic after a failure. In case of major perturbations, GR and FTS have the same behavior i.e. they use the shortest path to route the traffic and overdimension the network. However, for moderate perturbations, GR becomes more cost-efficient.

5.1.3 Double partial failure scenarios

In Figure 5.9, we present Gap_{FTS} as a function of the availability coefficient α assumed for failure scenarios corresponding to the failure of two simultaneous links close to each other. This proximity was determined as follows. Considering a planar graph (here *polska*), two link can fail in the same scenario only if an arc joining the two links can be drawn without cutting any other link. Hence, for network *polska* we will get 26 failure scenarios. Once again, note that in this context, we cannot present results on total node failures as we require the traffic to be fully rerouted in case of failure.

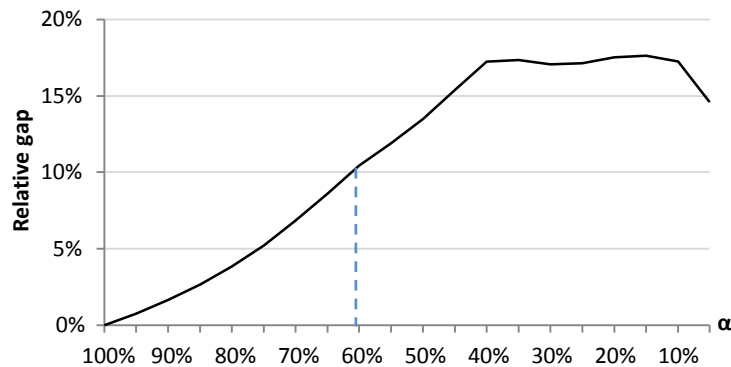


Figure 5.9 – Relative gap between GR and FTS depending on the availability coefficient α for network *polska* (failures of two links in close proximity).

Next, Figure 5.10 represents the Gap_{FTS} as a function of the availability coefficient α for double links failures i.e. the set of all pair of links (including single link failures). Table 5.2 gives for *polska* the size of scenarios set with respect to four types of failure scenarios.

Notice that, considering network *polska*, the four failure scenarios presented are related to a growing number of failures illustrated in Table 5.2.

	$ S $
single node failures	12
single link failures	18
two near-links failures	26
double links failures	324

Table 5.2 – Number of failures $|S|$ depending on the failure scenarios.

The results show that assuming FT instead of GR increases the cost by up to 17% maximum for the two failing links scenarios. Furthermore,

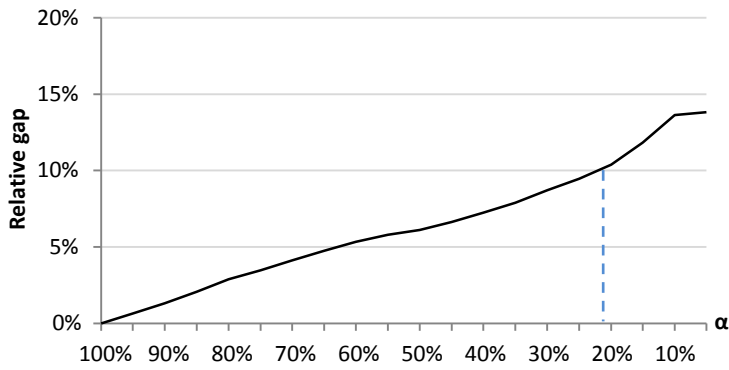


Figure 5.10 – Relative gap between GR and FTS depending on the availability coefficient α for network *polska* (double links failures).

considering failures of close links presented in Figure 5.9, FTS manages the failures with the same relative gap from $\alpha = 0.45$ (moderate perturbation) to $\alpha = 0$ (total failure). Summing up on the performance of *flow thinning*, FTS can be considered cost-efficient except for major single link failures as it always give a relative gap inferior to 30% for all network instances and all α . In the following section, we deal with the efficiency of the resolution algorithm for FTOP.

5.2 Computational efficiency of flow thinning optimization

5.2.1 Path generation

Recall that PG is the path generation algorithm for FTOP (i.e., problem (4.1) assuming the full set of admissible paths), based on the binary pricing problem (PP) formulation (4.14) (see Section 4.4.3), and that $\text{FTOP}^<$ denotes instances of (4.1) with limited sets of admissible paths.

Below we report results illustrating the computational efficiency of solving FTOP through PG, and of solving $\text{FTOP}^<$ through a direct use of CPLEX for a predefined set of admissible paths. The PG algorithm was initialized by putting only one shortest (with respect to the unit capacity costs ξ_e , $e \in E$) path on the path-list of each demand. (Note that such minimal path-lists are sufficient for an optimal solution for FTOP involving only the nominal state, i.e., for FTOP with $S = \{s_0\}$, and also for a feasible solution when links do not fail totally, i.e., when $\alpha_e^s > 0$, $e \in E$, $s \in S$.) For $\text{FTOP}^<$, the predefined sets of admissible paths contain all elementary paths having up to 5 links.

For the reported experiments we assume a uniform availability coefficient α for all the affected links in each failure state, i.e., $\alpha_e^s = \alpha$, $s \in S$, $e \in E_s$ (recall that E_s denotes the subset of links affected by failure s), and $\alpha_e^s = 1$, $s \in S$, $e \in E \setminus E_s$. Also, we use a uniform demand satisfaction coefficient β , i.e., $h_d^s = \beta h_d^0$, $d \in D$, $s \in S \setminus \{s_0\}$.

Tables 5.3 and 5.4 present results for three networks (*pdh*, *polska* and *nobel-us*) under the single-link failure scenario (SL) and the double-link failure scenario (DL). SL contains the nominal state plus one failure state for each link (with the assumed α for the failing link). DL contains SL

plus one failure state for each pair of links (with the assumed α for both failing links). Each type of scenario is considered for the four pairs (α, β) specified by all four combinations of $\alpha \in \{0.25, 0.5\}$ and $\beta \in \{0.75, 1\}$.

The following quantities denote the running time (in seconds) observed in our experiments:

- $t_{FTOP^<}$ – solving FTOP for the predefined set of admissible paths
- t_{GP} – generating the predefined set of admissible paths ($t_{FTOP^<}$ includes t_{GP})
- t_{FTOP} – solving FTOP through PG
- t_{PP} – solving the pricing problems (4.14) within the PG algorithm (t_{FTOP} includes t_{PP}).

In addition, the quantity in brackets in column “network” gives the number of predefined paths (generated in time t_{GP}), column “#iter.” gives the number of iterations of the PG algorithm, and column “#gen.paths” gives the number of paths generated by PG. We set the running time limit to 3 hours; the cases when the timeout had to be applied (this happened only for FTOP[<]) are denoted by *. Note that the total number of paths for network *pdh* (resp. *polska* and *nobel-us*) is equal to 6 639 paths (resp. 491 and 609 paths).

network	α	β	$t_{FTOP^<}$	t_{GP}	t_{FTOP}	t_{PP}	#iter.	#gen.paths
<i>pdh</i>	0.5	0.75	467	308	28	22	8	298
	0.5	1	460	299	46	35	11	335
	0.25	0.75	478	311	43	34	11	198
	0.25	1	293	473	52	43	12	316
<i>polska</i>	0.5	0.75	533	5	16	13	5	412
	0.5	1	560	6	28	22	8	466
	0.25	0.75	383	4	23	19	7	471
	0.25	1	2451	2	27	20	5	506
<i>nobel-us</i>	0.5	0.75	4	2	29	23	6	530
	0.5	1	4	2	43	35	8	591
	0.25	0.75	4	2	30	24	6	560
	0.25	1	4	2	40	33	7	621

Table 5.3 – Effectiveness of the path generation algorithm for SL.

Tables 5.3 and 5.4 show that the relative efficiency of FTOP[<] and of FTOP is instance-dependent. Still, FTOP (i.e., the PG algorithm), despite the necessity of solving the binary PP subproblems, is faster than FTOP[<] (predefining the admissible set of paths plus application of the linear solver) for 18 out of 24 instances tested, including 4 instances (*pdh* for DL) for which FTOP could not solve the problem within the time limit. The results also show that a significant fraction of the PG algorithm execution time is spent in the pricing problem.

5.2.2 Evolution of the PG iterations

An important aspect of the PG algorithm is the number of added paths and the evolution of the network cost. Figures 5.11-5.13 illustrate an important aspect of the PG algorithm, namely the evolution of the number

network	α	β	$t_{FTOP<}$	t_{GP}	t_{FTOP}	t_{PP}	#iter.	#gen.paths
<i>pdh</i>	0.5	0.75	*	*	2914	2506	13	354
	0.5	1	*	*	3612	3034	14	391
	0.25	0.75	*	*	1882	1524	10	354
	0.25	1	*	*	2653	2202	10	354
<i>polska</i>	0.5	0.75	539	54	345	286	8	466
	0.5	1	667	53	387	317	8	486
	0.25	0.75	425	55	858	746	8	511
	0.25	1	303	61	1193	1038	10	530
<i>nobel-us</i>	0.5	0.75	1798	162	755	649	8	542
	0.5	1	2804	140	2187	1925	10	572
	0.25	0.75	1999	160	1719	1444	9	615
	0.25	1	2741	137	2494	2020	11	646

Table 5.4 – Effectiveness of the path generation algorithm for DL.

of added paths $|P^+|$ and the network cost C_{FT} for the consecutive iterations. The two α cases for SL with $\beta = 1$ are illustrated for *pdh*, *polska* and *nobel-us*. Practically, for all these cases the optimal solution is obtained

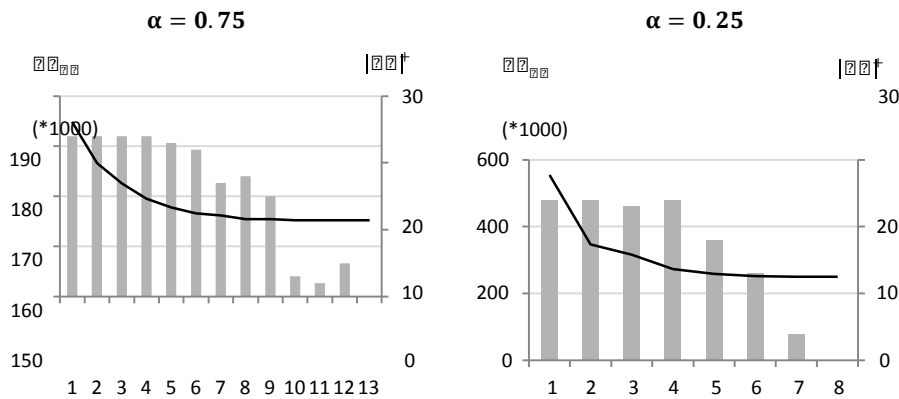


Figure 5.11 – Value of the objective function and number of paths added at each iteration for *pdh*.

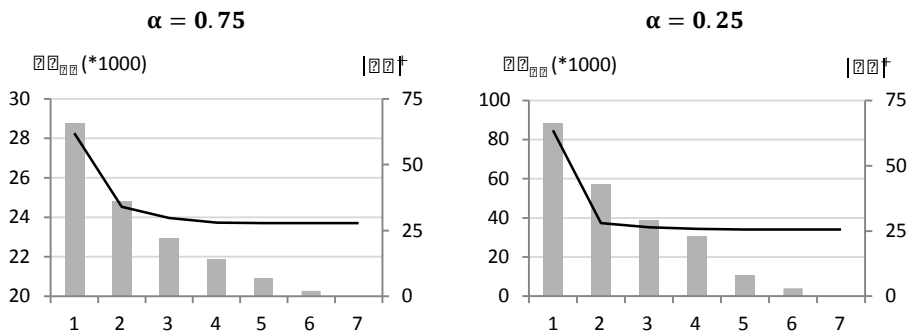


Figure 5.12 – Value of the objective function and number of paths added at each iteration for *polska*.

after only a few (approximately 7) PG iterations. Furthermore, the gap between the current solution and the optimal solution decreases drastically after only 1 or 2 PG iterations. Remind that the network instances are small networks and no conclusions can be drawn in general.

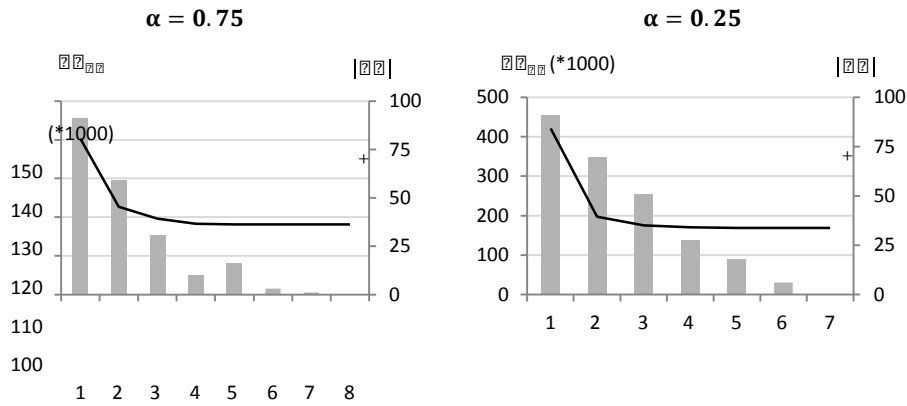


Figure 5.13 – Value of the objective function and number of paths added at each iteration for nobel-us.

scenario	network	GR	FT
SL	<i>pdh</i>	5	100
	<i>polska</i>	0	1
	<i>nobel-us</i>	1	2
DL	<i>pdh</i>	2262	*
	<i>polska</i>	6	269
	<i>nobel-us</i>	12	2367
TL	<i>pdh</i>	*	*
	<i>polska</i>	389	*
	<i>nobel-us</i>	281	*

Table 5.5 – Solution times in seconds for different routing variants.

5.2.3 Running times

Our second set of experiments compares the running times and the network costs for the optimization problems corresponding to the two protection strategies considered in this paper, i.e., for FT and GR. The study is performed for the three networks *pdf*, *polska* and *nobel-us* with the pre-defined path-lists containing all paths up to 5 links. Thus, we study the following two variants of protection strategies and their related problems:

- FT (flow thinning): problem (4.1), i.e., FTOP[<]
- GR (global rerouting): problem (4.1) minus constraints (4.1b) and (4.1e) (see Section 4.2.3)

We consider three kinds of state scenarios: single-link (SL), double-link (DL), and triple-link (TL). SL consists of the nominal state and failures of single links, DL contains SL plus failures of pairs of links, and scenario TL consist of DL and failures of triples of links.

Table 5.5 reports the running times in seconds, taking the averages of the times corresponding to the two cases of $\beta = 1$ and $\alpha = 0.5$. We set the time limit to 10 hours—the cases when the timeout occurred are denoted by *. Table 5.5 reveals that the running times for the general flow thinning procedure (FT) are highly impacted by the number of the simultaneously failing links. In particular, all instances assuming TL could not be solved to optimality within the time limit.

5.3 A lower bound formulation

Since FTOP is NP -hard, it is worthwhile finding a polynomial problem that provides a lower bound for its cost $C(\hat{r})$.

5.3.1 LBP – Lower Bound Problem

A compact linear program, referred to as LBP, whose solutions can be used as the lower bound of FTOP without the necessity of generating paths is as follows.

$$\text{LBP: } \min C^t = \sum_{e \in E} \xi_e y_e^0 \quad (5.1a)$$

$$\sum_{e \in \delta^+(v)} x_{ed}^0 - \sum_{e \in \delta^-(v)} x_{ed}^0 = 0, \quad d \in D, v \in V \setminus \{o(d), t(d)\} \quad (5.1b)$$

$$\sum_{d \in D} x_{ed}^0 \leq y_e^0, \quad e \in E \quad (5.1c)$$

$$\sum_{e \in \delta^+(o(d))} x_{ed}^s - \sum_{e \in \delta^-(o(d))} x_{ed}^s \geq h_d^s, \quad d \in D, s \in S \quad (5.1d)$$

$$\sum_{e \in \delta^+(v)} x_{ed}^s - \sum_{e \in \delta^-(v)} x_{ed}^s = 0 \quad (5.1e)$$

$$d \in D, v \in V \setminus \{o(d), t(d)\}, s \in S$$

$$x_{ed}^s \leq x_{ed}^0, \quad e \in E, d \in D, s \in S \quad (5.1f)$$

$$\sum_{d \in D} x_{ed}^s \leq \alpha_e^s y_e^0, \quad e \in E, s \in S_e \quad (5.1g)$$

$$\text{all variables } x, y^0 \text{ continuous and nonnegative.} \quad (5.1h)$$

Note that for any demand $d \in D$, the value of flow $x_d^0 = (x_{ed}^0, e \in E)$, i.e., $\sum_{e \in \delta^+(o(d))} x_{ed}^0 - \sum_{e \in \delta^-(o(d))} x_{ed}^0$, does not have to be explicitly specified in the formulation.

5.3.2 Comparing LBP and FTOP

The solution of formulation (5.1) is a lower bound for FTOP because all constraints of (5.1) are obviously fulfilled by the link-flows defined by any feasible solution $x_{dp}^0, x_{dp}^s, d \in D, p \in \hat{P}_d, s \in S$ of (4.1), that is, by the link-flows

$$x_{ed}^0 := \sum_{p \in \hat{P}_d} x_{dp}^0, \quad x_{ed}^s := \sum_{p \in \hat{P}_d} x_{dp}^s, \quad e \in E, d \in D, s \in S.$$

Conversely, an optimal solution $x_{ed}^0, x_{ed}^s, e \in E, d \in D, s \in S$, of (5.1) could be translated to an optimal solution of FTOP only if there existed elementary path-flows that would realize the nominal link-flows x_{ed}^0 , and at the same time could be appropriately thinned to realize the state-dependent link-flows x_{ed}^s for each state s .

However, this is in general not possible and hence optimal LBP solutions may in this sense be infeasible for FTOP already for the single-link failure scenarios (Section 4.6.1, Example 4.5) and for the single-node failure scenarios (Section 4.7.2, Example 4.6).

We evaluated the lower bound of FTOP resulting from compact formulation (5.1) for state scenarios SL, DL and SN (single-node failures) for the four cases of (α, β) considered in Section 5.2.1. As the tests were made for undirected networks, we implemented a version of formulation (5.1) with undirected links.

It turned out that for SL and SN, the cost of the LBP solution in all the considered network was smaller than the optimal cost of FTOP by 2–8 percent. The solution times of both approaches were comparable for SL. In the case of SN, however, the running time for LBP was much longer than that of the PG algorithm for FTOP—the LBP running time was always at least 30 times longer.

A comparison of LBP and FTOP in terms of the network cost and the running time assuming DL is presented in Tables 5.6 and 5.7 for *polska* and *nobel-us*, respectively. The tables show the ratios C_{FTOP}/C_{LBP} and T_{FTOP}/T_{LBP} . The quantities C_{FTOP} and C_{LBP} are the optimal costs for FTOP and for LBP, respectively, while T_{FTOP} and T_{LBP} are the respective running times. The results indicate that for the DL case the LBP cost is only marginally smaller than the FTOP cost, and that LBP becomes extremely time consuming. The reason for that large difference in running times is two-fold. First, assuming the common case $|D| \sim |V|^2$ and $|E| \sim |V|$, in the node-link formulation (5.1) the number of variables and constraints is of the order of $O(|V||D||S|) = O(|V|^3|S|)$. At the same time, for a given set of admissible paths P with $|P_d| \leq P, d \in D$ (where P is bounded and does not increase with the size of the network), the FTOP formulation (4.1) requires $O(|P||S|)$ variables and $O(|P||S|) = O(|P||S|)$ constraints, that is a lot less. Second, typically LBP requires much more Simplex iterations than FTOP. This is because for a given state $s \in S$ the number of thinned link-flows in LBP (for every link with limited availability in s all its non-zero demand specific link-flows are typically thinned) is in general much larger than the number of thinned path-flows in FTOP (only the non-zero path-flows through the links with limited availability in s are thinned). For example, for *polska* limited to 4 demands LBP stopped after 2053 iterations while FTOP required only 13 to 52 Simplex iterations at each of the 7 PG iterations, i.e., less than 365. Perhaps it would be possible to improve the solution time of LBP by applying a decomposition method such as Benders' decomposition. Nevertheless, this is out of the scope of this work.

α	β	C_{FTOP}/C_{LBP}	T_{FTOP}/T_{LBP}
0.25	0.75	1.04	0.011
0.25	1	1.05	0.013
0.5	0.75	1.08	0.017
0.5	1	1.07	0.029

Table 5.6 – LBP vs. FTOP for *polska* under DL.

Finally let us address the question raised in Section 5.3: why we consider FT instead of an LBP-based strategy (i.e., a kind of link-flow thinning) despite the fact that in general solutions of FTOP are strictly more expensive than solutions of LBP. The reason is three-fold.

First, as discussed earlier in this section, in general the state-dependent link-flows optimized with LBP do not correspond to any path-flows (tun-

α	β	C_{FTOP}/C_{LBP}	T_{FTOP}/T_{LBP}
0.25	0.75	1.02	0.03
0.25	1	1.02	0.01
0.5	0.75	1.04	0.007
0.5	1	1.03	0.002

Table 5.7 – *LBP vs. FTOP for nobel-us under DL.*

nels) that obey the assumptions of FT. Thus, the simple tunnel thinning and traffic control mechanism that with FT is performed at the traffic/tunnel originating nodes cannot be directly applied. Certainly, a link-flow thinning mechanism could be applied at the link originating nodes—this, however, would most likely lead to a substantially more complicated traffic control mechanism as compared to FT.

Second, as shown above, the cost advantage of LBP over FTOP is only marginal. In addition, as discussed above, except for small networks the running time required to solve LBP is significantly larger than that required by the PG algorithm of FTOP.

Third, as shown in Example 4.5, in general the optimal nominal link-flow x^0 delivered by LBP must contain loops. This can cause serious problems for traffic routing protocols.

FTS - Applications, implementation and practicability

Contents	
6.1	Affine versions of FTS 81
6.1.1	Affine flow thinning and the related version of FTOP . . . 82
6.1.2	Restricted affine flow thinning 83
6.2	Implementation issues 84
6.2.1	Implementation issues of FT 84
6.2.2	Practicability of FTOP and its affine version 85
6.3	Thickening versions of FTS 87
6.3.1	Flow Adjustment – The flow thickening extension of FTOP 87
6.3.2	Performance of Flow Adjustment 88
6.3.3	Why Flow Adjustment should be avoided 89
6.4	Concluding remarks 92

Practicability is of high importance when dealing with routing strategies. In this chapter, we deal with implementation issues of flow thinning. We first present, in section 6.1, a version of FTS called *affine flow thinning* (AFT) that reduces the number of failure states for which the strategy needs to be optimized. Next, we analyse implementation issues related to FTS and AFT. Then, in section 6.2 we present an extension of FTS denoted *flow adjustment* (FAS) that allows for thickening. This evolution leads to a more cost-efficient strategy but the management cost and other issues prevent us from adding this feature to FTS. Finally, we conclude this chapter in section 6.4 with some remarks on possible evolution of FTS.

6.1 Affine versions of FTS

As discussed in more detail in Section 6.2, the practical usefulness of the FT strategy in its form presented so far is limited for three reasons. First, broadcasting of the current link capacity information in real time could

be difficult. Next, it would be difficult to solve FTOP (and to store the resulting solution) for an excessive (exponential) number $|S|$ of states. Finally, even if we managed to achieve an optimal solution (and could store it) for a given (reasonably large) set of states S , for the valid states outside S the thinning would be undefined. The flow thinning strategy FT and its optimization model FTOP presented in the previous sections assume a predefined finite list of states S characterized by $\alpha^s, s \in S$, where $\alpha^s = (\alpha_e^s, e \in E)$. The considered states consist of multiple partial link failures and this means that the potential number of states is enormous, of the order of $2^{|E|}$ already for total link failures. To avoid potential intractability of FTOP, we have so far (see the beginning of Section 4.4) assumed the number of states in S to be polynomially bounded by the number of nodes and links; as we know, FTOP is NP -hard already for this case. Hence, what we can actually do in practice is to solve FTOP for a reasonable (for sure polynomial, but even then for not too large) number of states, and activate proper flow thinning (having the flow thinning factors stored in the originating nodes) when a particular predefined state α^s of links is detected and made known, by some signaling protocol, to the originating nodes of the demands, and using approximated thinning factors when a state not considered in FTOP occurs.

Because of that, in this section we will consider a version of FT called *affine flow thinning* (AFT) that addresses the above issues. In particular, we will study a variant of FTOP (called AFTOP) corresponding to AFT. AFTOP relies on using the decision rules largely used in multi-stage robust optimization, and more particularly, affine decision rules (Ben-Tal et al. 2004) (application of affine decision rules to traffic routing were introduced by Ouorou and Vial (2007) and further investigated by Poss and Raack (2013)). In essence, AFTOP turns out to be nothing else but a particular example of an adjustable robust linear program, where the uncertainty set is S and the adjustable variables are x_{dp}^s . Hence, rather than letting the adjustable variables depend on each specific state s , following Ben-Tal et al. (2004) in AFTOP we let these variables depend affinely on the parameters that describe the state: $\alpha_e^s, e \in E$. We adapt these ideas by considering two special types of FT described in Section 6.1.1 and Section 6.1.2 below.

Notice that AFT and its variants were mostly studied by Michael Poss, Michal Pioro and Dritan Nace in (Pioro et al. 2015). Hence, we will only briefly introduce AFT.

6.1.1 Affine flow thinning and the related version of FTOP

With AFT, the flows x_{dp}^0 reserved on paths $d \in D, p \in P_d$, are thinned to x_{dp}^s for the states $s \in S$ not in an arbitrary way, as with FT, but by assumption each such flow is thinned according to its individual affine function of the availability coefficients α^s . More precisely,

$$x_{dp}^s = A_{dp}(\alpha^s) = z_{dp}^0 + \sum_{e \in E} z_{dp}^e \alpha_e^s, \quad d \in D, p \in P_d. \quad (6.1)$$

Note that parameters $z_{dp}^0, z_{dp}^e, e \in E$, of the affine function A_{dp} specified for the given path $p \in P_d$ do not depend on the state s . The AFT version of

FTOP (formulated for FT in (4.1)) is referred to as AFTOP. In AFTOP, the parameters z become optimization variables and variables x^s play only an auxiliary role. Such a formulation is obtained by adding equations (6.1) to (4.1).

Observe also that with appropriately limited cardinality of S , traffic efficiency of AFT can be very close to traffic efficiency of FT. In fact, as shown in Poss and Raack (2013) for a similar problem, AFT becomes equivalent to FT if, and only if, S contains at most $|E| + 1$ states.

In the implementation aspect, AFT is more practical than FT. Although as FT, AFT must be aware of the current link states α^s , it will use the (optimized) affine functions A_{dp} , $d \in D$, $p \in P_d$, for thinning the nominal flows. Such thinning, contrary to FT, is applicable to all possible states, i.e., also to the states not in the set S assumed for optimization.

In spite of the above advantages, AFT suffers from two problems. First, introduction of affine coefficients (variables) z_{dp}^0, z_{dp}^e , $e \in E$, for each path $p \in P_d$, $d \in D$, and each link $e \in E$, leads to large linear programs for AFTOP, both when included in formulation (4.1). Second, AFT can still be hard to implement because the affine function of each path p involves, as arguments, the link availability coefficients of all links. Thus, disseminating the link-status information is as difficult as for FT.

6.1.2 Restricted affine flow thinning

As AFT, *restricted affine flow thinning* (RAFT) assumes that the flow on each path for each state is an affine function of the availability coefficients α_e^s , but only of the links that belong to path p :

$$x_{dp}^s = A_{dp}(\alpha^s(p)) = z_{dp}^0 + \sum_{e \in p} z_{dp}^e \alpha_e^s, \quad d \in D, \quad p \in P_d, \quad (6.2)$$

where $\alpha^s(p) = (\alpha_e^s, e \in p)$. Clearly, RAFT involves less variables z than AFT (an advantage), but it is not necessarily as efficient as FT, in particular for sets S with $|S| \leq |E| + 1$ (a disadvantage).

The most important advantage of RAFT over AFT (and for that matter, over FT as well) is that with RAFT disseminating link capacity information is, not a major issue, as a path-originating node thins the corresponding path-flow only on the basis of the current capacities of the links along the path. In fact, we could easily use even more information on the link availability states, namely the thinning function of a path-flow could be made dependent on coefficients α_e^s of all the links e belonging to the sets $\delta(v)$ for all nodes v traversed by the considered path, i.e.,

$$x_{dp}^s = A_{dp}(\underline{\alpha}^s(p)) = z_{dp}^0 + \sum_{e \in E^t(p)} z_{dp}^e \alpha_e^s, \quad d \in D, \quad p \in P_d, \quad (6.3)$$

where $\underline{\alpha}^s(p) = (\alpha_e^s, e \in E^t(p))$, $E^t(p) := \bigcup_{v \in V(p)} \delta(v)$, and $V(p)$ is the set of all nodes visited by path $p \in P_d$.

6.2 Implementation issues

6.2.1 Implementation issues of FT

A major implementation issue related to FT is to ensure proper online tunnel capacity control. Assuming that the nodes can instantaneously determine and apply appropriate *thinning factors* (thinning factor is the ratio of the tunnel capacity calculated for a given state to its nominal capacity) once the link availability state is made known to them, we need an on-going online signalling process of informing the network nodes about the changes in the current link availability coefficients. For the general version of FT this would call for a network protocol, like OSPF-TE (Open Shortest Path First – Traffic Engineering) (Katz et al. 2003), enabling the originating node of a link to broadcast a link-status message whenever the capacity of the link is changed. A difficulty is to make the message reach all the nodes in a reasonable time.

This issue can be overcome by applying RAFT (see Section 6.1.2) instead of FT. With RAFT, the thinning factors of each tunnel are specified by means of an affine function depending only on the availability coefficients of the links along the tunnel. Then the link-status message concerning a change of the availability coefficient of a particular link is sent by its originating node backwards along all the tunnels that traverse this node and use the link in question. In this way, by means of appropriately extended path-error messages in Resource Reservation Protocol – Traffic Engineering (RSVP-TE, the protocol from the TCP/IP stack that is used to establish, maintain, and erase LSPs, see Minei and Lucek (2011) and Awduche et al. (2001)), the nodes can be quickly informed (at least when the links are bidirectional—a typical case in WMN) about the changes in the links capacity they need to be aware of. (Observe that since a tunnel is thinned to 0 already when only one of its links fails totally, it is enough that the originating node of a tunnel is informed only about the total failure of its closest link.) Hence, the crucial mechanism for FT could be implemented in a network composed of IP routers interconnected by IP links with logical tunnels (LSP) realized by means of the MPLS mechanism (see Minei and Lucek (2011) and Andersson and Swallow (2003)).

Next, FT requires a low-level QoS packet admission control mechanism (including packet scheduling and policing), see Evans and Filsfils (2007) and Harhira and Pierre (2007) to assure, for each demand, that the instantaneous packet flow admitted to enter each of its dedicated tunnels does not exceed the current tunnel capacity. The users that compose a demand, in turn, could be notified (using some kind of explicit congestion notification mechanism, see Evans and Filsfils (2007)) about the current admitted-to-nominal traffic ratio. Then, the users could decide which sessions to realize so that the combined demand traffic would be served in an efficient way, minimizing the number of packets discarded by the demand originating node admission control.

Another issue is that with FT in general only a fraction of all possible link availability states can be explicitly considered in optimization, simply because the number of states can be exponential. Hence, the thinning factors for a non-considered state must be somehow calculated at

the tunnel originating nodes, having in mind that the resulting tunnel capacity cannot overload the links and the reduction of traffic is acceptable. For the general version of FT, finding a reasonable algorithm for that would be difficult but this task becomes tractable with RAFT (and, for that matter, also for AFT): for the non-considered states, the thinning factor for a given tunnel is automatically determined by the affine function once availability coefficients for its links are known. Moreover, the affine functions, even when optimized using a limited number of states, would tend to ensure reasonable thinning factors also for the states not explicitly considered in optimization. This is particularly sound when thinning is based on the non-negative value of an affine function: $x_{dp}^s = \max\{0, x_{dp}^0 - \sum_{e \in p} z_{dp}^e (1 - \alpha_e^s)\}$, $d \in D$, $p \in P_d$, where all $z_{dp}^e \geq 0$, $e \in E$.

6.2.2 Practicability of FTOP and its affine version

In our optimization model each link availability coefficient α (for a given link and a given state) is a fixed parameter while the nominal capacity y of the link is an optimization variable. Using the availability coefficients to determine the state-dependent link capacity as $\alpha \cdot y$ is a sound assumption for the following reason. In reality, in most cases link capacity is modular, i.e., equal to $M \cdot y$ where y is a number (an integer variable) of modules of size M installed on the considered link. Referring to the application examples discussed in Section 1.1.1, in the IP/MPLS over wireless case, where the modules correspond to parallel microwave radio links or wireless optical links, coefficient α corresponds to the modulation and coding scheme applied in the considered state. In the IP/MPLS over DWDM case, α is the percentage of modules (for example, 0%, 33%, 66%, 100%) that are not lost in the considered link as a result of the failure in the DWDM layer. That is, we assume the equal split of y modules to be realized over a given set of k disjoint DWDM paths ($k = 3$ paths in our example), whatever the value of y (provided y is divisible by k , to be precise). Note that capacity is linearized and variables y are made continuous to avoid unnecessary complication of the optimization model.

The state description assumed for FTOP is on purpose quite general. The implementation of FT described in Section 6.2, however, assumes that the demand reduction coefficients depend only on the vector α^s (i.e., $\beta^s = \beta(\alpha^s)$, $s \in S$), so that the state is in fact determined by the current vector α^s of link availability coefficients. This kind of state description, together with the assumption that the demands will accept reduction in their nominal traffic volumes in the states with significant link capacity reduction, opens a way to specify reasonable demand reduction coefficients by the network operator to avoid substantial network over-dimensioning, and hence excessive network cost. Clearly, the degree to which the preferable traffic volume is decreased should reflect the fractions of the nominal link capacity available in a given state.

A simple way for achieving reasonable reductions is to divide the set of states into several classes corresponding to several degrees of lost capacity, and to assume a certain percentage of the nominal traffic that should be carried for each demand in a given state. For example, we may distinguish

two classes of states, one of “almost nominal” states with the entire (100%) nominal traffic to be carried, and the second class of “degraded states” that allow for carrying only 75%, say, of nominal traffic for each demand. A more sophisticated way, applicable in the affine version of FTOP, is to relate the demand reduction coefficients to the link availability coefficients through the so called uncertainty polytope.

Another important practical issue is how to select a representative set of states of reasonable (tractable) size to be assumed for FTOP (or AFTOP). These states should, on one hand, represent a whole range of typical states, and, on the other hand, be sufficient to effectively approximate the thinning factors in the remaining feasible states (see Section 6.2). Such a selection is network dependent. For the IP/DWDM case the selection is basically simple. As it is a common practice to consider only single optical cable cuts (simultaneous cuts of more than one cable are very unlikely), the resulting multiple partial IP link failures are easy to list and include in FTOP. In the case of WMN, the situation is more complicated. For modeling link availability states implied by weather conditions in a metropolitan WMN, we could divide the nodes into disjoint groups and assume that the bad weather conditions affects either all of them or a subset of them at a time. If we distinguish only several groups then we are able to consider all the states corresponding to all subsets of the family of these groups, i.e., the partial failures (with different levels of link availability, depending for example on the rain intensity) of all nodes in the affected groups. The demand reduction coefficients would then depend on the total number of affected nodes and (demand-wise) on the position of the demand’s end nodes. This issue, however, requires a separate study.

For everyday network operation a pure flow allocation version of FTOP (or AFTOP) becomes important. Such a problem (let us call it FTOP/FA) arises when nominal link capacities are fixed, i.e., y^0 become given parameters. FTOP/FA has to be considered whenever the nominal traffic matrix (assumed for FTOP) is significantly perturbed causing the nominal flows and the thinning factors (tunnel flow-defining affine functions in the case of AFT) not fit the new traffic matrix anymore. Hence, the nominal flows and thinning factors have to be re-optimized. In order to properly accommodate the new traffic by the network, some kind of fairness criteria (link max-min fairness or proportional fairness (see Chapter 8 in Pióro and Medhi (2004b)) should be imposed on the values of h_d^s , $d \in D$, $s \in S$ (these quantities would become optimization variables on top of the tunnel flows), and reflected by an appropriate objective function used instead of (4.1a).

Observe that optimization of FTOP does not have to be decentralized since it could be done once for all in a central facility before the network is set to operation. FTOP/FA, in turn, should be resolved, using an off-line optimization algorithm, whenever the nominal traffic matrix (assumed for FTOP) is significantly perturbed. When perturbations are not frequent, such optimization could be done in a central facility as well. Otherwise, the computation should be decentralized and distributed among the network nodes. This would require some information exchange mechanism, for example an extension of the OSPF-TE. Such an algorithm, however, is outside the scope of this paper.

6.3 Thickening versions of FTS

A natural extension of FTS, and thus of FTOP denoted Flow Adjustment (FAS) (Fouquet et al. 2014a;c), allows for tunnel thickening limited by additional parameters $t_d \geq 1, d \in D$, called the *thickening bounds*.

6.3.1 Flow Adjustment – The flow thickening extension of FTOP

The extension is obtained by modifying constraint (4.1e):

$$x_{dp}^s \leq t_d x_{dp}^0, \quad d \in D, p \in P_d, s \in S. \quad (6.4)$$

Hence, FAS is represented by the following link-path linear programming (LP) formulation involving variables $x_{dp}^0 (d \in D, p \in P_d)$, $x_{dp}^s (d \in D, p \in P_d, s \in S)$ and $y_e^0 (e \in E)$:

$$P_{FAS}(P): \quad C(P) = \min \sum_{e \in E} \xi_e y_e^0 \quad (6.5a)$$

$$[\pi_e^0 \geq 0] \quad \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \leq y_e^0, \quad e \in E \quad (6.5b)$$

$$[\lambda_d^s \geq 0] \quad \sum_{p \in P_d} x_{dp}^s \geq h_d^s, \quad d \in D, s \in S \quad (6.5c)$$

$$[\pi_e^s \geq 0] \quad \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^s \leq \alpha_e^s y_e^0, \quad e \in E, s \in S_e \quad (6.5d)$$

$$[\sigma_{dp}^s \geq 0] \quad x_{dp}^s \leq t_d x_{dp}^0, \quad d \in D, p \in P_d, s \in S \quad (6.5e)$$

$$x^0, x^s, y^0 \geq 0 \text{ and continuous} \quad (6.5f)$$

Objective (6.5a) minimizes the overall cost of links. The first capacity constraint (6.5b) makes sure that the nominal link loads do not exceed the nominal link capacities. Next, the demand constraint (6.5c) assures that in each state $s \in S$, the thinned nominal flows are sufficient to realize the volume of each demand $d \in D$ assumed for this state. Then, the second capacity constraint (6.5d) assures that in each state $s \in S$, the available capacity of each link $e \in E_s$ is not exceeded. Finally, constraint (6.5e) assures thinning.

Clearly, with $t_d > 1$ a (limited) path-flow thickening for demand $d \in D$ is permissible, not only thinning. In fact, this modification (even when the thickening factor is made tunnel-dependent) has virtually the same properties as FTOP and can be approached essentially in the way discussed in Section 4.4 as done in (Fouquet et al. 2014a).

When the thickening bounds $t_d, d \in D$, are considered (constraint (4.1e) is substituted by (6.4)) then constraint (4.2c) in the related dual problem takes the form

$$\sum_{s \in S} \sigma_{dp}^s \leq \frac{1}{t_d} \sum_{e \in E} \delta_{edp} \pi_e^0, \quad d \in D, p \in P_d \quad (6.6)$$

while the rest of the constraints remain unchanged. When all $t_d \rightarrow \infty$, i.e., when the primal problem becomes the GR optimization problem, the right-hand side of constraint (6.6) becomes equal to 0 which forces all

dual variables $\sigma_{dp}^s, s \in S, d \in D, p \in P_d$, to be equal to 0. Hence, constraint (4.2d) takes the form

$$\lambda_d^s \leq \sum_{e \in E_s} \delta_{edp} \pi_e^s, \quad s \in S, d \in D, p \in P_d \quad (6.7)$$

and this makes the related dual separation (and path generation) problem polynomial. This could be expected, since the GR problem is, as mentioned in Section 4.2.3, polynomial.

6.3.2 Performance of Flow Adjustment

Although in general FAS is designed to handle multiple partial link failures, below we focus on the single link failures to illustrate the performance of the strategy compared to FTS.

network	$ V $	$ E $	$ D $
di-yuan	11	42	22
polska	12	18	66
nobel-us	14	21	91
nobel-germany	17	26	121

Table 6.1 – Network description

In Table 6.1 we describe the undirected network instances used in our tests taken from (Orlowski et al. 2010). We consider several scenarios with different values of the link availability coefficients α and the thickening coefficients t_d . In all cases, we assume 100% traffic restoration. In Figures 6.1-6.4, we compare the GR, FT and FA strategies with respect to their cost-effectiveness. In the comparisons, $C_{FTS}^\alpha, C_{FAS}^\alpha$ and C_{GR}^α denote the optimal value of the network cost for FAS and GR respectively. Remind that considering $t_d = 1, C_{FTS}^\alpha = C_{FAS}^\alpha$. Gap_{FAS} and Gap_{GR} expresses the

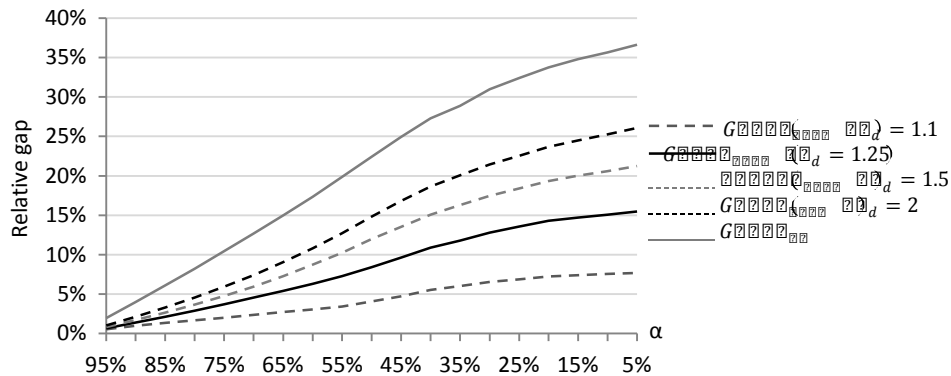


Figure 6.1 – Relative gap between FAS and GR for di-yuan.

relative gain of cost of C_{FAS}^α and C_{GR}^α , respectively, with respect to C_{FTS}^α for a given availability ratio α . Consider strategy S (S equals to FAS or GR), $Gap_S = (C_{FTS}^\alpha - C_S^\alpha) / C_S^\alpha$.

These figures shows that the gain of cost is limited for low intensity perturbation. However, the gain of cost becomes significant for strong perturbation with a gain of at least 15% for a thickening limit fixed to 25% ($t_d = 1.25$) for all networks. Notice that for sparse networks (*polska, nobel-us*), allowing to double the bandwidth leads to dimensioning

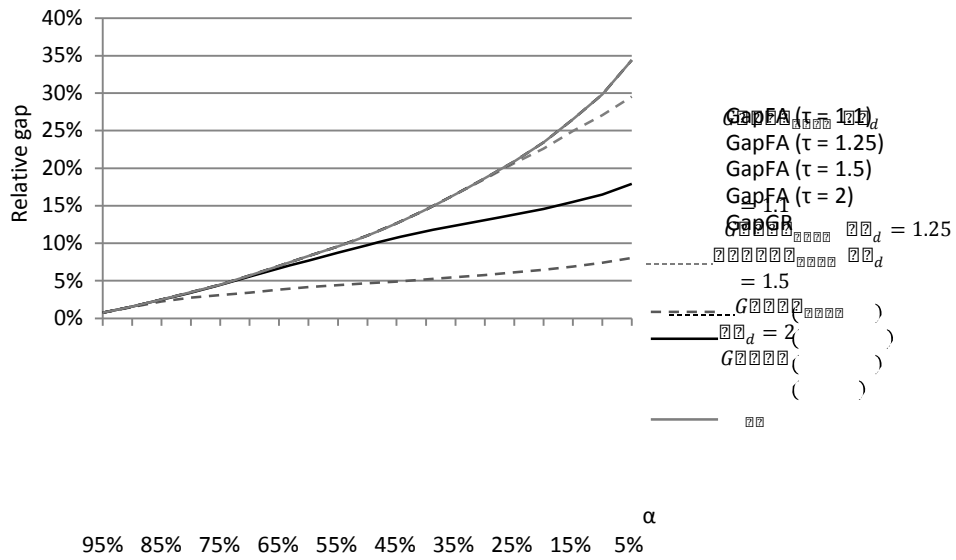


Figure 6.2 – Relative gap between FAS and GR for polska.

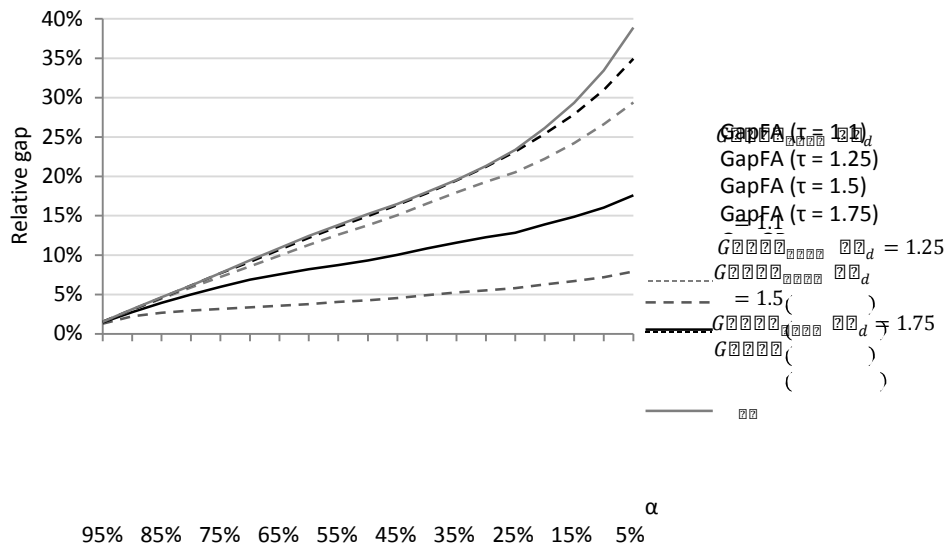


Figure 6.3 – Relative gap between FAS and GR for nobel-us.

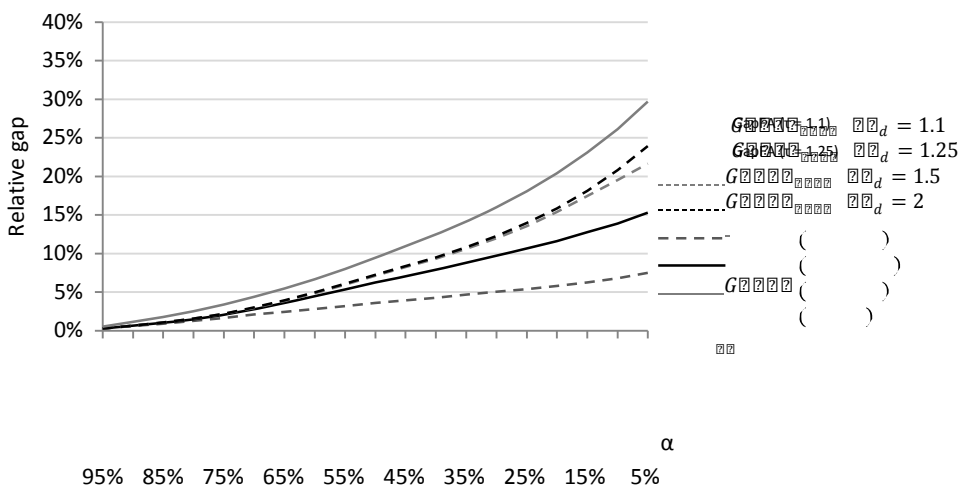


Figure 6.4 – Relative gap between FAS and GR for nobel-germany.

cost close to Global Rerouting. However, the gap between C_{FTS}^α and C_{GR}^α remains high for meshed networks (*di-yuan*). Note that more results of the cost-efficiency aspect of FAS are presented in (Fouquet et al. 2014a) and (Fouquet et al. 2014c). It is clear that Flow Adjustment is a lot more

cost-efficient than Flow Thinning.

6.3.3 Why Flow Adjustment should be avoided

Flow thickening leads to cheaper solutions as compared to pure thinning, and when all $t_d \rightarrow \infty$ the resulting modification of FTOP becomes equivalent to the GR optimization problem. As the corresponding modification of FT allowing for thickening can be implemented in the same way as the original FT (including the affine versions, see 6.1), and thickening can im-

prove traffic performance of FT, a natural question arises why we still opt for FT. The three main reasons are described in the following.

6.3.3.1 Higher number of paths to be adjusted

First, with FT the maximum number of flows that are adjusted when moving from state s_1 to state s_2 is equal to the number of flows that traverse links affected in state s_1 plus the number flows that traverse links affected in state s_2 . This number is quite small as compared to the number of all (non-zero) nominal flows. However, with thickening this bound is not valid and the number of adjusted flows can become significant. In general, the larger the thickening bound the more tunnel flows are adjusted, and this number is typically the largest for GR (with GR frequently virtually all flows can be adjusted even for modest changes in link capacity). Moreover, the scale of the perturbation, i.e., the ratio of the number of nominal path-flows changed when a failure occurs is increased in FAS. While for FTS the perturbation observed only for a number of paths ranging from 30% to 45%, the scale of the perturbation is much larger for FAS. We have notice that for all the single link failure states with $\alpha = 0.5$, almost all routing paths are subject to thinning or thickening (more than 98% of the total number of routing paths). All of this is shown in the following Table 6.2 which reports the average percentage of the routing paths required to be thinned () or thickened () per failure for FAS (remind that $t_d = 1$ represents FTS).

network	$t_d = 1$	$t_d = 1.1$	$t_d = 1.5$	$t_d = 2$				
di-yuan	0	19	69	30	55	44	38	61
polska	0	44	56	44	57	43	43	57
nobel-us	0	41	60	40	58	42	46	54
nobel-germany	0	48	52	48	49	51	40	60

Table 6.2 – Average percentage per link failure of nominal paths which are thinned or thickened in FAS

Even though this number clearly shows the advantage of using FT, the values of percentages. Thus, they must be reported to the actual number of paths in the optimal solution. In the following Figures 6.5 and 6.6, we represent the number of distinct paths per demand used by FAS for partial link failures ($\alpha = 0.5$) and for total link failure ($\alpha = 0$).

We first notice that FTS requires less paths than FAS, and that this difference increases with t_d . Moreover, except for *di-yuan*, the number of routing paths required by FAS for total single link failures is similar to the number of paths required for partial single link failure with $\alpha = 0.5$.

6.3.3.2 Transient behavior of the flow adjustment process

The main interest in the flow-thinning strategy stems from the simplicity of the path-flow handling process. Indeed, the reaction to a particular availability state basically consists in decreasing the flow on (some of) the affected paths. In the following we assume that a signaling protocol sends a message from the end node of the failed link to the source nodes of the

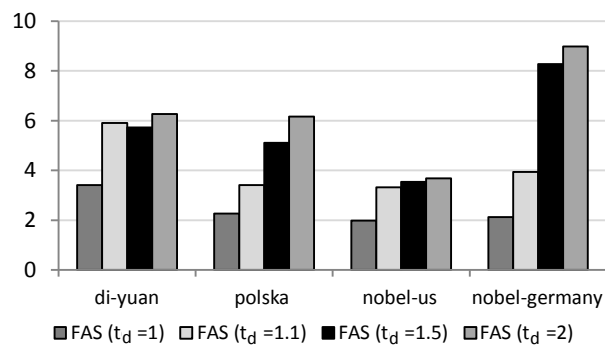


Figure 6.5 – Average number of distinct paths per demand used by FAS per single partial link failure ($\alpha = 0.5$).

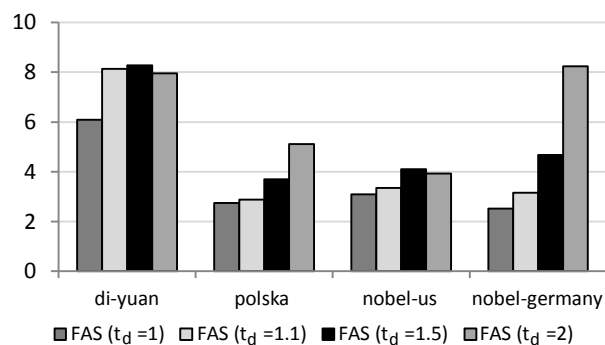


Figure 6.6 – Average number of distinct paths per demand used by FAS per single total link failure ($\alpha = 0$).

routing paths traversing this link when its capacity decreases by a certain fixed threshold. Hence, the time required to recover from a failure state with FTS is equal to the time required for the signaling messages to traverse the path from the end nodes of the perturbed links to the source nodes of the affected demands. Then, upon the receipt of such signaling messages, the source nodes will set/apply the appropriate flow value to their originating routing paths. The flow adjustment process in FAS is not as simple as in FTS. This process is composed of two simultaneous stages managed by the source and destination nodes of traffic demands. The source nodes of the demands will first need to decrease the flow on some affected paths in order to make room for enlarging the flows on some other paths. Then, in the next stage, the destination nodes will be able to increase the flow of the latter paths. To examine the process, we have run appropriate simulations. Figures 6.7 and 6.8 report the ratio of the perturbed demands for a fixed link failure as a function of time for network *polska* with, respectively, $\alpha = 0.5$ and $\alpha = 0$. The considered failed link is contained in the routing paths of 20% of demands that are in this way affected. Nevertheless, the flow adjustment process perturbs more demands, as the nominal path-flows of non-affected demands are in general subject to changes as well. In the process, the flows on a set of paths are first decreased to allow the flows on another set of paths to be subsequently increased. This makes a lot of demands to be temporarily perturbed reaching about 55% (respectively, up to 80%) of the number of

demands after 3 units of time for $\alpha = 0.5$ (resp. $\alpha = 0$). We notice that, as expected, the flow adjustment process is faster for the partial failure than for the total failure. Nevertheless, the process converges very fast as compared with GR. Finally, we notice that the ratio of perturbed demands remains high as compared with that observed for the flow-thinning strategy FTS (no flow thickening).

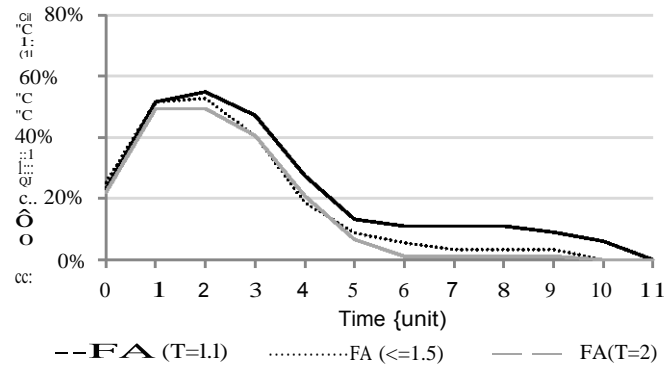


Figure 6.7- Evolution of the ratio of perturbed demands for polska ($\alpha = 0.5$).

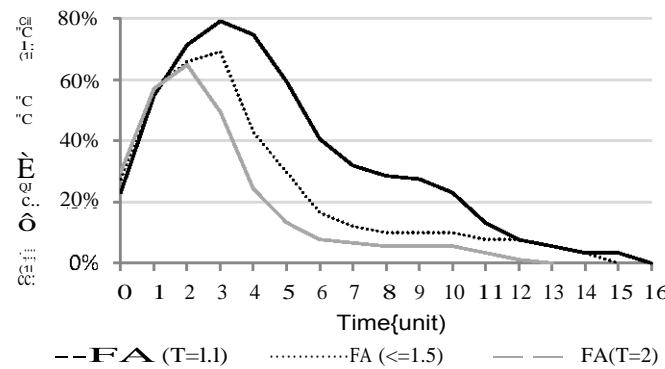


Figure 6.8 -Evolution of the ratio of perturbed demands for polska ($\alpha = 0$).

6-3-3-3 Link overloads

The third reason are link overloads caused by erroneous approximation of flow thickening factors. If a certain state is not considered in optimization, its tunnel thickening factors must be approximated, as it is done in FT for the thinning factors. However, since with thickening the nominal tunnel capacities are allowed to be exceeded in the states with reduced link capacity, wrong approximation of the thickening factors could easily lead to link overloads during the whole duration the unforeseen states. To summarize, as far as unforeseen traffic losses are concerned, FT is safer than its variants admitting flow thickening (including GR).

6-4 CONCLUDING REMARKS

Flow thinning (FT) is an original concept of a traffic routing and protection strategy for communication networks with variable link capacity. Its affine variant, AFT (more specifically, RAFT), opens a way for distributed

implementations. To our knowledge, similar strategies (working at the logical tunnel level and reacting online to link capacity fluctuations) have not been proposed and therefore studying optimization models for FT and AFT is worthwhile, the more that the related multicommodity flow problems contribute to the area of resilient network design and are interesting from the OR viewpoint.

In this chapter, we have studied relations between the investment and management cost and the restoration performance for the considered protection strategies (FAS, FTS, GR). It turns out that the flow adjustment approach of FAS (and FTS) assures acceptable investment and management cost while achieving high restoration performance. Nevertheless, an important issue that needs special care is the volume of information required to be maintained at each routing node that could become problematic in large networks. One way of alleviating this would be to limit the number of path-flows that are adjusted per failure situation. This can be done by making a distinction between affected and non-affected demands. This distinction could help to reduce the number of routing paths affected per failure situation but still the volume of information kept at each node remains almost as large. Thus, a variant of the method needs to be designed for distributed environments. These important interrogations were considered while conceiving a new recovery mechanism named *Elastic Flow Routing*, presented in the next part of this thesis, in the three next chapters.

Part III

Elastic Flow Rerouting

This part is dedicated to the *Elastic Flow Rerouting* (EFR) strategy. In Chapter 7, we introduce the mathematical formulation for total link failure scenarios (EFR-P) together with a complexity study. EFR-P is NP -hard and requires branch-and-price to be solved to optimality due to the distinction between affected and unaffected demands. Thus, we propose a greedy heuristic method to solve EFR. As this thesis deals with both total and partial failures, in chapter 8, we study practical applications of EFR to partial failures. In accordance with microwave communication context, we propose a formulation of the modulation schemes and show next how EFR can be combined with FTS to give a strategy capable to handle both type of failures. Finally, we study a distributed variant of EFR. In the last chapter of this part, Chapter 9, we study the efficiency of EFR and its formulations followed by an analysis of the heuristic method performance. We conclude this part with a study of restoration time in case of failure of both centralized and distributed versions of EFR.

Elastic Flow Rerouting for total link failures

Contents

7.1	Introduction to Elastic Flow Rerouting	97
7.2	Mathematical model of EFR	100
7.2.1	Notation	100
7.2.2	EFR-P formulation	101
7.3	Complexity discussion	102
7.4	Heuristic method	105
7.4.1	Description of the heuristic	106
7.4.2	Formulation of EFR-PF problem	107
7.4.3	Dual problem	108
7.4.4	Pricing problem	109
7.4.5	Binary formulation of the pricing problem	110

In this chapter, we study a method that combines positive features of the strategies discussed in Chapter 2. The method is intended to achieve a compromise between the capacity cost, restoration time, and management effort, while covering a large set of applications. The method is called *Elastic Flow Rerouting* (EFR). Its full study, including a mathematical formulation and a discussion on its complexity, is presented in (Fouquet et al. 2015b). We show that the problem becomes computationally intractable even for moderate size instances, so that there is a need for heuristic approaches. Hence, we propose a heuristic solution method accompanied with numerical results. EFR is dedicated to the management of total link failures. An adapted version of EFT, able to manage partial failures, will be presented in the next chapter.

7.1 Introduction to Elastic Flow Rerouting

Elastic Flow Rerouting (EFR, Fouquet et al. (2014e)) is intended to deal with total link failures. The approach works as follows. In the nominal state (when all links are operational), traffic is routed over a given set of

paths for each traffic demand. In each failure state (to simplify the explanation only total failures of single links are considered), EFR uses the same set of (nominal) routing paths but with possibly modified path-flow values. In each failure state we distinguish between affected flows and affected demands. In a given failure state the affected flows are those routed on paths going through a failing link, and the affected demands are those for which at least one path is affected and the total capacity of unaffected paths is not sufficient to carry the demand traffic. The main idea behind EFR is to restore traffic of the affected demands by means of increasing the flow on their routing paths, possibly together with decreasing flows of the unaffected demands. Thus, an increase of the path-flows is allowed only for the affected demands, and a decrease of path-flows is allowed only for the unaffected demands (besides the failed flows which are automatically removed). Notice that no new paths are allowed to be established. To limit the number of path-flows and prevent from establishing paths with no bandwidth (that could be adjusted upon failure occurrence), the thinning and thickening are bounded by a fixed percentage of their nominal path-flows. Summarizing, flows (on existing paths) can be increases due to:

- capacity released on affected paths
- capacity possibly released on some paths of unaffected demands
- protection capacity present on links due to network over-dimensioning.

Clearly, the capacity released on routing paths of unaffected demands is limited because the remaining flows on the routing paths for the unaffected demands must be sufficient to carry the requested traffic volume.

In short, we can say that EFR adds rerouting capabilities to PD but, unlike RR and GR, seeks to find the capacity needed for rerouting by decreasing some flows on the nominal routing paths of unaffected demands (in addition to the capacity released by the affected flows). Contrary to ShRR, EFR avoid path creation.

Example 7.1 Consider a network with 5 nodes A, B, C, D and E and 8 undirected links $(A, B), (A, C), (A, E), (B, D), (B, E), (C, D), (C, E)$ and (D, E) . The unit capacity cost of all eight links is equal to 1. We consider two demands d_1 and d_2 (each with demand volume equal to 1) between nodes A and E for d_1 and A and D for d_2 . We consider single-link failure scenarios in which each link can fail, but once at a time. We show below what is the result of the minimum link cost network when EFR is used to recover from single link failures.

The obtained solution gives a capacity of $\frac{1}{3}$ to link (B, E) , $\frac{2}{3}$ to link (C, E) , 0 to link (D, E) and 1 for the other five links. In the nominal state, as depicted in Figure 7.1, demand d_1 is realized on 3 paths: d_1 is routed on 3 paths: $p_1 = \{A - E\}$ with 1 flow value, $p_2 = \{A - B - E\}$ and $p_3 = \{A - C - E\}$ with $\frac{1}{3}$ flow value; while d_2 is routed on 2 paths: $p_4 = \{A - B - D\}$ and $p_5 = \{A - C - D\}$ both with $\frac{2}{3}$ flow value.

Figure 7.2 represents the failure of link (A, C) . Due to the failure, the traffic requirement for d_2 is no longer meet and there is no bandwidth remaining on

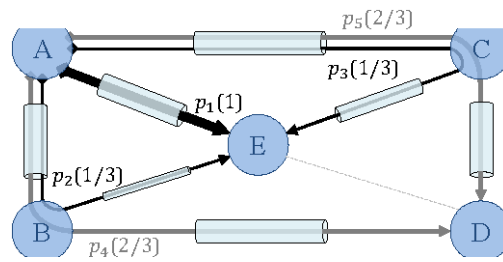


Figure 7-1- EFR- Nominal State.

the link (A,B) to increase the traffic of path P4. Hence, as demand d_1 is not affected by the failure, it releases all the bandwidth of path p_2 (enough bandwidth is remaining of path p_4). Enough bandwidth is now available to increase the bandwidth of P4 by and to meet traffic requirement of d_2 . Then, all demands are fully operational.

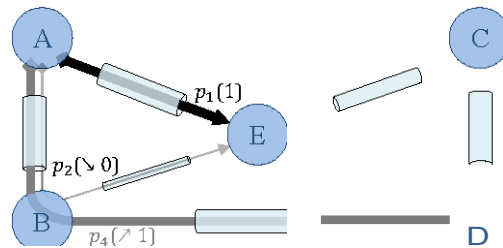


Figure 7.2 – EFR - Failure of link (A,C).

Another failure scenario is presented on Figure 7.3, concerning the failure of link (A,E). The traffic requirement for d_1 is no longer met and there is no bandwidth available to increase the traffic on paths p_2 and p_3 . As demand d_2 is not affected by the failure, it releases of bandwidth on path p_5 to make room for path p_3 thickening. Finally, d_1 increases the bandwidth of paths p_2 by and all demands recovered the failure.

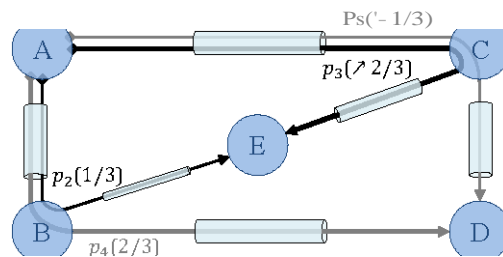


Figure 7-3- EFR- Failure of link (A,E).

The other failure scenarios are simple of similar to the two examples described above. Failures of links (B,E) and (C,E) do not cause any problem for demand d_1 . The other link failures are close to the failure of (A,C). Note the dimensioning cost of 5 is also the optimal cost for both GR and RR strategies.

We recall that differently to EFR :

- GR and RR allow path creation (longer process than bandwidth increase),
- GR suffers from failure propagations

7.2 Mathematical model of EFR

The optimization problem considered in this section is referred to as Elastic Flow Rerouting Problem (EFR-P) and is as follows. We want to minimize the total cost of link capacity assuming that in the nominal state of network operation, all demand volumes are realized by means of nominal path-flows. When the network is subject to a failure of a link then the demand volumes, possibly reduced, are realized for the duration of the failure state by appropriate thinning or thickening of the nominal path-flows depending on the state of their demand. A brief reminder of the notation used in this section is given in 7.2.1 and the detailed formulation of EFR-P is given in Subsection 7.2.2. Section 7.3 is devoted to its complexity study.

7.2.1 Notation

EFR-P requires the use of the following notations:

V	set of routers
E	set of links, directed or undirected, represented by a pair $\{v, w\}$ of some nodes $v, w \in V$
ξ_e	non-negative unit capacity cost of link $e \in E$, seen as a parameter
D	set of demands associated with an unordered pair of nodes $\{o(d), t(d)\}$. For the sake of simplicity we call $o(d)$ a source node and $t(d)$ a termination node.
h_d^0	volume that has to be sent between $o(d)$ and $t(d)$ for demand $d \in D$, seen as a parameter
P_d	set of nominal paths, parameters in the EFR-P formulation
S	set of failure scenario equal to all possible link failure
x_{dp}^0	nominal path-flow associated to path $p \in P_d$ for demand $d \in D$
y^e	capacity reservation for edge $e \in E$, which is an optimization variable
P_d^s	set of paths that are unaffected in failure state $s \in S$
\bar{P}_d^s	set of paths that are affected in failure state $s \in S$, i.e. that contain a link in s
$\bar{\delta}_{edp}$	link-path incidence coefficients ($\bar{\delta}_{edp}, e \in E, d \in D, p \in P_d$). It takes value $\bar{\delta}_{edp} = 1$ if path $p \in P_d$ traverses link $e \in E$, and $\bar{\delta}_{edp} = 0$ otherwise.
\bar{D}^s	affected demands in failure scenario $s \in S$, allowed to thicken their remaining path-flows. A demand is affected if its surviving nominal path-flows are not sufficient to carry the traffic volume assumed for this state, i.e., when $\sum_{p \in \bar{P}_d^s} z_{dp}^s + \sum_{p \in P_d^s} x_{dp}^0 < h_d^s$.

D^s	unaffected demands in failure scenario $s \in S$, allowed to thin their path-flows. A demand is unaffected if, by definition, it is not affected.
u_{dp}^s	value of thinned path-flow associated to path $p \in P_d^s$ for demand $d \in D^s$ in failure state $s \in S$
v_{dp}^s	value of thickened path-flow associated to path $p \in P_d^s$ for demand $d \in D^s$ in failure state $s \in S$
T_d^s	binary variable representing the status (affected/unaffected) of demand $d \in D$. $T_d^s = 1$ if d is an affected demand in s and $T_d^s = 0$ if d is not affected in s .

Note that the thinning/thickening state-dependent path-flows $u_{dp}^s, v_{dp}^s, s \in S, d \in D, p \in P_d^s$, are optimization variables and are bounded by ratio $a \leq 1$ (resp. $b \geq 0$) of their nominal values, or by 0 depending on the status of the demand.

7.2.2 EFR-P formulation

Optimization problem EFR-P is given in the non-linear path-flow formulation (7.1) for a given list of admissible path sets $P_d, d \in D$.

$$EFR-P(P): \min C = \sum_{e \in E} \xi_e y_e \quad (7.1a)$$

$$\sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \leq y_e, \quad e \in E \quad (7.1b)$$

$$\sum_{p \in P_d} x_{dp}^0 \geq h_d^0, \quad d \in D \quad (7.1c)$$

$$\sum_{d \in D} \sum_{p \in P_d^s} \delta_{edp} (x_{dp}^0 - u_{dp}^s + v_{dp}^s) \leq y_e, \quad s \in S, e \in E \setminus \{s\} \quad (7.1d)$$

$$\sum_{p \in P_d^s} (x_{dp}^0 - u_{dp}^s + v_{dp}^s) \geq h_d^s, \quad s \in S, d \in D \quad (7.1e)$$

$$\sum_{p \in P_d^s} x_{dp}^0 + M(T_d^s - 1) < h_d^s, \quad s \in S, d \in D \quad (7.1f)$$

$$\sum_{p \in P_d^s} x_{dp}^0 + h_d^s T_d^s \geq h_d^s, \quad s \in S, d \in D \quad (7.1g)$$

$$x_{dp}^0 - a^{-1} u_{dp}^s - b^{-1} v_{dp}^s \geq 0, \quad s \in S, d \in D, p \in P_d^s \quad (7.1h)$$

$$u_{dp}^s + M(T_d^s - 1) \leq 0, \quad s \in S, d \in D, p \in P_d^s \quad (7.1i)$$

$$v_{dp}^s - M T_d^s \leq 0, \quad s \in S, d \in D, p \in P_d^s \quad (7.1j)$$

$$x, y, u, v \text{ continuous, nonnegative, } T \text{ binary} \quad (7.1k)$$

In the formulation, objective (7.1a) minimizes the total cost of links, i.e., the dimensioning cost. Constraints (7.1b) do not allow that nominal link loads exceed nominal link capacities. Constraints (7.1c) assure that for each demand its paths have jointly sufficient capacity to satisfy the demand volume requested for the nominal state. Next, constraints (7.1d) make sure that in each state $s \in S$ the capacity of each working link $e \in E \setminus \{s\}$ is not exceeded. Constraints (7.1e) assure that in each failure state $s \in S$, the adjusted flows are sufficient to realize the (possibly reduced with respect to h_d^0) volume h_d^s of demand $d \in D$. Then, constraints (7.1h)

do not allow releasing by more than the fraction a of the nominal flow assigned to a path, or increasing by more than the fraction b of the nominal flow. The values of a and b are fixed parameters with $0 \leq a \leq 1$ and $b \geq 0$. Constraints (7.1f) assure that the binary variable T_d^s takes value $T_d^s = 1$ if the demand is affected (by more than a given $c > 0$) in the failure state s and constraints (7.1g) deal with unaffected demands. Finally, constraints (7.1h)-(7.1j) forces variables x , u and v to be correctly defined with respect to affected/unaffected demands and T binary variables.

We discuss next state-dependent constraints (7.1f),(7.1g),(7.1i) and (7.1j). If $0 \leq \sum_{p \in P_d^s} x_{dp}^0 < h_d^s$, demand d is affected by the failure state s and its path-flows can only be thickened $u_{dp}^s = 0$. Otherwise, demand d is not affected and its path-flows can only be thinned $v_{dp}^s = 0$. For this purpose, the constraints use a very large number called big- M coefficient, as well as a very small value c which, together with T_d^s binary variables, emulate *IF*-like constraints. Using a large value M could lead to numerical instability. To avoid this issue, constraints (7.1f),(7.1g),(7.1i) and (7.1j) are modeled in the solver CPLEX as indicator constraints. For instance, the “strictly less than” linear constraints (7.1f) is modeled as *IF* $(\sum_{p \in P_d^s} x_{dp}^0 \geq h_d^s)$ *THEN* $T_d^s = 0$ *ELSE* $T_d^s = 1$.

7.3 Complexity discussion

Formulation (7.1) is hard to solve exactly by mixed-integer linear solvers for mainly two reasons. First, the number of variables used in the formulation can be very large, since the sets of admissible paths can be of exponential sizes with respect to the size of a network. Second, it contains binary variables and big- M coefficients, which leads to numerical instability and very poor linear relaxations. All this gives a hint on the practical difficulty and the computational complexity of the EFR problem. Before stating on the complexity let us clarify, in the light of the above formulation, the connection of EFR to other restoration strategies mentioned in Section 4.1. First, taking $a = 0$ and $b = 0$ leads to PD strategy (neither thinning nor thickening are possible). Second, for $a = 1$ and $b \rightarrow \infty$ one can see that EFR converges to Global Rerouting while for $a = 0$ and $b \rightarrow \infty$ EFR becomes RR strategy. We can easily check that all requirements for each strategy are satisfied. For instance, taking $a = 0$ implies that flow thinning is not allowed, while $b \rightarrow \infty$ means that virtually new rerouting paths can be created; this is because the flow on some path can be enlarged at any finite value starting from practically insignificant flow values. All these special cases have different levels of complexity for the single link failure case: GR and PD fall into the polynomial time complexity class (Orlowski and Pióro 2012), while RR is shown to be *NP*-hard for both the directed case (Nace et al. 2013) and the undirected one (Tomaszewski 2013). The observation suggests that both problems will exhibit the same *NP* complexity.

Not surprisingly, we will use similar arguments to show the *NP*-hardness in question. The proof given for EFR is inspired by the *RR NP*-hardness proof presented in (Nace et al. 2013). The proof is based on a specific network constructed to show that finding an RR solution is equiv-

alent to solving the elementary path problem EL-PATH, which consists in answering the question whether there exists an elementary path going through a fixed link in a directed graph. Problem EL-PATH is NP -hard, because it can be reduced from 2-DIV-PATH, which itself is NP -hard (Fortune et al. 1980). The network instance utilized in the proof is depicted in Figure 7.4. It assumes a simplified failure scenario involving a single demand and failures of only two links. Considering a single demand only has a particular impact on using EFR, the method cannot employ thinning as there is no interest in considering cases when the demand is unaffected. Hence, in the following, we consider the EFR problem with a set to 0 and b set to a large but finite value, thus the nominal flows cannot be decreased, but they can be increased and multiplied by $b + 1$. Then, bearing in mind that we are looking for solutions involving only elementary routing paths, the following proposition can be proven.

Proposition 7.1 *EFR problem, represented by (7.1), is NP -hard already for the case with two failing links and one demand.*

Proof. The proof is deduced from the problem of existence of an elementary path in a directed graph traversing a given link. Let this problem be denoted by EL-PATH. As discussed in (Nace et al. 2013), EL-PATH is NP -hard.

The upper left part of Figure 7.4 represents an instance of EL-PATH, asking for finding whether there exists an elementary path from v to w traversing a given link (a, b) in a directed graph represented by the oval. The main part of the figure shows how this instance of EL-PATH can be reduced to an instance of the considered case of EFR with one demand d from node o to node t (with volume $h_d = 1$) and two failing links f_1 and f_2 .

In the main graph, the links depicted by solid lines have large unit costs ξ , while unit costs of the links drawn as dotted lines are negligible with respect to ξ . The two ovals in the main graph are the copies of the oval in the upper left part. Unit costs of all possible links in these two ovals are also negligible as compared to ξ . Note that the nodes in both ovals in the main graph have the same names, but this should not lead to a confusion. We first show that the objective function value C of any feasible solution of EFR must be greater than or equal to $\frac{5}{3}\xi$. To demonstrate this, consider the following cuts involving the expensive links, i.e., links (o, v_1) , (o, v_2) , (o, v_3) , (w_{31}, u_{31}) , and (w_{32}, u_{32}) :

- one cut for the normal state: $\{(o, v_1), (o, v_2), (o, v_3)\}$
- first cut for failure of f_1 : $\{(w_{31}, u_{31}), (w_{32}, u_{32}), (o, v_2)\}$
- second cut for failure of f_1 : $\{(w_{31}, u_{31}), (o, v_2), (o, v_3)\}$
- first cut for failure of f_2 : $\{(o, v_1), (o, v_3), (w_{32}, u_{32})\}$
- second cut for failure of f_2 : $\{(w_{31}, u_{31}), (w_{32}, u_{32}), (o, v_1)\}$

The capacity of each of these five cuts must be greater than or equal to the requested volume $h_d = 1$. Hence, by summing up the inequalities

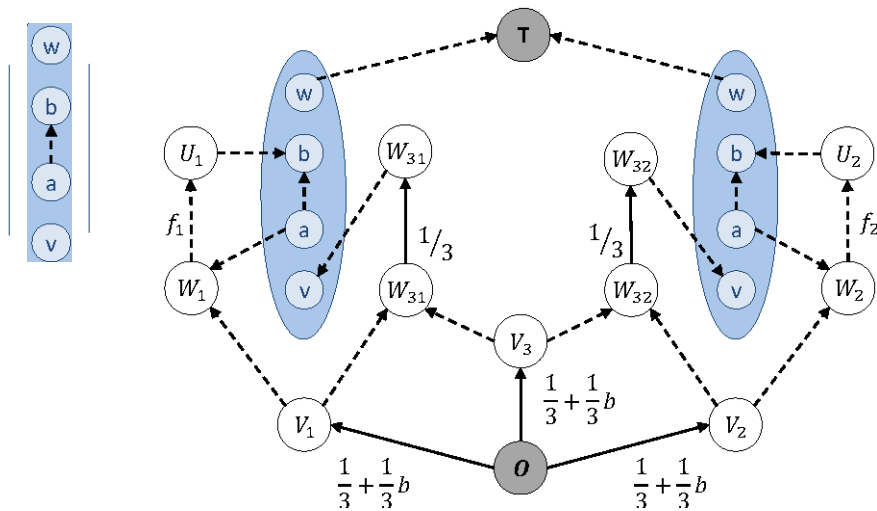


Figure 7-4- Network instance proving NP-hardness of EFR.

expressing these conditions for all five cuts we see that the sum of the capacity reservations of the five expensive links must be at least $\frac{5}{3}$. As

each of these links has the unit cost i , we conclude that $C \geq \frac{5}{3}$. Now we will establish bounds of optimal solutions of EFR in the case when an elementary path from v to w traversing link (a,b) in the oval exists (Case 1), and in the case when it does not exist (Case 2).

Case 1: Suppose that the oval contains an elementary path P from v to w traversing link (a,b) . Then, there exists a solution that achieves the cost $C^0 = \frac{5}{3}$. In this solution, the flows in the normal state are as follows:

- $\frac{1}{3}$: path $o - v_1 - w_1 - u_1 - (b - w \text{ part of } P) - t$,
- $\frac{1}{3}$: path $o - v_3 - w_{31} - u_{31} - v - (v - a \text{ part of } P) - w_1 - u_1 - b - (b - w \text{ part of } P) - t$,
- $\frac{1}{3}$: path $o - v_3 - w_{32} - u_{32} - v - (v - a \text{ part of } P) - w_2 - u_2 - b - (b - w \text{ part of } P) - t$,
- $\frac{1}{3}$: path $o - v_2 - w_2 - u_2 - (b - w \text{ part of } P) - t$,
- $\frac{1}{6}$: path $o - v_3 - w_{32} - u_{32} - v - P - t$,
- $\frac{1}{3}$: path $o - v_1 - w_{31} - u_{31} - v - P - t$,
- $\frac{1}{6}$: path $o - v_3 - w_{31} - u_{31} - v - P - t$,
- $\frac{1}{3}$: path $o - v_1 - w_{32} - u_{32} - v - P - t$.

When link f_1 fails, flow on $o - v_3 - w_{32} - u_{32} - v - P - t$ is enlarged to reach $\frac{1}{6}$ and flow on $o - v_1 - w_{31} - u_{31} - v - P - t$ is thickened to $\frac{1}{3}$. When link f_2 fails, the third and the fourth normal flows are rerouted in a symmetric way by enlarging respectively the flows on path $o - v_3 - w_{31} - u_{31} - v - P - t$ and path $o - v_1 - w_{32} - u_{32} - v - P - t$. The resulting capacity reservations of the expensive, solid-line links are all equal to $\frac{1}{3}$ (for links $w_{31} - v_{31}$ and $w_{32} - v_{32}$), and $(\frac{1}{3} + \frac{1}{3b})$ for the three other links (respectively $o - v_1 - 1$, $o - v_2 - 2$, and $o - v_3$) as depicted in Figure 7.4. The resulting capacity is $(\frac{5}{3} + \frac{1}{b})\xi$ and for $b = 36$ it gives a cost of $\frac{61}{36}\xi$.

Case 2: Now assume that there is no elementary path from v to w through (a, b) in the oval. We will show that in this case the cost C^0 of an optimal solution of EFR must be greater than or equal to $\frac{7}{4}\xi$. Indeed, it has been shown in (Nace et al. 2013) that any solution for RR in the considered network is necessarily greater than or equal to $\frac{7}{4}\xi$, which is greater than $\frac{61}{36}\xi$. This holds also for EFR as for $a = 0$ all EFR solutions are necessarily solutions of RR, so EFR cannot do better than RR. To summarize, solving the constructed instance of EFR for the optimal cost C^0 yields a solution of EL-PATH: an elementary path P in question exists if $C^0 \leq \frac{61}{36}\xi$, and does not exist when $C^0 \geq \frac{7}{4}\xi$. In this way, EL-PATH is reduced to EFR and hence EFR is NP -hard.

7.4 Heuristic method

As remarked in the previous section, formulation (7.1) combines the big- M constraints with a large number $O(|D||S|)$ of binary variables, and produces large-scale integer-programming problems which happen to be intractable even for moderate size networks.

Solving the EFR-P requires solving a combinatorial optimization problem over the combinations of affected/unaffected demands sets for each failure. Additionally, it requires applying path generation in the solution process. In consequence, the exact solution of EFR-P would require branch-and-price algorithms. Obviously, solving formulation (7.1) with binary variables T_a^s set to fixed values representing a certain choice of affected/unaffected demands would give an upper bound for the optimal solution of EFR-P. In fact, as we will see below, the optimization problem behind such upper bound solutions is much simpler to solve than EFR-P itself.

Hence, a natural heuristic approach to solve EFR-P would be to solve sequentially a set of problems with the unaffected/affected demands gradually fixed in each iteration. All this leads to a greedy heuristic approach described in Section 7.4.1. Next, the problem solved at each iteration, called EFR-PF (EFR-P fixed), is presented in detail in Section 7.4.2.

7.4.1 Description of the heuristic

In this section, we present the greedy heuristic method used to solve the EFR-P optimization problem. In each iteration, the heuristic creates and solves an instance of EFR-PF (version of EFR-P (7.1), in which all $T_d^s, d \in D, s \in S$, are fixed to values 0 or 1), and then adjusts the status (affected/unaffected) of the demands. The process is repeated until no improvement is achieved—more precisely, we stop when all demands' status remain unchanged.

The algorithm starts with an empty set of unaffected demands for each failure state, and with some initial admissible path-sets. Thus, at the beginning all demands are considered as potentially affected. After solving the corresponding LP, we know exactly which demands are affected and non-affected in the current solution x . This gives a hint how to update the unaffected demands for the next iteration—the set of unaffected demands is extended with the demands that are not affected in the obtained solution x . In this way the new unaffected demands can contribute to the restoration process by potentially releasing some flow on their paths. For the next iteration, the EFR-PF instance is extended with the constraints that characterize the demands just set to the unaffected status.

In consequence, any feasible solution will always keep these demands unaffected because of these constraints, which are not subject to removal in the subsequent steps of the iteration process. Therefore, at each iteration, the obtained solution x can only extend the set of unaffected demands and no demands that have been previously made unaffected can become affected. We proceed like this with updating the sets of unaffected demands, and stop when the current solution does not change the status of the demands assumed for the previous iteration. This is the key idea of the method used in Heuristic EFR-H detailed in Algorithm 4 below.

Algorithm 4: Heuristic method for EFR-P

- Step 0: Initialization of path-sets**
Define initial admissible path-sets $P_d \subseteq \bar{P}_d, d \in D$.
- Step 1: Initialization of demands status**
Put $D^s := \emptyset, s \in S$.
- Step 2: Solving EFR-PF**
Let x be the resulting solution. Set $D^s(x)$: the set of unaffected demands corresponding to solution x .
- Step 3: Update demands states**
If $D^s = D^s(x), s \in S$, then go to Step 4. Otherwise, $D^s := D^s(x), s \in S$, and go back to Step 2.
- Step 4: Return the solution**
-

We can easily deduce that the algorithm will always stop. According to the above, once a demand is considered unaffected it will remain unaffected for all consecutive iterations. On the other hand, affected demands can change their status and become unaffected. Hence, the process will necessarily stop since the set of unaffected demands cannot be extended to infinity, and the algorithm ends when we obtain the same set of unaffected demands for two consecutive iterations.

One of the disadvantages of the presented algorithm is its fast con-

vergence to local minima. If a large set of demands becomes unaffected in a single iteration, the heuristic may not be able to find high quality solutions. In order to alleviate this issue, the number of demands that can become unaffected in a failure state s in a single iteration is limited to MAX_d demands. If MAX_d takes a very low value, the heuristic will slowly converge, will explore a lot more scenarios, but the computation time will be much longer. In contrast, if MAX_d takes a high value, the computations will be fast, but the algorithm will quickly converge to a local minimum that can be very far from the true optimum. The most reasonable value of MAX_d depends on a network instance. However, we notice that $MAX_d = 5$ represents a good compromise between the computation time and the solution quality. Still, observe that the presented method is not exact; thus, obtained solutions are in general suboptimal.

7.4.2 Formulation of EFR-PF problem

Let us now look in detail how the EFR-PF problem used in each iteration of the above algorithm is formally defined. We first introduce additional notation by defining two sets: $E(d) = \{s \in S : d \in D^s\}$ is the set of links whose failure does not affect demand d , and $\bar{E}(d) = \{s \in S : d \in \bar{D}^s\}$ is the set of links that affect demand d . Note that in the EFR-PF problem the sets $D^s, s \in S$ are predefined. Then, a linear programming formulation for EFR-PF, denoted $P_F(D^s, P)$ and given in (7.2), can be obtained from (7.1) by fixing the variables accordingly.

$$EFR-PF(P): \min C = \sum_{e \in E} \xi_e y_e \quad (7.2a)$$

$$[\pi_e^0] \sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \leq y_e, \quad e \in E \quad (7.2b)$$

$$[\lambda_d^0] \sum_{p \in P_d} x_{dp}^0 \geq h_d^0, \quad d \in D \quad (7.2c)$$

$$[\pi_e^s] \sum_{d \in D} \sum_{p \in P_d^s} \delta_{edp} x_{dp}^0 - \sum_{d \in D^s} \sum_{p \in P_d^s} \delta_{edp} u_{dp}^s + \sum_{d \in \bar{D}^s} \sum_{p \in P_d^s} \delta_{edp} v_{dp}^s \leq y_e, \quad s \in S, e \in E \setminus \{s\} \quad (7.2d)$$

$$[\lambda_d^s] \sum_{p \in P_d^s} (x_{dp}^0 - u_{dp}^s) \geq h_d^s, \quad d \in D, s \in E(d) \quad (7.2e)$$

$$[\lambda_d^s] \sum_{p \in P_d^s} (x_{dp}^0 + v_{dp}^s) \geq h_d^s, \quad d \in D, s \in \bar{E}(d) \quad (7.2f)$$

$$[\sigma_{dp}^s] u_{dp}^s \leq a x_{dp}^0, \quad s \in E(d), p \in P_d^s \quad (7.2g)$$

$$[\sigma_{dp}^s] v_{dp}^s \leq b x_{dp}^0, \quad s \in \bar{E}(d), p \in P_d^s \quad (7.2h)$$

$$x, u, z, y \text{ continuous, nonnegative} \quad (7.2i)$$

In formulation (7.2), objective (7.2a) minimizes the total cost of links, and (7.2b)-(7.2f) are similar to (7.1b)-(7.1e). Constraint (7.1e) is split in two constraints (7.2e) and (7.2f). Then, constraints (7.2g) do not allow releasing more than a fraction a of the normal flow assigned to a path, and constraints (7.2h) ensure that flow increase on a disrupted path by more than the fraction b of the normal flow is not permitted. The values

of a and b are fixed parameters with $0 \leq a \leq 1, b \geq 0$. In the formulation, the quantities in brackets to the left are dual variables associated with constraints. All these variables are, by assumption, non-negative.

The condition stipulating that surviving flows of the affected demands are not sufficient to satisfy the required demand volume can be expressed as:

$$\sum_{p \in P_d^s} x_{dp}^0 \leq h_d^s - c, \quad s \in S, d \in \bar{D}^s, \quad (7.3)$$

for a very small $c > 0$. Nevertheless we have deliberately chosen to not introduce this constraint into the EFR-PF formulation (7.2)¹. In practice, the EPRF-PF formulation (7.2) solved in the first iteration of the algorithm does not contain constraints (7.2e). Having a solution, we check each demand $d \in D$ if it satisfies (7.3). If constraint (7.3) is not satisfied for a particular demand, the demand is added to set D^s . At this stage, the obtained solution x remains feasible for the new LP problem. Thus, the heuristic generates a sequence of improving approximate solutions.

Although EFR-PF is an LP, the problem is still not easy to solve because of the possibly exponential numbers of path-flow variables x, u , and v . To avoid the non-compact nature of the formulation, we are forced to generate path on the fly using Path Generation (PG) applied to formulation (7.2). Unfortunately, PG turns out to be a difficult problem as we show in the next subsections.

7.4.3 Dual problem

Let us consider the problem dual to problem (7.2) for a given set of admissible paths P . The dual, denoted by $EFR - DF(P)$, is as follows (for the derivation see for example Pióro and Medhi (2004b)):

$$EFR-DF(P): \max W = \sum_{d \in D} (h_d^0 \lambda_d^0 + \sum_{s \in S} h_d^s \lambda_d^s) \quad (7.4a)$$

$$\pi_e^0 + \sum_{s \in S \setminus \{e\}} \pi_e^s = \xi_e, \quad e \in E \quad (7.4b)$$

$$\begin{aligned} \lambda_d^0 + \sum_{s \in E(d) \setminus p} (\lambda_d^s + a \sigma_{dp}^s) + \sum_{s \in \bar{E}(d) \setminus p} (\lambda_d^s + b \sigma_{dp}^s) \\ \leq \sum_{s \in S \setminus p} |\rho|^s + |\rho|^0, \quad d \in D, p \in P_d \end{aligned} \quad (7.4c)$$

$$\lambda_d^s + \sigma_{dp}^s \geq |\rho|^s, \quad d \in D, p \in P_d, s \in E(d) \setminus p \quad (7.4d)$$

$$\lambda_d^s - \sigma_{dp}^s \leq |\rho|^s, \quad d \in D, p \in P_d, s \in \bar{E}(d) \setminus p \quad (7.4e)$$

$$\lambda, \pi, \sigma \text{ continuous, nonnegative} \quad (7.4f)$$

where, for a given path $p \in P$, we have used the notation $|\rho|^0 = \sum_{e \in p} \pi_e^0$ and $|\rho|^s = \sum_{e \in p} \pi_e^s$.

We notice that the optimal solution of (7.4) must give for $d \in D, p \in P_d, s \in E(d) \setminus p$:

$$\sigma_{dp}^s = \begin{cases} 0, & \text{if } |\rho|^s \leq \lambda_d^s \\ |\rho|^s - \lambda_d^s, & \text{if } |\rho|^s > \lambda_d^s \end{cases} \quad (7.5)$$

¹The idea behind this relies on the way the above heuristic works: we do not need to constrain the solution obtained at each step of the heuristic to the fixed set of affected/unaffected demands but only keeping the set of unaffected ones as growing.

and for $d \in D, p \in P_d, s \in \bar{E}(d) \not\equiv p$:

$$\sigma_{dp}^s = \begin{cases} |p|^s - \lambda_d^s, & \text{if } |p|^s < \lambda_d^s \\ 0, & \text{if } |p|^s \geq \lambda_d^s \end{cases} \quad (7.6)$$

Observe that if we replace σ_{dp}^s by these values, formulation (7.4) and formulation (7.9) given below have the same sets of feasible variables π and λ . We introduce some additional notation. For $d \in D, p \in P_d$:

$$F_d(p) = \{s \in E(d) \not\equiv p : |p|^s > \lambda_d^s\} \quad (7.7)$$

Symmetrically, for $d \in D, p \in P_d$:

$$\bar{F}_d(p) = \{s \in \bar{E}(d) \not\equiv p : |p|^s < \lambda_d^s\} \quad (7.8)$$

Considering (7.5)-(7.8), the transformed formulation of the dual problem is as follows:

$$\max W = \sum_{d \in D} (h_d^0 \lambda_d^0 + \sum_{s \in S} h_d^s \lambda_d^s) \quad (7.9a)$$

$$\pi_e^0 + \sum_{s \in S \setminus \{e\}} \pi_e^s = \xi_e, \quad e \in E \quad (7.9b)$$

$$\lambda_d^0 + a \sum_{s \in F_d(p)} \Delta_{dp}^s + b \sum_{s \in \bar{F}_d(p)} \Delta_{dp}^s - \sum_{s \in S \setminus p} \Delta_{dp}^s \leq |p|^0, \quad d \in D, p \in P_d \quad (7.9c)$$

$$\lambda, \pi \text{ continuous, nonnegative} \quad (7.9d)$$

with $\Delta_{dp}^s = |p|^s - \lambda_d^s, d \in D, p \in P_d, s \in S \setminus p$.

7.4.4 Pricing problem

Let \bar{r}_d denote the set of all paths in the network graph between $o(d)$ and $t(d)$. The essence of the pricing problem related to the dual problem (7.9) formulated for a given (limited) admissible path-sets $P_d, d \in D$, is to check whether for at least one demand $d \in D$ there exists a path $q \in \bar{r}_d \not\equiv P_d$ (i.e., path q outside the current admissible path-set P_d), for which (7.9c) cannot be satisfied for any set of non-negative dual variables $\sigma_{dq}^s, s \in \bar{E}(d) \not\equiv q$. In fact, this is the case if and only if:

$$\Delta_{dq}^0 + \sum_{s \in S \setminus q} \Delta_{dq}^s - a \sum_{s \in F_d(q)} \Delta_{dq}^s - b \sum_{s \in \bar{r}_d(q)} \Delta_{dq}^s < 0 \quad (7.10)$$

with $\Delta_{dq}^0 = |q|^0 - \lambda_d^0, d \in D$.

The Pricing Problem (PP) for a given π, λ and a given demand $d \in D$ is as follows:

$$\text{minimize } /q/ \text{ over } q \in \hat{r}_d \quad (7.11)$$

where

$$/q/ = \Delta_{dq}^0 + \sum_q \Delta_{dq}^s - a \sum_{s \in F_d(q)} \Delta_{dq}^s - b \sum_{s \in \bar{r}_d(q)} \Delta_{dq}^s.$$

Observe that if $/q/ < 0$ for a path q satisfying (7.11), then the current optimal dual solution π, λ violates the dual constraints (7.9c) for path q . Moreover, the violation is maximal over all paths $p \in \hat{r}_d$.

Finally, we note that PP is difficult. In particular, the term $\sum_{s \in S \neq q} \sum_{e \in p} \pi_e^s$ included in $\sum_{s \in S \neq q} \Delta_{dq}^s$ is difficult to deal with. This problem is also known in the literature as the quadratic shortest path problem, since its cost depends not only on the arcs included in the path, but also on the arcs outside. Furthermore it is encountered in the pricing problem of RR known to be NP -hard, see (Maurras and Vanier 2004, Mereu et al. 2009, Orłowski 2003, Nace et al. 2013) and a survey on path generation (Orłowski and Pióro 2012). In our experiments we have incorporated all potential paths in the initial path set.

7.4.5 Binary formulation of the pricing problem

The pricing problem formulated in (7.11) is difficult because of the particular form of the total dual length $\|q\|$. Nevertheless, the problem can be stated as a binary programming problem by means of formulation (7.12) given below. In the formulation, binary variables $u_e, e \in E$, specify the path q we are looking for: $q = \{e \in E : u_e = 1\}$. Binary variables $z^s, s \in S$, in turn, denote if $|p|^s > \lambda^s$. If that is the case, then $z^s = 1$; otherwise $z^s = 0$. Besides, $\delta^+(v)$ and $\delta^-(v)$ denote the sets of all links outgoing from, and all links incoming to, respectively, node $v \in V$, constant d indicates a considered demand, node o is its originating node and t is its terminating node.

$$\begin{aligned} \min L = & \sum_{e \in E} u_e \pi_e^0 + \sum_{s \in S} (1 - u_s) \left(\sum_{e \in E} u_e \pi_e^s - \lambda_d^s \right) \\ & - a \sum_{s \in S} z^s (1 - u_s) (1 - T_d^s) \left(\sum_{e \in E} u_e \pi_e^s - \lambda_d^s \right) \\ & - b \sum_{s \in S} (1 - z^s) (1 - u_s) T_d^s \left(\sum_{e \in E} u_e \pi_e^s - \lambda_d^s \right) \end{aligned} \quad (7.12a)$$

$$\sum_{e \in \delta^+(o)} u_e - \sum_{e \in \delta^-(o)} u_e = 1 \quad (7.12b)$$

$$\sum_{e \in \delta^+(v)} u_e - \sum_{e \in \delta^-(v)} u_e = 0, \quad v \in V \setminus \{o, t\} \quad (7.12c)$$

$$\lambda^s (1 - z^s) \geq \lambda^s - \sum_{e \in E} \pi_e^s u_e, \quad s \in S \quad (7.12d)$$

$$z^s \sum_{e \in E} \pi_e^s \geq \sum_{e \in E} \pi_e^s u_e - \lambda^s, \quad s \in S \quad (7.12e)$$

$$u_e \in \{0, 1\}, e \in E; z^s \in \{0, 1\}, s \in S. \quad (7.12f)$$

Constraints (7.12b) and (7.12c) assure that variables u_e that are equal to 1 form a path from o to t . Constraints (7.12d) and (7.12e) force each variable $z^s, s \in S$, to be equal to 1 when the length, with respect to π^s , of the path q defined by variables u is greater than λ^s ; and to be equal 0 otherwise. T_d^s are already given from the instance of the problem in hand.

To get rid of bi- and tri-linearities involving variables z^s and u_e in the objective function, we can introduce five binary variables $W^s, V^s, X_e^s, Y_e^s, Z_e^s, e \in E, s \in S$. The objective function should be rewritten

as

$$L = \sum_{e \in E} u_e \pi_e^0 + \sum_{s \in S} \left(\sum_{e \in E} X_e^s \pi_e^s - (1 - u_s) \lambda_d^s \right) - a \sum_{s \in S} (1 - T_d^s) \left(\sum_{e \in E} Y_e^s \pi_e^s - W_e^s \lambda_d^s \right) - b \sum_{s \in S} T_d^s \left(\sum_{e \in E} Z_e^s \pi_e^s - V_e^s \lambda_d^s \right) \quad (7.13)$$

and add the following set of constraints

$$X_e^s \geq u_e - u_s, \quad e \in E, s \in S \quad (7.14a)$$

$$Y_e^s \leq z^s, \quad Y_e^s \leq (1 - u_s), \quad Y_e^s \leq u_e, \quad e \in E, s \in S \quad (7.14b)$$

$$Z_e^s \leq (1 - z^s), \quad Z_e^s \leq (1 - u_s), \quad Z_e^s \leq u_e, \quad e \in E, s \in S \quad (7.14c)$$

$$W^s \geq z^s - u_s, \quad s \in S \quad (7.14d)$$

$$V^s \geq 1 - u_s - z^s, \quad s \in S. \quad (7.14e)$$

EFR — Practical applications to Multiple Partial Link Failures

Contents

8.1 Application to Fixed Wireless Networks	113
8.1.1 Introduction to EFR_{PLF} and Notations	114
8.1.2 Mathematical formulation of EFR_{PLF-P}	115
8.1.3 Modulation Schemes management	116
8.2 Towards a distributed version of EFR	118

Chapter 7 focused on total single link failures. In this chapter, our intention is to extend the study to multiple partial link failures that can arise in broadband wireless communications networks, presented in Chapter 1, (Willebrand and Ghuman 2002, Son and Mao 2010). The extended version of EFR, denoted EFR_{PLF} is based on both *Flow-Thinning* and *Elastic Flow Rerouting* strategies.

8.1 Application to Fixed Wireless Networks

In this section we present an extension of EFR, referred to as EFR_{PLF} , intended to handle partial link failures. It is among the first attempts to address flow protection against multiple partial failures in an implementable way. The approach is based on the two (re)routing strategies presented in this document: *Flow-Thinning* and *Elastic Flow Rerouting*. In short, EFR_{PLF} can be seen as an extension of EFR to partial failures using the features of FT. The approach works as follows. When a link partially fails, the affected nominal path-flows can only be thinned following the idea of FT. Next, flow adjustments are made according to EFR. In this section, we present the optimization problem (referred to as EFR_{PLF-P}) associated to the EFR_{PLF} approach discussed above.

8.1.1 Introduction to EFR_{PLF} and Notations

We keep the notations presented in 7.2.1. However, we need to introduce a few more notations.

Network links are subject to capacity variations that we consider as (partial) failures. The set of failure states is still denoted by S . Each failure state $s \in S$ is identified with a set of links ($s \subseteq E$) that cannot operate at full capacity and a set of bandwidth availability coefficient $\alpha_e^s, s \in S, e \in E$ with value $\alpha_e^s < 1$ if s contains e , and $\alpha_e^s = 1$ otherwise. First, the nominal path-flow x_{dp}^0 affected by s ($p \in P_d^s$) can only be thinned (remind that in EFR, affected nominal path-flows are lost). They are thinned to z_{dp}^s , which are optimization variables. A traffic demand d is said to be affected by a failure state s if its surviving nominal path-flows (thinned or not) are not sufficient to carry the traffic volume assumed for this state, i.e., when $\sum_{p \in P_d^s} z_{dp}^s + \sum_{p \in P_d^s} x_{dp}^0 < h_d^s$. Otherwise, the demand is unaffected. The rest of variables are such as in EFR-P formulation and the process follows the same idea as for EFR.

Example 8.1 Consider the same example than Example 7.1 dedicated to EFR. We show below what is the result of the minimum link cost network when EFR is used to recover from single partial link failures, where a failing link suffers a degradation of 50% of its capacity due to a change in the modulation scheme. As the aim is to describe the behavior of EFR_{PLF}, the presented solutions is only near-optimal. The obtained solution gives a capacity of 1 to link (A, E), $\frac{1}{2}$ to link (B, E), and $\frac{3}{4}$ to each other link, which gives a global cost equal to $\frac{18}{4} < 5$. Several nominal routing gives the same dimensioning cost. In the nominal state, as depicted in Figure 8.1, Demand d_1 is realized on 2 paths: $p_1 = \{A - E\}$, $p_2 = \{A - B - D - E\}$ with respectively 1 and $\frac{1}{4}$ flow value; and demand d_2 is routed on 2 paths: $p_3 = \{A - B - D\}$ with $\frac{1}{2}$ flow value, $p_4 = \{A - C - D\}$, with $\frac{3}{4}$ flow value.

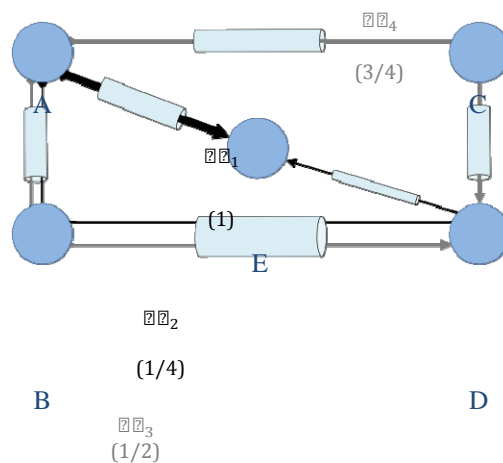


Figure 8.1 – EFR - Nominal State.

Figure 8.2 represents the failure of link (A, E). In this scenario, capacity of link (A, C) is decreased down to $\frac{1}{2}$. Hence, bandwidth of path p_1 has to meet the capacity limit of link (A, C), i.e. $\frac{1}{2}$. Now that paths routed through a failed link has been thinned (or not), we can determine the status of each demand. Clearly, due to the failure, the traffic requirement for d_1 is no longer met and d_1 is considered affected, while d_2 remains unaffected. There is no bandwidth remaining on

links (A, B) and (B, D) to increase the traffic of d_1 on path p_2 . Hence, as demand

d_2 is not affected by the failure, it releases $\frac{1}{4}$ of bandwidth on path p_3 (enough bandwidth is remaining of path p_4). $\frac{1}{4}$ unit of bandwidth is now available on links (A, B) and (B, D) and the bandwidth of p_2 is increased to $\frac{1}{2}$ to meet traffic requirement of d_1 . Then, both demands are fully operational.

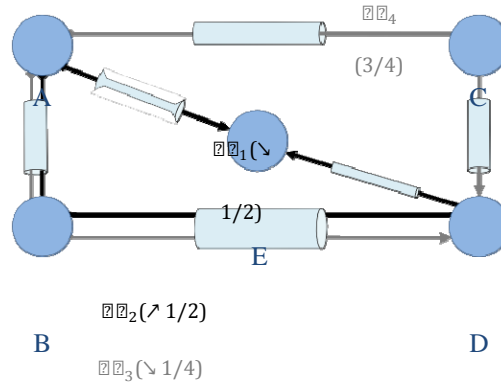


Figure 8.2 – EFR – Partial Failure of link (A, E).

The other failure scenarios can be managed in a similar manner.

8.1.2 Mathematical formulation of

EFR_{PLF-P}

The optimization problem considered in this section is referred to as Elastic Flow Rerouting Problem for Partial Link Failures (EFR_{PLF-P}) and is as follows. We want to minimize the total cost of link capacity assuming that in the nominal state of network operation, all demand volumes are realized by means of nominal path-flows. When the network is subject to a failure of a set of links then the demand volumes, possibly reduced, are realized for the duration of the failure state by appropriate thinning or thickening of the nominal path-flows depending on the state of their demand and the fact that they are perturbed.

The mathematical formulation of EFR_{PLF-P}, inspired from EFR, is given below. In 8.1, the objective function of EFR_{PLF-P} is to minimize the total cost of the network. Constraints (8.1b) do not allow the nominal link loads to exceed the nominal link capacities. Constraints (8.1c) ensure that the nominal flows are sufficient to realize the volume of each demand $d \in D$. In a failure state $s \in S$, the best modulation scheme cannot be used for the affected links as they no longer meet the SNR requirement, see (8.1d). When $y^w = 1$ for $w \in W_e$, the link capacity available in state s is assumed to be equal to a given value b_e^{ws} , $e \in E, s \in S$ with $b_e^{ws} \leq b_e^{w0}$ for all $s \in S$. In general, in a failure state more than one link can have its availability ratio decreased by the modulation scheme. Constraints (8.1e) assure that in each failure state $s \in S$, the adjusted flows are sufficient to realize the volume of each demand $d \in D$ and constraints (8.1d) check that the surviving capacity of each link $e \in E$ is not exceeded. Constraints (8.1f) - (8.1g) are related to the status of demands. They restrict T_d^s values to be equal to 1 for an affected demand and 0 otherwise. Note that (8.1f) is handled by CPLEX MIP solver in the same way as (7.1f) in Section 7.2. Constraints (8.1h) - (8.1i) express bound constraints for variables u, v , and

z.

$$\min C = \sum_{e \in E} \xi_e y_e \quad (8.1a)$$

$$\sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \leq y_e, \quad e \in E \quad (8.1b)$$

$$\sum_{p \in P_d} x_{dp}^0 \geq h_d^0, \quad d \in D \quad (8.1c)$$

$$\sum_{d \in D} \left(\sum_{p \in P_d^s} \delta_{edp} (x_{dp}^0 - u_{dp}^s + v_{dp}^s) + \sum_{p \in P_d^s} \delta_{edp} z_{dp}^s \right) \leq \alpha_e^s y_e, \quad s \in S, e \in E \quad (8.1d)$$

$$\sum_{p \in P_d^s} \delta_{edp} (x_{dp}^0 - u_{dp}^s + v_{dp}^s) + \sum_{p \in P_d^s} \delta_{edp} z_{dp}^s \geq h_d^s, \quad s \in S, d \in D \quad (8.1e)$$

$$\sum_{p \in \bar{P}_d^s} z_{dp}^s + \sum_{p \in P_d^s} x_{dp}^0 + M(T_d^s - 1) \leq h_d^s - c, \quad s \in S, d \in D \quad (8.1f)$$

$$\sum_{p \in \bar{P}_d^s} z_{dp}^s + \sum_{p \in P_d^s} x_{dp}^0 + h_d^s T_d^s \geq h_d^s, \quad s \in S, d \in D \quad (8.1g)$$

$$0 \leq u_{dp}^s \leq a x_{dp}^0, \quad s \in S, d \in D, p \in P_d^s \quad (8.1h)$$

$$0 \leq v_{dp}^s \leq b x_{dp}^0, \quad s \in S, d \in D, p \in P_d^s \quad (8.1i)$$

$$0 \leq z_{dp}^s \leq x_{dp}^0, \quad d \in D, s \in S, p \in \bar{P}_d^s \quad (8.1j)$$

$$u_{dp}^s + M(T_d^s - 1) \leq 0, \quad s \in S, d \in D, p \in P_d^s \quad (8.1k)$$

$$v_{dp}^s - M T_d^s \leq 0, \quad s \in S, d \in D, p \in P_d^s \quad (8.1l)$$

$$x, y, z, u, v \text{ continuous, nonnegative, } T \text{ binary} \quad (8.1m)$$

Note that $\text{EFR}_{\text{PLF-P}}$ is obviously as difficult as EFR as for total failures it reduces to EFR. This statement holds since Flow Thinning can be applied only for partial failures—it completely disappears when the link failure cuts the traffic entirely (total failure). Therefore the already demonstrated NP -hardness of EFR immediately implies the following corollary.

Corollary 8.1 *The EFR_{PLF} problem represented by (8.1) is NP -hard.*

In practice, $\text{EFR}_{\text{PLF-P}}$ is intrinsically harder than EFR as it includes two consecutive optimization procedures, namely Flow Thinning of affected flows and Elastic Flow Rerouting for the rest of flows. In the line of the heuristic proposed above for EFR, a similar process can be built for $\text{EFR}_{\text{PLF-P}}$.

The source of difficulty is three-fold: first, the combinatorial aspect related to the distinction between the affected and the unaffected demands; second, potentially exponential number of multiple partial link failure states to be considered; third, the difficulty of the associated pricing problem.

8.1.3 Modulation Schemes management

In this section, we adjust the $\text{EFR}_{\text{PLF-P}}$ problem presented in 8.1 to manage channels modulation.

Each link $e \in E$ is assigned a set of usable frequencies W_e (7MHz, 17MHz, 28MHz . . . see Table 1.1) and a non-negative cost associated which

now depends of the frequency, ξ_e^w , $w \in W_e$. The capacity of link e , denoted by y_e , is determined by the combination of the modulation and the frequency ($y_e = \sum_{w \in W_e} b_e^{w0} y_e^w$), where b_e^{w0} gives the bandwidth efficiency coefficient used in optimal conditions while in degraded conditions (partial failure s) this is given by $b_e^{ws} < b_e^{w0}$ which leads to a decreased capacity of the link. The objective of EFR_{PLF-P} is to minimize the total cost of the network. Hence, 8.1a is modified as follows:

$$C = \sum_{e \in E} \sum_{w \in W_e} \xi_e^w y_e^w. \quad (8.2)$$

Clearly, a link can only use one frequency:

$$\sum_{w \in W_e} y_e^w = 1, e \in E. \quad (8.3)$$

In the nominal state, constraint 8.1b do not allow the nominal link loads to exceed the nominal link capacities. The nominal capacity of link e is now equal to b_e^{w0} , where $w \in W_e$ and $y_e^w = 1$

$$\sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \leq \sum_{w \in W_e} b_e^{w0} y_e^w, e \in E \quad (8.4)$$

In a failure state $s \in S$, the best modulation scheme cannot be used for the affected links as they no longer meet the SNR requirement. When $y_e^w = 1$ for $w \in W_e$, the link capacity available in state s is assumed to be equal to a given value b_e^{ws} , $e \in E$, $s \in S$ with $b_e^{ws} \leq b_e^{w0}$ for all $s \in S$. In general, in a failure state more than one link can have its availability ratio decreased by the modulation scheme. The following constraints assure that in each failure state $s \in S$, the surviving capacity of each link $e \in E$ is not exceeded.

$$\begin{aligned} & \sum_{d \in D} \sum_{p \in P_d^s} \delta_{edp} (x_{dp}^0 - u_{dp}^s + v_{dp}^s) \\ & + \sum_{d \in D} \sum_{p \in P_d^s} \delta_{edp} z_{dp}^s \leq \sum_{w \in W_e} b_e^{ws} y_e^w, s \in S, e \in E \end{aligned} \quad (8.5a)$$

Hence, for given sets of admissible paths P_d , $d \in D$, problem EFR_{PLF-P} can be represented by the path-flow mixed-integer programming (MIP) formulation involving nonnegative continuous variables x_{dp}^0 ($d \in D$, $p \in P_d$), x_{dp}^s ($s \in S$, $d \in D$, $p \in P_d$), z_{dp}^s ($s \in S$, $d \in D$, $p \in \bar{P}_d^s$), u_{dp}^s ($s \in S$, $d \in D$, $p \in P_d^s$), v_{dp}^s ($s \in S$, $d \in D$, $p \in P_d^s$), and binary y_e^w ($e \in E$, $w \in W_e$) and

$T_d^s (s \in S, d \in D)$. This MIP formulation is as follows:

$$\min C = \sum_{e \in E} \sum_{w \in W_e} \xi_e^w y_e^w \quad (8.6a)$$

$$\sum_{d \in D} \sum_{p \in P_d} \delta_{edp} x_{dp}^0 \sum_{w \in W_e} y_e^w, \quad e \in E \quad (8.6b)$$

$$\sum_{p \in P_d} x_{dp}^0 \geq h_d^0, \quad d \in D \quad (8.6c)$$

$$\begin{aligned} & \sum_{d \in D} \sum_{p \in P_d^s} \delta_{edp} (x_{dp}^0 - u_{dp}^s + v_{dp}^s) + \sum_{d \in D} \sum_{p \in P_d^s} \delta_{edp} z_{dp}^s \\ & \leq \sum_{w \in W_e} b_e^{ws} y_e^w, \quad s \in S, e \in E \end{aligned} \quad (8.6d)$$

$$\sum_{p \in P_d^s} \delta_{edp} (x_{dp}^0 - u_{dp}^s + v_{dp}^s) + \sum_{p \in P_d^s} \delta_{edp} z_{dp}^s \geq h_d^s, \quad s \in S, d \in D \quad (8.6e)$$

$$\sum_{p \in P_d^s} z_{dp}^s + \sum_{p \in P_d^s} x_{dp}^0 + M(T_d^s - 1) \leq h_d^s - c, \quad s \in S, d \in D \quad (8.6f)$$

$$\sum_{p \in P_d^s} z_{dp}^s + \sum_{p \in P_d^s} x_{dp}^0 + h_d^s T_d^s \geq h_d^s, \quad s \in S, d \in D \quad (8.6g)$$

$$u_{dp}^s \leq a x_{dp}^0, \quad s \in S, d \in D, p \in P_d^s \quad (8.6h)$$

$$v_{dp}^s \leq b x_{dp}^0, \quad s \in S, d \in D, p \in P_d^s \quad (8.6i)$$

$$z_{dp}^s \leq x_{dp}^0, \quad d \in D, s \in S, p \in \bar{P}_d^s \quad (8.6j)$$

$$u_{dp}^s + M(T_d^s - 1) \leq 0, \quad s \in S, d \in D, p \in P_d^s \quad (8.6k)$$

$$v_{dp}^s - M T_d^s \leq 0, \quad s \in S, d \in D, p \in P_d^s \quad (8.6l)$$

$$\sum_{w \in W_e} y_e^w = 1, \quad e \in E \quad (8.6m)$$

$$x, y, z, u, v \text{ continuous, nonnegative, } T \text{ binary} \quad (8.6n)$$

The main drawback of EFR remains the fact that it is a centralized strategy. In case of failure, it consider that all nodes are aware of the failures and that all paths can be modified. The following section present a distributed version of EFR.

8.2 Towards a distributed version of EFR

The EFR strategies is not directly usable in a network restoration process. The main difficulty is storing a large amount of information in each router node. Indeed, EFR can require significant management efforts, since extensive state information (for instance, release/increase of bandwidth according to the failure state) is needed at each routing node to maintain all the nominal routing paths. This may, however, prevent the strategy from scaling in large networks. Applying this method in a restoration process involves releasing the bandwidth of some non-affected nominal paths of non-affected demands. This implies that the respective source nodes of these demands need to know about the failure and the amount of bandwidth to be released (or added) for each nominal path. All this means that a lot of information has to be stored at each node, while each source node has to be informed about the occurrence of failures. One way to handle

this problem is to restrict the amount of information stored at each node while omitting to communicate information about the state failure to all routing nodes. Consequently, we propose a modification of the method such that each node acts according to the local information, thus making this method more suitable for distributed environments.

The distributed versions of EFR is described below. The principle of DEFR (Distributed Elastic Flow Rerouting) consists in restricting the restoration to the extremity nodes of affected demands only. This is a natural solution, as these nodes will be aware of the failure and they will quickly be able to react to restore the lost traffic. These nodes may also be source and/or destination nodes for other traffic demands that have not necessarily been disrupted, and thus they can act directly to modify the bandwidth of some paths. The benefit of this method is twofold: first, the state information will concern only a subset of (failure) links; secondly, the nodes concerned with the traffic restoration are informed about the failure without any additional action as they are placed at the extremities of the affected path. In terms of cost-effectiveness, we would expect the DEFR strategy to perform slightly less well than EFR.

Let us illustrate how the DEFR strategy operates using the example 8.2, given below.

Example 8.2 Consider the same example than Example 8.1 dedicated to EFR_{PLF} . However, we change the source node of demand d_1 to node B. Demand d_2 now routes the same volume of traffic (1 unit) from node B to node E. We keep on considering single partial link failures where a failing link suffers a degradation of 50% of its capacity due to a change in the modulation scheme. We show below what is the result of the minimum link cost network when EFR Considering EFR, the solution gives a capacity of 1 to link (B, E), $\frac{1}{2}$ to link (A, E), and $\frac{3}{4}$ to each other links, which gives a global cost equal to $\frac{18}{4}$.

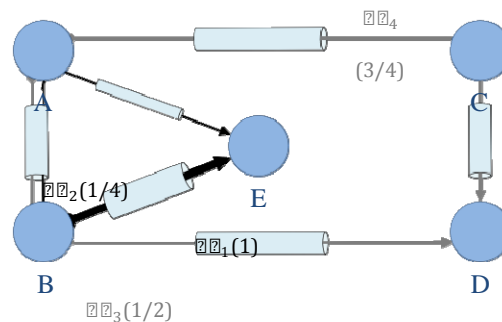


Figure 8.3 – EFR - Nominal State.

Figure 8.4 represents the failure of link (B, E). Clearly, this example is very similar the management of the failure of link (A, E) in Example 8.1.

However, when considering DEFR, the two examples are very different. When link (B, E) fails, node B and E are aware of the failure as they are extremity nodes of demand d_1 , which has disturbed paths. In example 8.1, demands d_1 and d_2 shares the same source node. Hence, paths of d_2 can be thinned if required. The solution cost remains the same, with a global cost of $\frac{18}{4}$.

In this new example, demand d_2 can be modified only by node A and D, which are not aware of the failure. Solution of EFR depicted in Figure 8.4 is not feasible

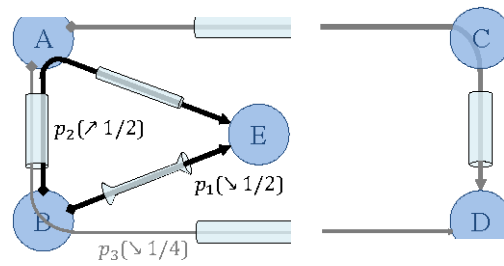


Figure 8-4- EFR- Partial Failure of link (B, E).

as traffic requirement for d_1 is no longer met and capacities used by d_2 on link (A,B) cannot be released. DEFR must overdimension link (A,B) with a capacity of $\mathbf{1}$ instead of $\mathbf{0.5}$. The global cast has increase to $\mathbf{1.5}$.

Thus, DEFR is on general slightly more expensive than EFR but in some case, it does not require any additional cast, especially when the set of demands is large.

Notice that implementation issues related to DEFR are present in appendix A.2.

EFR - Efficiency of the strategy

Contents

9.1 Dimensioning cost	121
9.2 Heuristic performance	123
9.3 Efficiency of EFR applied to partial failures	123
9.4 Restoration time	125
9.5 Conclusion on the efficiency of EFR	126

In this chapter, we present results of a computational study illustrating the performance of EFR-P and the heuristic on the total single link failure case followed by some preliminary results for the EFR_{PLF} problem.

9.1 Dimensioning cost

We consider undirected networks; thus, the optimization problems are treated in their versions for undirected network graphs. Network instances used in our tests, listed in Table 9.1, are taken from SNDlib (Orlowski et al. 2010). In the table depicted are the number of vertices, edges and demands of each networks, as well as the maximum and minimum degree of the corresponding graphs ($\Delta(G)$, $\delta(G)$). When solving the problems, we consider sets of all paths for *polska*, *nobel-us*, and *nobel-germany* (as these are low meshed network instances), and all paths with up to 7 links for *di-yuan* (a highly meshed network with 11 nodes). The study was carried out on a computer equipped with an Intel Xeon CPU E5-2670 2.60GHz processor and 132 GB of RAM, using CPLEX 12.5 (CPLEX 2013).

We recall that although in theory EFR-P gives the exact formulation and solving it should provide the optimal solution of the problem, using big- M coefficients in the formulation yields numerical issues. To handle this, constraints with the big- M coefficients are replaced by appropriate CPLEX indicator constraints.

Below we compare the cost effectiveness of the following traffic protection strategies: Elastic Flow Rerouting (EFR), Path Diversity (PD), Restricted Restoration (RR), i.e., end-to-end rerouting with stub release, and Global Rerouting (GR).

<i>network</i>	$ V $	$ E $	$ D $	$\Delta(G)$	$\delta(G)$
<i>di-yuan</i>	11	42	22	9	7
<i>polska</i>	12	18	66	5	2
<i>nobel-us</i>	14	21	91	6	2
<i>nobel-germany</i>	17	26	121	4	2

Table 9.1 – Network description.

In the reported experiments, we always use the same traffic reduction ratio β for all demands in all failure states, i.e., $h_d^s = \beta h_d^0$ for all $d \in D$ and $s \in S$. We also assume that the decreasing factor a is always equal to 1.0, while the increasing factor b takes values 0.1, 0.5, 1.0, or 3.0, which means that the nominal flows can be increased by up to 10% (resp. to 50%, multiplied by 2, or multiplied by 4). The decreasing and increasing factors are maximum limits applied to each path-flows in case of failures. In our experiments, we have incorporated all potential paths in the initial path set.

Let C_{GR} denote the optimal value of the objective function (i.e., minimum cost of the link capacity) for GR, and C_{EFR}^b , C_{PD} , and C_{RR} the respective values for the remaining strategies. Note that C_{EFR}^a is computed by solving EFR-P for the assumed thickening factor b . The rest of the costs are computed through solving appropriate optimization problems that can be found for example in (Pióro and Medhi 2004b). Certainly, the computed link capacities (and thus the link capacity cost) ensure routing of all traffic demands in the nominal state and the guaranteed traffic restoration (specified by β) in all failure states. As we already mentioned, the cost C_{GR} , indicated by GR, is not greater than any of the remaining costs, as GR assumes the least restricted flow restoration mechanism. Hence, we can define the relative cost increase for each strategy S (where S stands for EFR, PD, RR, or GR) with respect to GR as $G_S = \frac{C_S - C_{GR}}{C_{GR}} \times 100\%$. In the following, this quantity will be called the relative gap. Note that, by definition, $G_{GR} = 0$.

In Table 9.2, for all four network instances, we present G_{RR} , G_{PD} , and G_{EFR}^a , as a function of the traffic reduction ratio β assumed for the failure states. We consider 5 scenarios with different values of β varying from 60% to 100%. In the table, numbers in bold correspond to the scenarios, in which EFR performs better than RR and PD in terms of the cost. Experiments for values of β less than 60% show that for EFR and GR the cost does not increase comparing to the cost of routing the demands in the nominal state.

In Table 9.2, interesting results in terms of the cost-effectiveness are indicated. As expected, the cost of the network for EFR is often lower than C_{RR} for $b = 1$ for most of β . In fact, C_{RR} can be seen as $C_{EFR}^{+\infty}$ with $a = 0$. Hence, because increasing b decreases C_{EFR}^b , the gain of cost using EFR instead of RR comes from the ability to decrease the bandwidth of paths of unaffected demands. We also notice that the gap between the lower bound—the cost of C_{GR} and $C_{EFR}^{3.0}$ remains low, with a maximum of 12.6%.

	β	G_{RR}	G_{PD}	$G_{EFR}^{0.1}$	$G_{EFR}^{0.5}$	$G_{EFR}^{1.0}$	$G_{EFR}^{3.0}$
<i>di-yuan</i>	0.6	16.5	41.0	37.2	27.3	20.5	10.2
	0.7	19.4	47.8	43.4	31.9	23.9	11.9
	0.8	21.0	53.1	48.0	34.9	25.9	12.6
	0.9	17.5	60.1	46.9	31.6	22.0	11.4
	1.0	9.4	60.2	46.0	30.9	19.7	11.0
<i>polska</i>	0.6	14.4	47.2	35.5	20.6	15.5	7.7
	0.7	17.6	66.3	52.1	21.8	15.9	7.6
	0.8	10.5	69.0	53.6	17.3	8.1	1.8
	0.9	4.1	69.0	53.6	15.5	4.9	1.3
	1.0	0.2	69.0	53.6	14.5	4.1	1.2
<i>nobel-us</i>	0.6	14.8	59.6	48.1	31.0	23.3	11.9
	0.7	15.9	77.9	62.8	33.3	24.2	11.4
	0.8	12.5	88.3	70.6	31.9	19.7	7.4
	0.9	5.8	88.3	71.7	34.1	20.5	11.2
	1.0	1.9	88.3	71.7	33.6	16.9	8.1
<i>n-germany</i>	0.6	14.1	44.0	31.4	17.0	12.1	6.0
	0.7	10.7	51.8	38.3	19.1	15.2	12.5
	0.8	5.0	51.9	39.7	15.5	10.2	6.6
	0.9	1.6	51.9	39.7	14.8	8.0	4.1
	1.0	0.0	51.9	39.7	14.6	5.9	3.0

Table 9.2 – Relative gap (in %) for all strategies.

9.2 Heuristic performance

We now study the efficiency of the heuristic method expressed as a relative gap between the objective function of the EFR-P problem using formulation (7.1) (denoted C_{EFR}^b) and the cost achieved by the heuristic algorithm C_{EFR-H}^b . Table 9.3 shows the values of $H_{EFR}^b = \frac{C_{EFR-H}^b - C_{EFR}^b}{C_{EFR}^b} \times 100\%$ —the relative gap between C_{EFR-H}^b and C_{EFR}^b —as a function of the traffic reduction ratio. The results are given for the settings used in Table 9.2. We notice that the heuristic method performs well for all networks indicating the maximum relative gap of less than 2%. However, as this method contains some randomness, the gap with the exact solution can be different if we compute the solution of the heuristic a second time. Still, the heuristic seems to be able to find a solution close to the optimal solution, with a gap not exceeding 2% in most cases. In the table, we present the best obtained gap after running the heuristic for three times. Thus, we notice that the heuristic is usually able to find the optimal solution after just a few repetitions, as indicated by the 0.0 value of the gap (in bold).

9.3 Efficiency of EFR applied to partial failures

Finally, we study the performance of EFR_{PLF} , i.e., EFR applied to partial failures. Let C_{GR} denote the optimal value of the objective function (i.e., minimum cost of the link capacity) for GR, and $C_{EFR_{PLF}}^b$, C_{FT} the respective values for the remaining strategies. We define the relative cost increase for each strategy S (where S stands for EFR_{PLF} or FT) with respect to GR as $K_S = \frac{C_S - C_{GR}}{C_{GR}} \times 100\%$. To make a comparison with results for FT and GR possible, some vital changes are needed in the way the capacities are

	β	$H_{EFR}^{0.1}$	$H_{EFR}^{0.5}$	$H_{EFR}^{1.0}$	$H_{EFR}^{3.0}$
<i>di-yuan</i>	0.6	0.0	0.0	0.0	0.0
	0.7	0.0	0.0	0.0	0.0
	0.8	0.0	0.0	0.0	0.0
	0.9	0.0	0.0	0.0	0.0
	1.0	0.0	0.0	0.0	0.0
<i>polska</i>	0.6	0.0	0.0	0.0	0.0
	0.7	0.0	0.9	0.4	0.2
	0.8	0.0	0.6	0.7	0.0
	0.9	0.0	1.1	0.7	0.0
	1.0	0.0	1.2	0.1	0.0
<i>nobel-us</i>	0.6	0.0	0.0	0.0	0.0
	0.7	0.0	0.0	0.0	0.0
	0.8	0.0	0.0	0.0	0.1
	0.9	0.0	0.0	0.1	0.1
	1.0	0.0	0.1	0.1	0.1
<i>n-germany</i>	0.6	0.0	0.0	0.0	0.0
	0.7	1.1	0.0	0.0	0.0
	0.8	0.0	0.0	0.0	0.0
	0.9	0.0	0.0	0.0	0.0
	1.0	0.0	0.1	0.3	0.0

Table 9.3 – Relative gap (in %) between the exact solution and the heuristic solution.

affected by the perturbation for EFR_{PLF} . Instead of variables $y_e^w, e \in E, w \in W_e$ we use $y_e, e \in E$. By extension, instead of $b_e^{ws} y_e^w, e \in E, w \in W_e, s \in S$ we use $\gamma y_e, e \in E, s \in S$, where γ express the perturbation ratio. The results are given for several levels of perturbation γ : 50%, 75%, and 95%. As no network instance could be solved within a limited period of time (10 hours), we present results of network *polska* with a limited set of demands: 20 demands instead of 66.

γ	K_{FT}	$K_{EFR_{PLF}}^{0.1}$	$K_{EFR_{PLF}}^{0.5}$	$K_{EFR_{PLF}}^{1.0}$	$K_{EFR_{PLF}}^{3.0}$
50%	14.3	11.3	6.6	5.0	3.3
75%	26.0	20.9	10.9	7.8	3.9
95%	43.9	35.8	18.3	11.6	7.0

Table 9.4 – Relative gap (in %) of EFR_{PLF} and FT compared to GR.

We notice that EFR_{PLF} performs generally better than FT. For lower perturbation ratios, for instance $\gamma = 50\%$, differences of costs of the network for FT and EFR_{PLF} remains insignificant—the gap between FT and EFR_{PLF} for all values of b is less than 10%. However, if we consider stronger perturbation ratios, for instance $\gamma = 95\%$, EFR_{PLF} becomes clearly more cost-efficient than FT. When the thickening ratio is very low, for instance 10%, the gap between FT and EFR_{PLF} is equal to 5.7%¹. If we increase the thickening ratio, the gap between FT and EFR_{PLF} reaches 25.7% ($b = 3$).

¹This gap value is computed using results for each strategy as done for the gap with GR.

9.4 Restoration time

We here analyze the solutions given by the EFR, GR and RR strategies in terms of average number of paths, length of the paths and restoration time. Table 9.5 shows the average number of paths in the nominal state, the total number of distinct paths and the number of paths to thick or create. Moreover, the Table shows the average length of the paths, in the nominal state and in the failure states. We notice that even though GR and RR use less paths in the nominal state; nevertheless, the average total number of paths used by these strategies is significantly higher. Note that, for EFR, increasing b seems to decrease the average number of paths and length of paths.

	$EFR^{0.5}$	$EFR^{1.0}$	GR	RR
Avg number of paths per demand in NS	4.12	3.02	1.27	1.51
Avg number of distinct paths per demand	4.12	3.02	7.26	7.36
Avg number of paths to thicken/create per demand	0.94	0.61	1.24	0.41
Avg length of the paths in NS	3.41	3.14	3.13	2.46
Avg length of all paths	3.41	3.14	4.44	5.57

Table 9.5 – Paths analysis on the solutions obtained by EFR, GR and RR for network *pol ska*

In the following, we study the restoration time of EFR, compared to GR and RR. In the experiment, we use a simplified restoration model that returns a fraction of satisfied demands in a function of time elapsed after the failure. The results are presented in Figures 9.1 and 9.2, where the former represents the ratio of satisfied demands in time in the network *pol ska* for a total failure of a link at the border of the network, while the latter presents the same result for a total failure of a link in the center of the network.

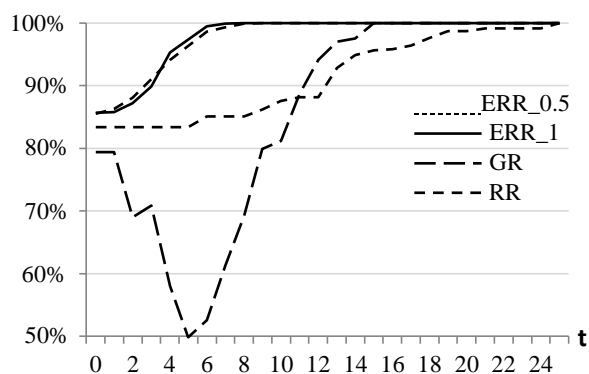


Figure 9.1 – Demand satisfaction ratio of the network *pol ska* after failure of Kolobrzeg-Szczecin (at the border of the network).

As expected, EFR leads to a substantial gain in terms of restoration time, roughly two times faster than GR and three times faster than RR. This is mainly due to two factors: EFR do not create new paths and the total number of paths it uses is smaller. Therefore, the time when EFR practically ends the whole restoration process is also the time when GR

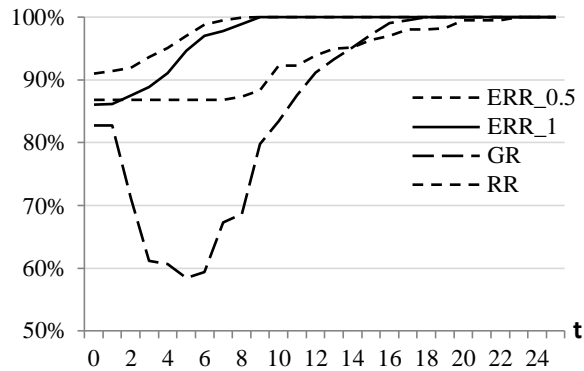


Figure 9.2 – Demand satisfaction ratio of the network *pol ska* after failure of *Lodz-Warsaw* (in the center of the network).

strategy ends disconnecting flows that are to be rerouted. The reason GR outperforms RR in terms of total restoration time is the average length of restoration paths used by the strategies. The longer restoration paths of RR strategy induce longer restoration times. Another important aspect, mentioned earlier, is the capability of EFR to maintain a high level of satisfaction ratio of demands during the restoration process. From the figures we notice that GR encumbers the process as a lot of routing paths should first be released and others have to be settled from the scratch, while EFR benefits from optimized flow paths at a large part available in most of failure situations. The issue can be perfectly seen in the figures around time 5-7, when the demand satisfaction ratio for GR plunges to merely 50%, while at the same time reaching 100% for EFR and about 85% for RR. This feature is expected to be even more important for partial failure situations.

9.5 Conclusion on the efficiency of EFR

When it comes to the relation of link capacity cost, management effort, and traffic restoration time, the EFR and EFR_{PLF} strategies appear to be encouraging, especially when traffic is supposed to be partially restored. The heuristic seems to perform well, leading to optimal or near-optimal solutions for most network instances used in our tests.

Conclusion and future work

The work done on this thesis led to the design of two recovery mechanisms: *Flow Thinning* (FT) and *Elastic Flow Rerouting* (EFR).

FT is an original concept of a traffic routing and protection strategy for communication networks with variable link capacity. Its affine variant, AFT, opens a way for distributed implementations. To our knowledge, similar strategies (working at the logical tunnel level and reacting online to link capacity fluctuations) have not been proposed and therefore studying optimization models for FT and AFT is worthwhile, the more that the related multicommodity flow problems contribute to the area of resilient network design. We have studied in-depth a basic FT optimization problem (FTOP, see Section 4.2), and its affine version (AFTOP, see Section 6.1). FTOP turns out to be NP -hard and its non-compact link-path LP formulation requires path generation for which we have found a quite effectively solvable (though NP -hard) pricing problem. We have also exhibited some special cases when path generation is polynomial and formulated a polynomially solvable algorithm for finding accurate lower bounds for FTOP. The numerical study shows that in terms of cost/traffic efficiency, FT exhibits cost not significantly superior to that of GR—the theoretically most cost efficient strategy, at least when (reasonable) demand reduction coefficients are acceptable.

EFR is based on the concept of elastic (failure state-dependent) path-flows. EFR is designed to handle total link failures, while its extension, called EFR_{PLF} , deals with multiple partial link failures related to broadband wireless (FSO, microwave) networks. When it comes to the relation of link capacity cost, management effort, and traffic restoration time, the EFR strategy appears to be encouraging, especially when traffic is supposed to be partially restored. From the theoretical viewpoint, optimization of EFR gives rise to a challenging multi-commodity flow problem EFR-P, combining two difficult issues: a combinatorial subproblem dealing with the choice of the demand states, and a likely NP -hard pricing subproblem for path generation. To the best of our knowledge, such a problem has not been considered in the literature on traffic protection strategies before. We have introduced an exact MIP formulation of EFR-P that works for a given set of admissible paths, and proposed a fast suboptimal heuristic approach for EFR-P. Looking for an exact approach with a reasonable efficiency will be the subject of our future work. We have also considered the case with partial failures. The corresponding problem, denoted by EFR_{PLF-P} , is even more difficult, since it combines two levels of decisions: at the first level it chooses flows that have to be thinned and at the second level it optimizes the way the affected demands can enlarge their flows using resources released from the unaffected demands. All this makes the problem highly combinatorial. Additionally, the corresponding

pricing problem is very complicated, as it embraces features coming from two other NP -hard pricing problems: one for FT and the other for EFR.

The above strategies are designed under the consideration of management cost hypotheses. A further study on implementation issues is needed to assess the relevance of such considerations. At this very moment, (July 2015), members of IETF are currently working on a way to manage partial link failures on MPLS networks (Long et al. 2015) and an extension of RSVP-TE signaling is proposed. This extension can be used to set up a label switching path (LSP) that contains links with discretely variable bandwidth. Each link has a set of <capacity, availability> information and paths will have a set of <capacity, availability> requirements. In accordance with above, we are currently looking how the proposed strategies can be implemented.

Finally, from an optimization point of view, the heuristic method for EFR may be improved to obtain solutions of lower cost and an affine version could lead to a problem easier to solve and to a solution resilient to more failure scenarios. Next, the principle of EFR_{PLF} can be applied to other reroute get strategies, namely RR. Then, it is possible to apply FT as proposed and handle the remaining lost traffic with RR. Finally, one of difficulties that one will encounter when dealing with partial failures in fixed wireless networks is the extremely large number of failure states with respect to weather conditions to be considered. Hence, this calls for robust optimization methods covering a very large part of these states while providing high restoration ratio for the remaining states. We are now looking for an appropriated way of failure state generation through a constraint generation process in an iterative algorithm.

APPENDICES

A

A.1 MPLS Networks

MPLS (Multi-Protocol Label Switching) (Rosen et al. 2001) is a mechanism in telecommunications networks which combines IP routing concept of level 3, and level 2 switching mechanisms as implemented in ATM or Frame Relay. The aim of this section is to explain how data are routed and rerouted in such a network. We will not explain MPLS in deep but only the main features as label switching principle, routing protocol and its "standard" rerouting algorithm.

A.1.1 Label Switching

When a packet enters a MPLS network (1), depending of its FEC (Forwarding equivalence class), the ingress node reads its switching table (2), assigns a label to this packet (3), and forwards it to the following LSR (4).

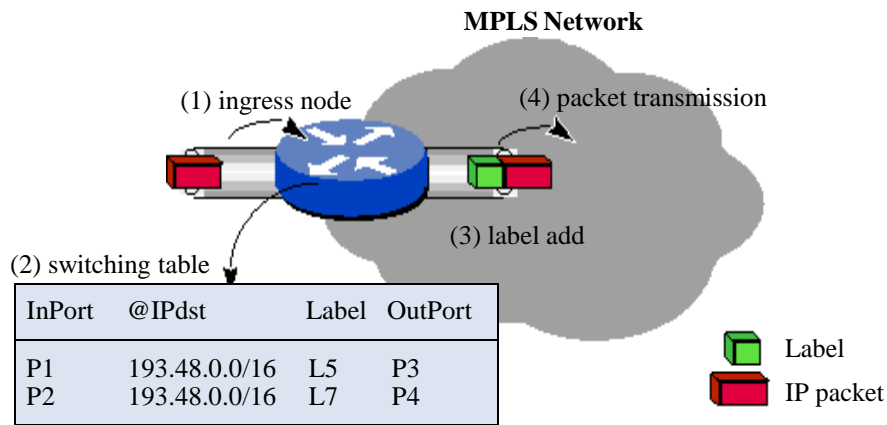


Figure A.1 – Ingress Node in MPLS network.

When the packet arrives in a LSR inside the MPLS network (1), the routing protocol running on this node finds, in its label database (Label Base Information), the next label to assign to this packet in order to send it to its destination (2). The LSR updates the MPLS header of the packet

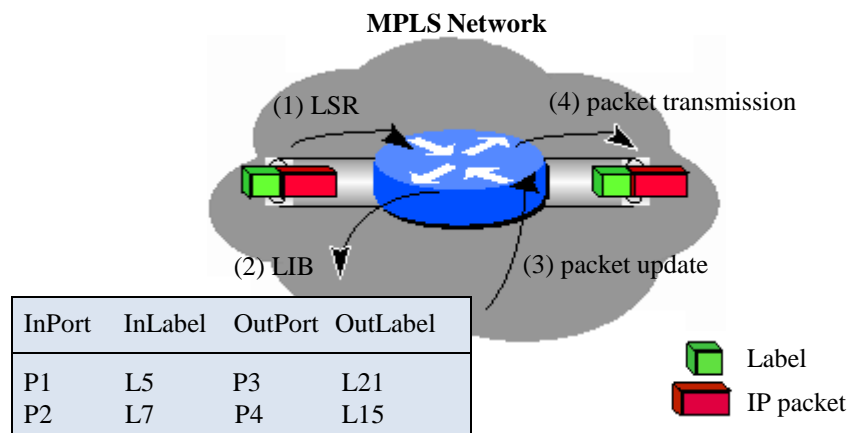


Figure A.2 – Label Switch Router (LSR) in MPLS network.

(switch the label, update the TTL field ...) (3) and sends it to the following

node (LSR or egress node) (4). We can notice that in an internal LSR, the routing protocol of the network layer is never used.

Finally, once the MPLS packet arrives in the egress node (1), the router removes all MPLS labels (2) and transmits the packet to the network layer.

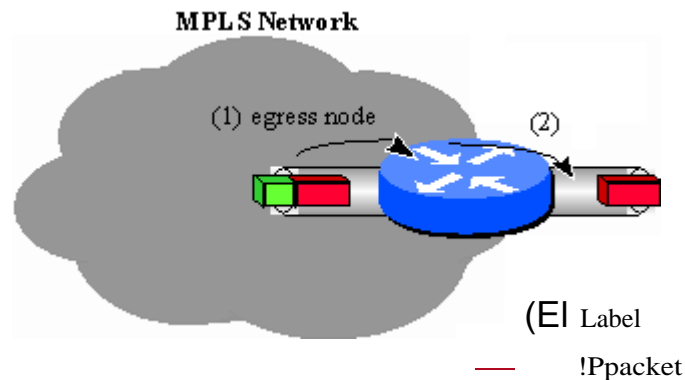


Figure A.3- Egress Node in MPLS network.

Thus, we notice that the routing of a demand is made between its extremity nodes. We now need to find a routing algorithm to find all needed LSPs to route the traffic.

A.1.2. Routing Principle

As described above, the routing strategy is based on label switching. A LSR in the MPLS network switches labels by studying the incoming label and port, reads the switching table and then sends the packet to the next node. Labels are assigned to a packet only once in the border of the MPLS network by an ingress node E-LSR (Edge Label Switch Router) where a calculation is done on the datagram to find the specific label. This calculation is done only once by the ingress node, depending of its destination. Thus, the ingress node chooses the end-to-end routing of the demands.

The routing of a demand is done using RSVP-TE (Resource Reservation Protocol-Traffic Engineering) (Awduche et al. 2001). We will not go further into this routing protocol but we will point out the main characteristics of this routing. Each demand may be routed by one or several LSP(s), and every LSP associated to a flow may be configured in order to ensure QoS.

A.2 EFR - Implementation issues

The numerical results clearly show that both "centralized" and distributed ShRR strategies are cost-effective, with results close to that of global rerouting (GR), which is a lower bound of the cost of a network for all routing strategies. In several cases EFR strategies are also more cost-effective than end-to-end rerouting with stub release (RR). Nevertheless, the gap between these strategies and ShRR remains significant. Hence, a natural question here would be: what is the interest of EFR/DEFR strategies? The answer is directly linked to the management cost these strategies will have in practice, as discussed below. Let us focus first on the distributed variant of ShRR, that is to say DShRR. This strategy uses only extremity nodes of affected demands to initiate the recovery process after a failure. We notice that when a failure occurs new paths may have to be created. This action can only be performed by the source node of the demand. This creation of a new path is preceded by decreasing the bandwidth of some paths, which can be performed by both source and destination nodes. Unlike the DShRR strategy, DERR does not need to create new paths to recover the traffic after a failure. This strategy only increases or decreases the bandwidth of some (possibly) affected paths to recover the traffic. It is a feature of great interest, since the restoration delay is significantly reduced in comparison to DShRR. To show this, let us describe in detail the process of creation of a new path.

Depending on the transport protocol, establishing a new path from the node source to its destination will have a cost which greatly depends on the number of links. Let us look how this is accomplished using the Resource Reservation Protocol (RSVP) (see (Braden et al. 1997, Braden and Zhang 1997)). First, the source node sends a PATH message, which contains the required flow parameters (TSPEC), to the destination node using the routing protocol. Then, every visited router has two operations to perform: storing the PATH-STATE, i.e. the message received from the last node to have transmitted the message, and modifying the ADSPEC field, i.e. the flow parameters, when the required bandwidth is not available. When the destination node receives the PATH message it uses both the TSPEC and the ADSPEC fields to determine the parameters of the flow, and sends the flow descriptor, using a RESV message, to the source node using the same path as the PATH message. Finally, the source node sends a RESV-CONF message to the destination node validating the opening of the flow. This means that the time needed to create a new path is almost 3 times as long as the time needed to transfer data from the source to the destination node (c denotes the end-to-end transfer time). In contrast, the time required for decreasing or increasing the bandwidth is significantly lower. In order to maintain the flow, periodically, the source node sends a PATH message to the destination node, and the destination node sends a RESV message to the source node. The flow parameters included in these messages can be modified to increase or decrease the bandwidth of the path. We will use this property to perform bandwidth modification for DShRR and DERR. Hence, if the destination node of some affected demands initiates the decreasing (resp. decreasing/increasing) process for DShRR (resp. DEFR), the time needed will be equal to c . This is possible,

since the source can use the updated path immediately after receiving the PATH message from the destination node (Figure A.4.a). If it is the source node that initiates the process, the time needed to modify the bandwidth is equal to $2c$, because the source node has to wait for the confirmation before using the updated path (Figure A.4.b). It will be remarked that this procedure cannot be employed to create a new path.

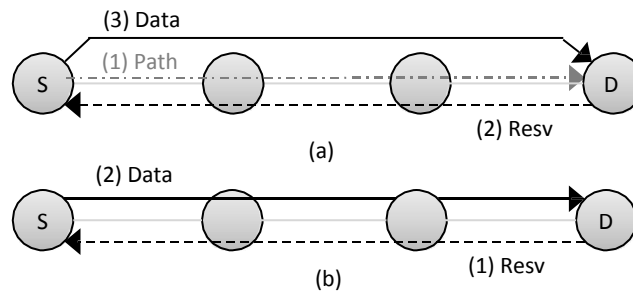


Figure A.4 – Bandwidth increase procedure, managed by source node (a) and destination node (b) for DEFR.

In both strategies it is important to begin by decreasing the bandwidth of paths of non-affected demands, before increasing the bandwidth of non-affected paths of the affected demands. Indeed, before the bandwidth of any path is increased it must be certain that there is sufficient available bandwidth, otherwise the process will fail and must be restarted. This situation is depicted in Figure A.5.

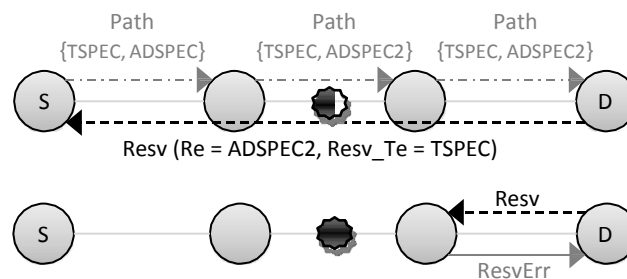


Figure A.5 – Failed attempt to increase the bandwidth of a path managed by source node (up) and destination node (down).

The DEFR strategy would thus appear to be faster than other end-to-end rerouting strategies that employ new rerouting paths, including DShRR.

A.3 SNDlib - Graph details

SNDlib (Orlowski et al. 2010) is a library of test instances for Survivable fixed telecommunication Network Design. Its purpose is to make realistic network design test instances available to the research community and to serve as a standardized benchmark for testing, evaluating, and comparing network design models and algorithms. In the document, we use undirected networks *pdh*, *di-yuan*, *polska*, *nobel-us* and *nobel-germany*. Table A.1 represents the number of vertices, edges and demands of each networks, as well as the maximum and minimum degree of the corresponding graphs ($\Delta(G)$, $\delta(G)$).

<i>network</i>	$ V $	$ E $	$ D $	$\Delta(G)$	$\delta(G)$
<i>pdh</i>	11	34	24	9	7
<i>di-yuan</i>	11	42	22	9	7
<i>polska</i>	12	18	66	5	2
<i>nobel-us</i>	14	21	91	6	2
<i>nobel-germany</i>	17	26	121	4	2

Table A.1 – Description of SNDLib Network instances

In the sequel of this section, we present the topology of all five networks.

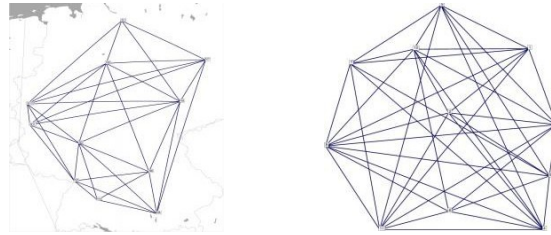


Figure A.6 – Topology of networks *pdh* (left) and *di-yuan* (right).

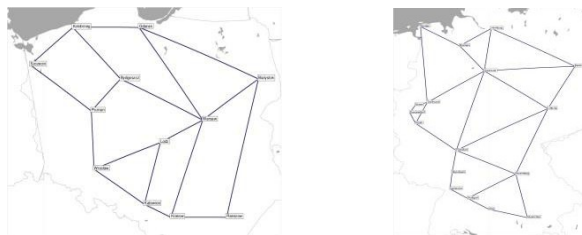


Figure A.7 – Topology of networks *polska* (left) and *nobel-germany* (right).



Figure A.8 – Topology of network *nobel-us*.

Bibliography

- R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993. (Cité page 51.)
- H. Anderson. Fixed broadband wireless system design. *J. Wiley & Sons*, 2003. (Cité page 3.)
- L. Andersson and G. Swallow. RFC3468: The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols, February 2003. (Cité page 84.)
- D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. Rsvp-te: Extensions to rsvp for lsp tunnels, December 2001. (Cité pages 84 and 131.)
- A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004. (Cité page 82.)
- J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(3):238–252, 1962. (Cité page 29.)
- J. Berthold, A. A. M. Saleh, L. Blair, and J. M. Simmons. Optical networking: Past, present, and future. *Journal of Lightwave Technology*, 26(9):1104–1118, May 2008. (Cité page 5.)
- Q. Botton, B. Fortz, L. Gouveia, and M. Poss. Benders decomposition for the hop-constrained survivable network design problem. *Inform's Journal on Computing*, 25(1):13–26, 2011. (Cité page 6.)
- R. Braden and L. Zhang. Resource reservation protocol (rsvp) – message processing rules, 1997, September 1997. (Cité page 132.)
- R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource reservation protocol (rsvp) – version 1 functional specification, September 1997. (Cité page 132.)
- G. Claßen, D. Coudert, A. Koster, and N. Nepomuceno. Bandwidth assignment for reliable fixed broadband wireless networks. *12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, June 2011. (Cité pages 8 and 19.)
- D. Coudert, P. Datta, S. Perennes, H. Rivano, and M.-E. Voge. Shared risk resource group: Complexity and approximability issues. *Parallel Processing Letters*, 17(6):169–184, 2007. (Cité page 56.)

- CPLEX. *IBM ILOG CPLEX 12.5 Reference Manual*. ILOG CPLEX Division, Gentilly, France, 2013. (Cité page 68.)
- D. Dhody, U. Palle, and R. Singh. Pcep extensions for mpls-te lsp automatic bandwidth adjustment with stateful pce, March 2015. (Cité pages 8 and 21.)
- J. Evans and C. Filsfils. *Deploying IP and MPLS QoS for Multiservice Networks – Theory and Practice*. Morgan-Kaufmann, 2007. (Cité page 84.)
- S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111–121, 1980. (Cité page 103.)
- B. Fortz, M. Labbé, and A. Violin. Dantzig-wolfe reformulation for the network pricing problem with connected toll arcs. *Electronic Notes in Discrete Mathematics*, 41:117–124, 2013. (Cité page 6.)
- Y. Fouquet and D. Nace. Elastic flow rerouting - a complexity study. In *Proceedings of the 7th IFIP International Conference on New Technologies, Mobility and Security (NTMS 2015)*, Paris, France, July 2015. (Cité pages vii and ix.)
- Y. Fouquet, D. Nace, M. Pióro, and M. Poss. Flow adjustment - a flexible routing strategy for demand protection against multiple partial link failures. In *Proceedings of the 4th International Conference on Advanced Communications and Computation (INFOCOMP 2014)*, Paris, France, July 2014a. (Cité pages vii, ix, 87, and 89.)
- Y. Fouquet, D. Nace, M. Pióro, M. Poss, and M. Żotkiewicz. An optimization model for communication networks resilient to multiple partial link failures. In *Proceedings of the 12th INFORMS Telecommunications Conference (INFORMS Telecom 2014)*, Lisbon, Portugal, March 2014b. (Cité pages vii and ix.)
- Y. Fouquet, D. Nace, M. Pióro, M. Poss, and M. Żotkiewicz. Flow adjustment methods for survivable networks. In *Proceedings of the 16th International Telecommunications Network Strategy and Planning Symposium (NETWORKS 2014)*, Funchal, Madeira Island, Portugal, 2014c. (Cité pages vii, ix, 87, and 89.)
- Y. Fouquet, D. Nace, M. Pióro, M. Poss, and M. Żotkiewicz. Elastic rerouting - models and resolution methods. In *Proceedings of the 7th International Workshop on Reliable Networks Design and Modeling (RNDM 2014)*, Barcelona, Spain, October 2014d. (Cité pages vii and ix.)
- Y. Fouquet, D. Nace, M. Pióro, and S. Shinko. Elastic routing: a distributed variant, implementation issues, and numerical results. In *Proceedings of the 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC 2013)*, Compiègne, France, October 2014e. (Cité pages vii, ix, 8, and 97.)
- Y. Fouquet, D. Nace, M. Pióro, and M. Poss. Elastic flow rerouting : a novel rerouting strategy resilient to multiple partial link failures. In

- Proceedings of the 7th International Network Optimization Conference (INOC 2015)*, Warsaw, Poland, May 2015a. (Cité pages vii and ix.)
- Y. Fouquet, D. Nace, M. Pióro, M. Poss, and M. Żotkiewicz. Generalized elastic flow rerouting scheme (submitted). *Networks*, 2015b. (Cité pages vii, ix, and 97.)
- A. Fundo, A. Bashllari, D. Nace, and I. Shinko. A hybrid rerouting scheme. *Telecommunication Systems*, 56(1):69–78, August 2013. (Cité pages 7, 18, and 42.)
- A. Goldsmith and S. Chua. Variable-rate variable-power m-qam for fading channels. *IEEE Transactions on Communications*, 45(10):1218–1230, 1997. (Cité page 4.)
- M. Grotschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer Verlag, Berlin, 1988. (Cité pages 49 and 51.)
- H. Harhira and S. Pierre. A mathematical model for the admission control problem in MPLS networks with end-to-end delay guarantees. In *Proceedings of the 16th International Conference on Computer Communications and Networks (ICCCN 2007)*. Honolulu, Hawaii, USA, August 2007. (Cité page 84.)
- S. Harjeevan and M. Rajan. Performance analysis of terrestrial free space optical (fso) communication link using m-qam modulation technique. *International Journal of Scientific and Engineering Research*, 4(6):805–808, June 2013. (Cité page 5.)
- D. Katz, K. Kompella, D. Gan, and D. Yeung. Rfc3630: Traffic engineering (te) extensions to ospf version 2, September 2003. (Cité page 84.)
- J. Kennington. A survey of linear cost multicommodity network flows. *Operation Research*, 26(2):209–236, 1978. (Cité page 6.)
- L. Khachian. A polynomial algorithm for linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979. (Cité page 56.)
- B. Korte and J. Vygen. *Combinatorial Optimization Theory and Algorithms*. Springer, 2012. (Cité page 57.)
- A. Koster, A. Zymolka, M. Jäger, and R. Hülsermann. Demand-wise shared protection for meshed optical networks. *Journal of Network and Systems Management*, 13(1):35–55, March 2005. (Cité page 15.)
- L. Lasdon. *Optimization Theory for Large Systems*. MacMillan, 1970. (Cité pages 26, 49, 51, and 55.)
- H. Lehpamer. Microwave transmission networks: Planning, design, and deployment. *McGraw-Hill*, 2010. (Cité page 3.)
- H. Long, M. Ye, G. Mirsky, A. D’Alessandro, and H. Shah. Rsvp-te signaling extension for links with variable discrete bandwidth, July 2015. (Cité page 128.)

- J.-C. Lutton, D. Nace, and J. Carlier. Assigning spare capacities in mesh survivable networks. *Telecommunication Systems*, 13(2-4):441–451, 2000. (Cité page 32.)
- E. Mannie and D. Papadimitriou. Recovery (protection and restoration) terminology for generalized multi-protocol label switching (gmpls), March 2006. (Cité pages 7 and 14.)
- J.-F. Maurras and S. Vanier. Network synthesis under survivability constraints. *4OR*, 2(1):53–67, 2004. (Cité page 110.)
- A. Mereu, D. Cherubini, A. Fanni, and F. A. Primary and backup paths optimal design for traffic engineering in hybrid igp/mpls networks. In *Proceedings of the 7th International Workshop on Design of Reliable Communication Networks (DRCN 2009)*, pages 273–280, Washington, DC, October 2009. (Cité page 110.)
- I. Minei and J. Lucek. *MPLS-Enabled Applications: Emerging Developments and New Technologies*. John Wiley & Sons, 2011. (Cité page 84.)
- M. Minoux. *Mathematical Programming: Theory and Algorithms*. John Wiley & Sons, 1986. (Cité pages 26, 49, 51, and 55.)
- M. Minoux. Multicommodity network flow models and algorithms in telecommunications. *Handbook of Optimization in Telecommunications*, pages 163–184, 2006. (Cité page 6.)
- T. S. Motzkin. Two consequences of the transposition theorem on linear inequalities. *Econometrica*, 19:184–185, 1951. (Cité page 51.)
- D. Nace, M. Pióro, A. Tomaszewski, and M. Żotkiewicz. A polynomial multicommodity flow problem with difficult path generation. In *Proceedings of the 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT 2011)*, Budapest, Hungary, October 2011. (Cité page 17.)
- D. Nace, M. Pióro, A. Tomaszewski, and M. Żotkiewicz. Complexity of a classical flow restoration problem. *Networks*, 62(2):149–160, 2013. (Cité pages 17, 102, 103, 105, and 110.)
- G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988. (Cité pages 49 and 51.)
- S. Orłowski. Local and global restoration of node and link failures in telecommunication networks, phd thesis, 2003. (Cité page 110.)
- S. Orłowski and M. Pióro. Complexity of column generation in network design with path-based survivability mechanisms. *Networks*, 59:132–147, 2012. (Cité pages 42, 48, 56, 102, and 110.)
- S. Orłowski, R. Wessäly, M. Pióro, and A. Tomaszewski. SNDlib 1.0 – survivable network design library. *Networks*, 55(3):276–286, 2010. (Cité pages 35, 67, 88, 121, and 134.)

- A. Ouorou and J.-P. Vial. A model for robust capacity planning for telecommunications networks under demand uncertainty. In *Proceedings of the 6th International Workshop on Design and Reliable Communication Networks (DRCN 2007)*, pages 1–4, 2007. (Cité page 82.)
- P. Pan, G. Swallow, and A. Atlas. Fast reroute extensions to rsvp-te for lsp tunnels, May 2005. (Cité pages 8 and 19.)
- M. Pióro and D. Medhi. Routing, flow, and capacity design in communication and computer networks. *Morgan-Kaufman*, 2004a. (Cité pages 7 and 15.)
- M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan-Kaufman, 2004b. (Cité pages 48, 86, 108, and 122.)
- M. Pióro, D. Nace, and Y. Fouquet. On protected traffic routing in wireless networks with partial multiple link failures. In *Proceedings of the 8th International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA 2013)*, Compiègne, France, October 2013a. (Cité pages vii and ix.)
- M. Pióro, D. Nace, and Y. Fouquet. An optimization model for multicommodity flow networks with partial multiple link failures. In *Proceedings of the 5th International Workshop on Reliable Networks Design and Modeling (RNDM 2013)*, Almaty, Kazakhstan, September 2013b. (Cité pages vii, ix, 8, and 42.)
- M. Pióro, Y. Fouquet, D. Nace, and M. Poss. Optimizing flow thinning protection in multicommodity networks with variable link capacity (submitted). *Operation Research (OR)*, 2015. (Cité pages vii, ix, 8, and 82.)
- M. Poss and C. Raack. Affine recourse for the robust network design problem: between static and dynamic routing. *Networks*, 61(2):180–198, 2013. (Cité pages 82 and 83.)
- A. L. Puryear. Optical communication through the turbulent atmosphere with transmitter and receiver diversity, wavefront control, and coherent detection, phd thesis, 2011. (Cité page 5.)
- E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture, January 2001. (Cité page 130.)
- I. Shinko, Y. Fouquet, and D. Nace. Elastic routing for survivable networks. *International Journal of Grid and Utility Computing (IJGUC)*, 6(2), 2015. (Cité pages vii and ix.)
- S. Shinko, Y. Fouquet, and D. Nace. A study on a distributed rerouting scheme. In *Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA 2013)*, Barcelona, Spain, March 2013. (Cité pages vii and ix.)
- I. K. Son and S. Mao. Design and optimization of a tiered wireless access network. In *IEEE INFOCOM*, pages 1–9, 2010. (Cité pages 3 and 113.)

-
- A. Tomaszewski. The final answer to the complexity of a basic problem in resilient network design. *Electronic Notes in Discrete Mathematics*, 41: 455–462, 2013. (Cité page 102.)
- A. Tomaszewski. Design of optical wireless networks with fair traffic flows. *Journal of Applied Mathematics*, 14, June 2014. (Cité pages 8 and 20.)
- A. Tomaszewski, M. Pióro, and M. Żotkiewicz. On the complexity of resilient network design. *Networks*, 55:109–118, 2010. (Cité pages 33 and 47.)
- H. Willebrand and B. S. Ghuman. Free space optics: Enabling optical connectivity in today's networks. *Sams Publishing*, 2002. (Cité pages 3 and 113.)
- Z. Xiaoming and J. M. Kahn. Free-space optical communication through atmospheric turbulence channels. *IEEE Transactions on Communications*, 50(8):1293–1300, August 2002. (Cité page 5.)

List of Publications

Journal Article

M. Pióro, Y. Fouquet, D. Nace and M. Poss, Optimizing flow thinning protection in multicommodity networks with variable link capacity, *Operations Research* (submitted).

Y. Fouquet, D. Nace, M. Pióro, M. Poss and M. Żotkiewicz, Generalized Elastic Flow Rerouting Scheme, *Networks*, 2015.

I. Shinko, Y. Fouquet and D. Nace, Elastic routing for survivable networks, *International Journal of Grid and Utility Computing (IJGUC)*, 6(2), 2015.

Invited Conference

Y. Fouquet and D. Nace, Elastic Flow Rerouting - a complexity study, *In Proceedings of the 7th IFIP International Conference on New Technologies, Mobility and Security (NTMS 2015)*, 2014, Paris, France.

Y. Fouquet, D. Nace, M. Pióro, M. Poss and M. Żotkiewicz, Flow adjustment methods for survivable networks, *In Proceedings of the 16th International Telecommunications Network Strategy and Planning Symposium (NETWORKS 2014)*, Funchal, Madeira Island, Portugal.

In Proceedings

Y. Fouquet, D. Nace, M. Pióro and M. Poss, Elastic Flow Rerouting: a novel rerouting strategy resilient to multiple partial link failures, *In Proceedings of the 7th International Network Optimization Conference (INOC 2015)*, 2015, Warsaw, Poland.

Y. Fouquet, D. Nace, M. Pióro, M. Poss and M. Żotkiewicz, Elastic routing: models and resolution methods, *In Proceedings of the 6th International Workshop on Reliable Networks Design and Modeling (RNDM 2014)*, 2014, Barcelona, Spain.

Y. Fouquet, D. Nace, M. Pióro and M. Poss, Flow Adjustment - a flexible routing strategy for demand protection against multiple partial link failures, *In Proceedings of the 4th International Conference on Advanced Communications and Computation (INFOCOMP 2014)*, 2014, Paris, France.

M. Pióro, Y. Fouquet, D. Nace, M. Poss and M. Żotkiewicz, An optimization model for communication networks resilient to multiple partial link

failures, *In Proceedings of the 12th INFORMS Telecommunications Conference (INFORMS Telecom 2014)*, 2014, Lisbon, Portugal.

M. Pióro, D. Nace and Y. Fouquet, An optimization model for multi-commodity flow networks with partial multiple link failures, "Best Paper Award", *In Proceedings of the 5th International Workshop on Reliable Networks Design and Modeling (RNDM 2013)*, 2013, Almaty, Kazakhstan.

Y. Fouquet, D. Nace, M. Pióro and I. Shinko, Elastic routing scheme: Distributed variant, implementation issues *and numerical results*. In Proceedings of the 8th International Conference on Broadband, Wireless Computing, Communication and Applications (3PGCIC 2013), 2013, Compiègne, France.

M. Pióro, D. Nace and Y. Fouquet, On protected traffic routing in wireless networks with partial multiple link failures, *In Proceedings of the 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (BWCCA 2013)*, 2013, Compiègne, France.

I. Shinko, Y. Fouquet and D. Nace, A study on a distributed rerouting scheme, *In Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA 2013)*, 2013, Barcelona, Spain.

List of Abbreviations

DL	double-link (failure scenario), includes SL
DSP	dual separation problem
EFR	elastic flow rerouting for total link failures
EFR _{PLF}	elastic flow rerouting for partial link failures
FA	flow adjustment
FT	flow thinning
FTOP	flow thinning optimization problem
FTOP ^{<}	flow thinning optimization problem with predefined admissible paths
GR	global rerouting
LBP	lower bound problem
LP	linear programming
LR	local rerouting
LSP	label-switched path
MILP	mixed integer linear programming
MPLF	multiple partial link failures
PD	path diversity
PG	path generation
PP	pricing problem
RR ⁺ or RR	restricted restoration with stub release
RR ⁻	restricted restoration without stub release
SL	single-link (failure scenario)
SN	single-node (failure scenario)
TL	triple-link (failure scenario), includes DL

List of Figures

1.1	Adverse weather conditions decrease wireless channel capacity.	5
1.2	Partial failures in logical layers of wired networks.	6
2.1	Classification of recovery mechanisms from the literature.	12
2.2	The network.	13
2.3	The demands.	14
2.4	1+1 Protection before (left) and after (right) failure of link (B,E).	14
2.5	N:M Protection before (left) and after (right) failure of link (B,E).	15
2.6	PD before (left) and after (right) failure of link (B,E).	16
2.7	Local Rerouting before (left) and after (right) failure of link (B,E).	16
2.8	Global Rerouting before (left) and after (right) failure of link (B,E).	17
2.9	RR ⁺ before (left) and after (right) failure of link (B,E).	17
2.10	RR ⁻ before (left) and after (right) failure of link (B,E).	18
2.11	ShRR before (left) and after (right) failure of link (B,E).	18
2.12	MPLS Fast Reroute before (left) and after (right) failure of link (B,E).	19
2.13	PR before (left) and after (right) failure of link (B,E).	20
2.14	ETF before (left) and after (right) failure of link (B,E).	20
2.15	GR before (left) and after (right) failure of link (B,E).	21
3.1	Relative gap (in %) for all strategies for network polska.	36
3.2	Relative gap (in %) for all strategies for network nobel-us.	36
3.3	Relative gap (in %) for all strategies for network nobel-germany.	37
4.1	FTS - Nominal State.	43
4.2	FTS - Failure of link (C, D).	43
4.3	FTS - Failure of link (A, B).	44
4.4	Evolution of the optimal cost of the network using FTS depending on the failure coefficient.	44
4.5	A simple network example	45
4.6	A network example with a loop in the optimum link-flow	59
4.7	Original graph G	61
4.8	Transformed graph G(d)	61
4.9	A network example	65

5.1	Rainfall radar over Poland for network <i>polska</i>	68
5.2	Relative gap between GR and FTS depending on the availability coefficient α for network <i>pdh</i> (link failures).	69
5.3	Relative gap between GR and FTS depending on the availability coefficient α for network <i>polska</i> (link failures).	69
5.4	Relative gap between GR and FTS depending on the availability coefficient α for network <i>nobel-us</i> (link failures).	70
5.5	Relative gap between GR and FTS depending on the availability coefficient α for network <i>nobel-germany</i> (link failures).	70
5.6	Relative gap between GR and FTS depending on the availability coefficient α for network <i>polska</i> (node failures).	71
5.7	Relative gap between GR and FTS depending on the availability coefficient α for network <i>nobel-us</i> (node failures).	71
5.8	Relative gap between GR and FTS depending on the availability coefficient α for network <i>nobel-germany</i> (node failures).	71
5.9	Relative gap between GR and FTS depending on the availability coefficient α for network <i>polska</i> (failures of two links in close proximity).	72
5.10	Relative gap between GR and FTS depending on the availability coefficient α for network <i>polska</i> (double links failures).	73
5.11	Value of the objective function and number of paths added at each iteration for <i>pdh</i>	75
5.12	Value of the objective function and number of paths added at each iteration for <i>polska</i>	75
5.13	Value of the objective function and number of paths added at each iteration for <i>nobel-us</i>	76
6.1	Relative gap between FAS and GR for <i>di-yuan</i>	88
6.2	Relative gap between FAS and GR for <i>polska</i>	89
6.3	Relative gap between FAS and GR for <i>nobel-us</i>	89
6.4	Relative gap between FAS and GR for <i>nobel-germany</i>	89
6.5	Average number of distinct paths per demand used by FAS per single partial link failure ($\alpha = 0.5$).	91
6.6	Average number of distinct paths per demand used by FAS per single total link failure ($\alpha = 0$).	91
6.7	Evolution of the ratio of perturbed demands for <i>polska</i> ($\alpha = 0.5$).	92
6.8	Evolution of the ratio of perturbed demands for <i>polska</i> ($\alpha = 0$).	92
7.1	EFR - Nominal State.	99
7.2	EFR - Failure of link (A, C)	99
7.3	EFR - Failure of link (A, E)	99
7.4	Network instance proving NP -hardness of EFR.	104
8.1	EFR - Nominal State.	114
8.2	EFR - Partial Failure of link (A, E)	115
8.3	EFR - Nominal State.	119
8.4	EFR - Partial Failure of link (B, E)	120
9.1	Demand satisfaction ratio of the network <i>polska</i> after failure of Kolobrzeg-Szczecin (at the border of the network).	125

9.2	Demand satisfaction ratio of the network <i>po/ska</i> after failure of Lodz-Warsaw (in the center of the network).	126
A.1	Ingress Node in MPLS network.	130
A.2	Label Switch Router (LSR) in MPLS network.	130
A.3	Egress Node in MPLS network.	131
A.4	Bandwidth increase procedure, managed by source node (a) and destination node (b) for DEFR.	133
A.5	Failed attempt to increase the bandwidth of a path managed by source node (up) and destination node (down).	133
A.6	Topology of networks pdh (left) and di-yuan (right).	134
A.7	Topology of networks polska (left) and nobel-germany (right).	134
A.8	Topology of network nobel-us.	134