WARSAW UNIVERSITY OF TECHNOLOGY

DISCIPLINE OF SCIENCE INFORMATION AND
COMMUNICATION TECHNOLOGY
FIELD OF SCIENCE ENGINEERING AND TECHNOLOGY

Ph.D. Thesis

Ilya Kalesnikau, M.Sc.

**Optimization of communication networks
with variable capacity of links**

Supervisor
Artur Tomaszewski, Ph.D., D.Sc.

WARSAW 2022

Acknowledgement

I would like to express my gratitude to my supervisor, Artur Tomaszewski, for constant advice, support, and valuable feedback while writing this thesis.

Next, I am thankful to Dritan Nace, Michael Poss and Jacek Rak for giving me an opportunity to work with them.

I would also like to thank Marinela Shehaj for fruitful research collaboration.

The work on this thesis was financed by the National Science Centre, Poland, under grant no. 2017/25/B/ST7/02313: "Packet routing and transmission scheduling optimization in multi-hop wireless networks with multicast traffic". Besides, parts of this thesis are based on earlier research conducted under the National Science Centre grant no. 2015/17/B/ST7/03910: "LTCC – Logical Tunnel Capacity Control – a traffic routing and protection strategy for communication networks with variable link capacity".

Finally, I would like to recognize the guidance of Michał Pióro, the leader of the aforementioned grants, from whom I learned a lot about optimization of communications networks.

Abstract

Optimization of communication networks with variable capacity of links

Logical Tunnel Capacity Control is a traffic routing and protection strategy designed for communications networks characterized by frequent link capacity fluctuations. The key component of the LTCC strategy is Flow Thinning – a novel method of controlling the size of flows assigned to network tunnels in reaction to available link capacity changes.

The thesis presents Mixed-Integer Linear Programming problem models and optimization solution algorithms of designing the network that uses the LTCC strategy, and elaborates their effectiveness and efficiency. It constitutes a comprehensive study of network design problem models and solutions corresponding to the variants of the FT mechanism and the variants of the network links availability description. To deliver computationally efficient network design methods, the problem models and solution algorithms are based on the path generation and state generation approaches. The thesis elaborates the resulting network design problem decomposition and the subproblems' formulations pertaining to the variants of the FT formula. The effectiveness of the derived formulations is elaborated in an extensive numerical study, where real traffic and network data is used to examine the efficiency of the models and algorithms as well as the efficiency of the network solutions.

The thesis proves that the LTCC strategy is an efficient network solution of routing and protecting traffic. It shows that the variants of the network design problem that result from the variants of the FT mechanism can be effectively modeled and solved within a single conceptual framework. The thesis demonstrates that due to multiple variants of the FT formula the strategy is a flexible approach to designing and operating networks with varying link capacities, offering a capability to balance implementation feasibility, routing robustness, network cost, and optimization time.

Keywords: resilient communications networks, multicommodity flow networks, flow thinning, variable link capacity, linear and mixed-integer programming, robust optimization, wireless mesh networks, free space optics

Streszczenie

Optymalizacja sieci telekomunikacyjnych ze zmienną przepustowością łączy

Logical Tunnel Capacity Control jest strategią kierowania i zabezpieczania ruchu przeznaczoną dla sieci teleinformatycznych charakteryzujących się częstymi zmianami przepustowości łączy. Kluczowym elementem strategii LTCC jest oryginalny mechanizm Flow Thinning pocieniania przepływów sterujący wielkością strumieniu ruchu w odpowiedzi na zmieniający się stan łączy.

Rozprawa przedstawia modele problemów programowania liniowego całkowitoliczbowego i algorytmy optymalizacyjne służące do projektowania sieci wykorzystujących strategię LTCC oraz analizuje ich skuteczność i wydajność. Stanowi wszechstronne studium modeli i rozwiązań problemu projektowania odpowiadających zdefiniowanym wariantom mechanizmu FT oraz wariantom sposobu opisu stanów dostępności łączy. Aby zapewnić efektywność obliczeniową proponowanych metod projektowania sieci, rozważane modele i algorytmy wykorzystują podejście oparte na generacji ścieżek i na generacji stanów. Rozprawa szczegółowo analizuje wynikającą z takiego podejścia dekompozycję problemu projektowania oraz szczegółowe sformułowania podproblemów, odpowiadające poszczególnym wariantom mechanizmu FT. Szerokie badania efektywności i wydajności uzyskanych sformułowań przy użyciu rzeczywistych danych o ruchu i sieci, pozwalają ocenić zarówno wydajność proponowanych modeli i algorytmów, jak również uzyskiwanych rozwiązań sieciowych.

Przedstawione wyniki dowodzą, że strategia LTCC jest efektywnym rozwiązaniem kierowania i zabezpieczania ruchu sieciowego, a warianty problemu projektowania sieci związane z poszczególnymi odmianami mechanizmu FT można skutecznie modelować i rozwiązywać korzystając z jednolitego szkieletu pojęciowego. Rozprawa udowadnia, że dzięki różnorodności wariantów mechanizmu FT strategia LTCC jest elastycznym podejściem do problemu projektowania i zarządzania sieciami teleinformatycznymi ze zmienną przepustowością łączy, dając możliwość poszukiwania kompromisu pomiędzy odpornością kierowania, kosztem sieci, łatwością wdrożenia i czasem optymalizacji.

Słowa kluczowe: odporne sieci teleinformatyczne, sieci przepływów wielotowarowych, pocienianie przepływów, zmienna przepustowość łączy, programowanie liniowe i całkowitoliczbowe, optymalizacja odporna, bezprzewodowe sieci kratowe

Contents

List of Figures

List of Tables

List of Formulations and Algorithms

List of Abbreviations

AFT	Affine Flow Thinning
DFT	Dual to Flow Thinning
DL	Double Link degradation scenario
DWDM	Dense Wavelength Division Multiplexing
DWSP	Demand-Wise Shared Protection
ER	Elastic Routing
FSO	Free Space Optics
FT	Flow Thinning
GMPLS	Generalized Multi-Protocol Label Switching
GR	Global Rerouting
IP	Internet Protocol
LP	Linear Programming
LR	Link Restoration
LTCC	Logical Tunnel Capacity Control
MAN	Metropolitan Area Network
MCS	Modulation and Coding Scheme
MILP	Mixed-Integer Linear Programming

MIP	Mixed-Integer Programming
MIQP	Mixed-Integer Quadratic Programming
MP	Master Problem
MPLS	Multi-Protocol Label Switching
PDP	Path Diversity Protection
PG	Path Generation
PGA	Path Generation Algorithm
PMAN	Paris Metropolitan Area Network
PP	Pricing Problem
SDN	Software Defined Network
SGA	State Generation Algorithm
SL	Single Link degradation scenario
SNDlib	Survivable fixed telecommunication Network Design library
SRLG	Shared-Risk Link Group
TL	Triple Link degradation scenario
UR	Unrestricted Reconfiguration
WMN	Wireless Mesh Networks

List of Symbols

$\alpha(e, s)$	availability coefficient of link e in state s
$\beta(e, s)$	degradation coefficient of link e in state s
$\delta(e, d, p)$	link-path incidence binary coefficient indicating whether link e is traversing path p of demand d
$\delta(v)$	set of links incident to node v
$\hat{\mathcal{P}}$	set of all elementary (simple) paths of all demands $d \in \mathcal{D}$
$\hat{\mathcal{P}}(d)$	set of all elementary (simple) paths of demand d
$\hat{\mathcal{R}}(d, e)$	set of all elementary (simple) paths of demand d that traverse link e
$\mathbb{R}$	set of real numbers
$\mathbb{R}_+$	set of nonnegative real numbers
$\mathcal{A}$	set of arcs
$\mathcal{D}$	set of demands
$\mathcal{E}$	set of links
$\mathcal{E}(d, p)$	set of links traversing path p of demand d
$\mathcal{E}(s)$	set of links not fully operational in state s
$\mathcal{G}$	graph
$\mathcal{P}(d)$	set of admissible paths of demand d
$\mathcal{R}(d, e)$	set of paths of demand d that traverse link e

$\mathcal{S}$	set of states
$\mathcal{S}(e)$	set of states with not fully operational link e
$\mathcal{V}$	set of nodes
$\xi(e)$	unit capacity cost of link e
$c(e)$	capacity of link e (parameter)
$h(d)$	nominal traffic volume of demand d
$h(d,s)$	traffic volume of demand d to be realized in state s
$o(d)$	originating node of demand d
$t(d)$	terminating node of demand d
x^0_{dp}	nominal flow of demand d allocated on path p
x^s_{dp}	flow of demand d allocated on path p in state s
$X^{\|2\|}$	set of all 2-element subsets of set X
y_e	capacity of link e (variable)

Chapter 1

Introduction

1.1 Flow thinning

There exists a class of communications transport networks that are characterized by link capacity variations. A primary example are wireless networks, where fluctuations of the signal-to-noise ratio, which reflect varying radio signal propagation conditions, cause temporary changes of the modulation and coding scheme used on the link and thus of the link's bit rate. Since link capacity changes may lead to network overload and result in traffic losses, appropriate network management and control mechanisms are necessary that would protect traffic flows against link capacity variations and guarantee the required quality of service.

Logical Tunnel Capacity Control (LTCC) is a strategy of protecting traffic flows against overloads that are caused by link capacity variations proposed in [58]. A method of routing and protecting traffic, it assumes that each aggregated end-to-end traffic flow is load balanced across the set of dedicated end-to-end logical tunnels whereas the amount of traffic directed to each tunnel is an affine combination of link capacities. The characteristic feature of the method is that the set and the routing of logical tunnels is static, only the size of the flow assigned to each tunnel changes. Moreover, the computational complexity of the load balancing function is extremely low. Due to the use of the affine function, the decrease of link capacities usually results in decreasing the size of the flows on selected tunnels. Hence, this traffic control mechanism is referred to as Flow Thinning (FT), which is also a common name of the traffic routing and protection method. Finally, it is important that the traffic control mechanism is applied independently to each end-to-end traffic flow, and is executed locally at the source node of the flow. Combined together, those features offer potential implementation simplicity and operations efficiency of the LTCC strategy.

Practical deployment of LTCC requires as a minimum developing appropriate network design methods, defining modifications and extensions of the existing network control architectures and protocols, and evaluating the performance of the strategy, in particular, in order to examine whether it matches the performance of more complex traffic protection schemes, e.g., those that are based on rerouting of established tunnels. The aim of this thesis is to develop efficient methods of network design, evaluate their computational performance, and examine the effectiveness of the LTCC strategy.

1.2 Potential applications

The network design problem considered in this thesis is relevant to any communication network that makes use of logical tunnels and treats varying link capacity as typical, potentially being capable of adjusting logical tunnel capacity in reaction to link capacity changes. The most interesting and promising field of application of the considered network design models are Wireless Mesh Networks (WMN) utilizing Free Space Optics (FSO) transmission and Multiprotocol Label Switching (MPLS) tunneling technologies.

WMNs are packet-switched networks composed of Internet gateways and fixed IP routers interconnected with links that are realized by means of wireless technologies, such as FSO, microwave or radio. WMN is undoubtedly a promising solution to provide broadband Internet access for fixed and mobile users in short time in the areas lacking network infrastructure or where deployment of the wired infrastructure is hardly feasible or very costly. This is due to the fact that in many cases WMNs are cheaper and simpler to plan, deploy, maintain and operate as compared to wired networks [35]. Another example of WMN application is a corporate network deployed in a large city using FSO links installed on the roofs of buildings.

FSO is a wireless optical transmission technology, assuming the transmission link is realized by means of the optical laser beam sent between a pair of transceivers that are placed within the line-of-sight of each other. The key advantages of the FSO technology in comparison to the optical fiber communication are license free operation, ease of deployment, high data rates, and increased security [22]. However, at the same time the FSO transmission system is vulnerable to weather disruptions (such as heavy rainfall, fog, haze) that lead to substantial degradation of transmission quality. Fortunately, the FSO link is capable of adapting to constantly varying channel propagation conditions appropriately adjusting the Modulation and Coding Scheme (MCS) of the transmission [20] to match the current channel state. Still, even if not complete unavailability of the transmission link, the result is link capacity decrease. Therefore, one of the major issues in the design of FSO-based networks is to assure network resilience defined as the ability of the network to guarantee an acceptable level of service in the face of faults and challenges to its normal operation [62].

WMNs with links based on the FSO technology is the main application of the network design models considered in this thesis. However, IP/MPLS networks using Dense Wavelength Division Multiplexing (DWDM) transmission technology also can make use of the considered models: since the IP link might be a bundle composed of a number of network connections of the DWDM optical network layer, when a failure occurs in the DWDM layer, the link is likely to lose only a part of its capacity [56].

It is worth noticing, that there is also a somehow specific, though technology-agnostic, situation when the models can even be applied to networks with constant capacities of links: if a network realizes connections with (at least) two priorities, from the perspective of the low-priority connections the available capacity of links varies as it depends on the configuration of high-priority connections; this scenario might apply, in particular, to DWDM networks that use lambda-switching [75], which, as it is worth noting, is not a packet-switching- but circuit-switching-based technology.

1.3 Problem description

The development of the LTCC strategy is a scientific problem that consists of the following subproblems.

- Elaboration of formal mathematical models and effective solution algorithms for link dimensions, path routing and flow protection design in communications networks with variable link capacity and FT traffic control mechanism.

- Elaboration of functional models of the data, control and management planes of the network, covering architectural, functional, and protocol extensions required by the implementation of the FT traffic control mechanism.
- Verification of both the mathematical and the functional models in a proof of concept simulation framework.

The core of this thesis is dedicated to the first subproblem, whereas the latter two subproblems are discussed only briefly.

Communications network design [16, 30], and more generally, communications network planning, can be considered as one of the areas of the operations research, the others being, e.g., production planning [41], inventory management [68], logistics design [61], to name a few. Thus, to solve the network design problems mathematical programming methods are used. The problems are regarded as optimization problems that are formally described as optimization problem models and solved with appropriate mathematical programming algorithms.

The network design problems considered in this thesis are best described and formalized using the theory of multicommodity flow networks [1]. The network is modelled by means of the graph whose vertices and edges correspond to network nodes and network links, respectively. The network traffic is modelled as a set of demands, with each aggregated end-to-end traffic flow corresponding to a separate commodity, and the demand volume being equal to the size of the aggregated flow. Following the principles of the FT method, the demands are realized as network flows in a bifurcated way – each demand uses a set of paths. The paths are assigned path flows so that the total size of those path flows is equal to the demand's volume.

Due to the link capacity variations the problem is best expressed using a multi-state variant of the multicommodity flow network model [72]. The set of network states defines a set of network configurations. Each state corresponds to a combination of link capacities. The state that every link has its maximum capacity is regarded as the nominal state, while in each non-nominal state the available capacity of any link can be lower than its nominal capacity, the fraction being defined by the link availability coefficients. Hence, the states are referred to as link availability states. Since, due to the nature of link capacity variations, in a typical state only some links are fully available while on the other links only a fraction of the nominal capacity can be used, we consider what we call a multiple partial link availability state model of the multicommodity flow network: in each state multiple links may have only part of its nominal capacity available.

While for every demand in each non-nominal state only the paths can be used that are used in the nominal state, the sizes of the path flows are state-dependent – in general, they can be different for each state. Due to the nature of the FT traffic control mechanism, the size of the path flow in the non-nominal state must not be larger than the size of that flow in the nominal state and it must be a function of the link capacities. While the total size of the demand's path flows must still be equal to the demand's volume, in the most severe states, when substantial network capacity has been lost, it is allowed to realize demands partially, decreasing the demand volumes while still providing acceptable level of service to the aggregated traffic flows.

Given the unit costs of link capacity, the set of link availability states, the values of link availability coefficients, and the demand volumes for each state, the network design problem consists in defining the set of paths for every demand, and the size of each path flow for every state.

The problem of finding either a feasible, or, e.g., a minimum-cost solution, of the network design problem can be formulated either as a Linear Programming (LP) problem model, especially if the path flow size is a linear function of link capacities, like in the case of the path flow size being an affine combination of link capacities, or more, generally, as a Mixed-Integer Linear Programming (MILP) [40] problem model. Due to inherent nature of the traffic control mechanism, in particular the inter-dependance of the path flows in the nominal and the non-nominal state, the link-path formulation of the multicommodity flow problem must be used. Therefore, the resulting formulation is non-compact as the number of network paths is exponential. In general, the number of network states is also exponential, since the multiple partial link availability model is assumed.

Thus, it is virtually impossible to optimally solve the optimization models directly already for medium-sized networks: the number of paths and the number of states growing exponentially with the size of the network make the models' sizes beyond comprehension. This thesis considers effective solution algorithms for the models, capable of dealing with exponential growth of the two sets.

The optimization algorithms considered in this thesis must be based both on well-known state-of-art optimization methods and on rather specific techniques. The basic approach that enables solving the problem must be path generation, which is a common application of the column generation technique [23]. The approach requires defining the master problem and appropriate pricing problems that can be solved efficiently. Those problems are rather specific and become complex and difficult to solve due to the form of the FT function. The pricing problems must be developed using the dual theory [31].

Additionally, to make both the master problem and the pricing problems solvable efficiently the formulations must be strengthened by defining effective problem cuts. In order to limit the size of problem formulations and to solve the problems efficiently, row generation and cutting-plane generation techniques must be used [7] As a result, the considered problem solution algorithms must be based on the branch-and-price-and-cut approach [15, 2].

Last but not least, to deal with a potentially huge number of network states, it might be beneficial to consider network states indirectly, using more sophisticated and more specific problem modelling and optimization techniques of affine decision rules [5, 47, 60] and uncertainty sets [8, 3].

1.4 Related work

The thesis considers optimization problems that belong to the area of resilient multicommodity flow networks. Significant amount of research was done in this area in recent years (see [76, 56, 21]). Additionally, some research focused on auxiliary aspects of routing and protecting network traffic such as traffic uncertainty [10, 78, 71, 9], fair network optimization [32, 37, 42, 48], and multicast transmission [59, 45, 46]. This thesis considers optimization problems that arise when combining two specific aspects of the traffic routing and protection design being the link capacity model and the traffic control mechanism of the FT method.

The link capacity model considered in this thesis assumes multiple partial link availability or, equivalently, multiple partial link failures. The case of single total link failures has been studied in [13, 11, 19, 74]. This case can also be found in [56] under the name Link Restoration (LR). The case of multiple link failures, though limited to total failures, is known in the literature as Shared-Risk Link Group (SRLG) and has been studied, in particular, in [69, 43]. One of a very few proposed traffic protection mechanisms that assumes multiple partial link availability is Unrestricted Reconfiguration (UR) (see [56]). UR, also known in the literature as Global Rerouting (GR), admits unlimited reconfiguration of flows in case of a network state change. In other words, with UR all flows are first disconnected and then reestablished from scratch in the surviving link capacity. Exactly for that reason, UR is hardly applicable in real networks as it is operationally complex, time-consuming, and may lead to large traffic losses.

In turn, the traffic control mechanism considered in this thesis is novel. FT was first proposed in the conference papers [58] and [57], and later described in the journal articles [50] and [17]. FT is a traffic routing and protection method designed specifically to take into account multiple partial link availability: the size of the path flow is adjusted dynamically in reaction to fluctuations of available link capacity, the adjustment consisting in appropriate thinning of the flow compared to its nominal size. FT is related to and inspired by the following traffic protection and routing mechanisms: Elastic Routing (ER) and Demand-Wise Shared Protection (DWSP). ER (see [67, 18]) assumes multiple total link failures. ER enables increasing to a certain extent traffic flows on still available paths of the demands affected by the failure at the cost of decreasing path flows of the unaffected demands. DWSP (see [29, 77], and [73, 43] under the name Path Diversity Protection – PDP) also assumes multiple total link failures. DWSP disconnects path flows affected by the failure, and guaranties that the unaffected path flows are sufficient to realize (possibly decreased) demand traffic volumes.

1.5 Thesis contribution

The preliminary results contained in the original publications [58, 57, 50, 17] showed that FT (and especially its so called affine version, AFT) is a promising traffic protection and routing method. However, when first proposed, the FT concept was at a preliminary stage of development, lacking thorough elaboration of optimization models and solution algorithms, and missing network-technology-related deployment solutions; further development was thus required in order to make FT applicable in real networks.

Most of the work on the FT method was then carried out within the research project "LTCC – Logical Tunnel Capacity Control – a traffic routing and protection strategy for communication networks with variable link capacity" granted by the National Science

Centre of Poland – the grant no. 2015/17/B/ST7/03910. The results of that project have been presented in the following publications.

- Conference publication [52] and its extended journal version [53] contain an optimization model for affine FT together with an effective solution algorithm for the model based on the so called path generation and pricing problem.
- Conference publication [51] and its extended journal version [55] contain path generation algorithm and pricing problem for other potentially implementable variants of FT, the so called general FT and quadratic FT.
- Conference publication [54] and its extended journal version [27] add robustness to the optimization models of FT, that is achieved by characterizing the link availability states by means of the so called state polytope instead of the explicit list of preselected states. Furthermore, these papers contain effective solution algorithms for the models based on the combination of path generation algorithm and the so called cutting-plane algorithm.
- Journal publication [28] extends optimization models of FT by the so called max-min fairness concept. Moreover, it contains a method for estimating availability of FSO link capacity in adverse weather conditions.
- Book chapter [49] discusses optimization models of FT along with optimization models designed specifically for networks in which links interfere with each other.
- Technical reports [36] and [26] focus on FT implementation options and the FT simulation framework, respectively.

Additionally, though not directly related to the LTCC project and primarily not focussed on the FT method, the following publications present results that played an important role in the FT development.

- Conference and journal publication [25] contains a comparison of two pricing problems for path generation related to UR optimization model.
- Conference publications [14, 65, 64] and journal publications [38, 66] consider resilient optimization models of UR based on the so called uncertainty sets and Benders decomposition. Apart from the optimization models, these articles concentrated on development of a method of translating historical weather records into

appropriate modulation and coding scheme data, which in turn determines FSO link availability. Moreover, both the optimization models and the translation method were tested on a realistic FSO network instance, created especially for the purpose of the research on the basis of the real data of Paris metropolitan area. PhD thesis [63] summarizes these publications.

The author of this thesis is a co-author of the majority of the above publications, with the exception of [36, 14, 63]. The author's work on the LTCC strategy, and the FT method in particular, concentrated on the algorithmic aspects of the developed optimization models and on their computational efficiency. While each of the publications targeted a specific aspect of the LTCC strategy, this thesis constitutes a comprehensive study of the FT optimization problem models, problem solution algorithms, and their efficiency, and focusses on the author's contribution to the development of the LTCC strategy and the FT method. In particular, the thesis does not consider deployment models of the LTCC strategy, which were elaborated in detail in [36], where both centralized, based on the Software Defined Network (SDN) concept and technology, and distributed, based on the Generalized Multi-Protocol Label Switching (GMPLS) architecture of the control plane, architectural solutions of LTCC deployment were defined.

The thesis consists of five chapters. After the description of the problem, the scope of the research that has been carried out, and the contribution of this thesis in Chapter 1, Chapter 2 presents optimization models of designing the network that operates according to the FT principle, defining a general form of the model, and developing its specific variants that result from different forms of the FT function and different link availability description approaches. Chapter 3 develops solution methods of solving the optimization problem, introducing a generic algorithm based on path generation and state generation, the formulation of the master problem and the set of specific formulations of the pricing problem corresponding to the variants of the FT function. Chapter 4 describes a series of numerical experiments aimed at evaluating the computational effectiveness of the solution algorithms and at comparing the traffic efficiency of the considered variants of the FT mechanism. Finally, Chapter 5 concludes the thesis.

Chapter 2

Optimization problems

In this chapter we formally define network design problems. We express the design problems as MILP optimization problem models, for each problem defining the parameters of the problem, its optimization variables and constraints, and the objective function. We start, however, by defining the formal model of the network, which is shared by all optimization models. We note that while defining network and problem models we consistently use the convention that while referring to the element of the parameter vector we put the index in brackets, and while referring to the element of the variable vector we put it either in the subscript or in the superscript.

2.1 Network model

We model the considered network with graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of links. We assume that, unless stated otherwise, graph $\mathcal{G}$ is undirected, although most of the following considerations apply to directed graphs as well. We also assume that $\mathcal{G}$ is simple (has no multiple edges); then each link $e \in \mathcal{E}$ corresponds to an undirected pair of nodes: $e = \{v, w\}$, where $v, w \in \mathcal{V}$ and $v \neq w$. For each $v \in \mathcal{V}$, $\delta(v) \subseteq \mathcal{E}$ denotes the set of links incident to node v. Vector $\xi = (\xi(e), e \in \mathcal{E})$ denotes unit capacity costs of the links, i.e., $\xi(e)$ is the cost of realizing one demand unit on link e. Nominal link capacities are represented by vector $c = (c(e), e \in \mathcal{E})$ when the capacities of links are given (they are parameters of the optimization model), or by vector $y = (y_e, e \in \mathcal{E})$ when they are optimization variables.

We model network traffic with set $\mathcal{D}$ of (traffic) demands: each demand models an end-to-end aggregated traffic flow entering the network. We assume that traffic demands are directed, thus for each $d \in \mathcal{D}$, let $o(d)$ and $t(d)$ be, respectively, the originating (ingress) and the terminating (outgress) node of demand d. Vector $h = (h(d), d \in \mathcal{D})$ denotes nominal volumes of the demands.

For each $d \in \mathcal{D}$, $\hat{\mathcal{P}}(d)$ is a set of all simple paths connecting end nodes $o(d)$ and $t(d)$ of demand d, and $\mathcal{P}(d) \subseteq \hat{\mathcal{P}}(d)$ is the set of paths that can actually be used by the logical tunnels of demand d. We also define set $\hat{\mathcal{P}} := \bigcup_{d \in \mathcal{D}} \hat{\mathcal{P}}(d)$ of all network paths and set $\mathcal{P} := \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$ of all admissible network paths. For all $d \in \mathcal{D}$, $p \in \hat{\mathcal{P}}(d)$, $\mathcal{E}(d,p) \subseteq \mathcal{E}$ denotes the set of links of path p of demand d. Alternatively, we also express the relationship between links and paths with link-path incidence coefficients: for all $e \in \mathcal{E}$, $d \in \mathcal{D}$, $p \in \hat{\mathcal{P}}(d)$, $\delta(e,d,p)$ equals 1 if, and only if, $e \in \mathcal{E}(d,p)$, and 0 otherwise. For all $d \in \mathcal{D}$, $e \in \mathcal{E}$, $\mathcal{R}(d,e) \subseteq \mathcal{P}(d)$ and $\hat{\mathcal{R}}(d,e) \subseteq \hat{\mathcal{P}}(d)$ denote the sets of paths of demand d that traverse link e: $\mathcal{R}(d,e) := \{p \in \mathcal{P}(d) : e \in \mathcal{E}(d,p)\}$ and $\hat{\mathcal{R}}(d,e) := \{p \in \hat{\mathcal{P}}(d) : e \in \mathcal{E}(d,p)\}$. Finally, for all $d \in \mathcal{D}$, $p \in \hat{\mathcal{P}}(d)$, $\mathcal{V}(d,p) \subseteq \mathcal{V}$ denotes the set of nodes of path p of demand d.

Paths correspond to logical tunnels that carry traffic, and thus we model traffic flows carried by tunnels as path flows. The demand is realized by a set of path flows. For all $d \in \mathcal{D}$, $p \in \mathcal{P}(d)$, variable x^0_{dp} denotes the nominal size of the path flow of demand d on path p.

We model network configurations with set $\mathcal{S}$ of network states: at any instant of time the network finds itself in some state $s \in \mathcal{S}$. Each state corresponds to a combination of link capacities. Since the actual capacity of the link is some available portion of the link's total capacity $c(e)$, states are also referred to as link availability states. For each $s \in \mathcal{S}$, the capacities of the links in state s we characterize with the vector of link availability coefficients $\alpha(s) = (\alpha(e,s), e \in \mathcal{E})$, $\alpha(e,s) \in [0,1]$, which define the fraction of the nominal link capacity that is available in state s: for all $e \in \mathcal{E}$, $s \in \mathcal{S}$, the capacity of link e available in state s is defined as $y^s_e = \alpha(e,s)y^0_e$ and $c(e,s) = \alpha(e,s)c(e)$. We also use the vector of link degradation coefficients $\beta(s) = (\beta(e,s), e \in \mathcal{E})$, $\beta(e,s) \in [0,1]$, where $\beta(e,s) = 1 - \alpha(e,s)$ for all $e \in \mathcal{E}$, $s \in \mathcal{S}$.

The state s with all links fully operational (i.e, $\alpha(e,s) = 1$, for all $e \in \mathcal{E}$, $s \in \mathcal{S}$) is called the nominal state. For each $s \in \mathcal{S}$, the set of links that are not fully operational in state s is defined as $\mathcal{E}(s) := \{e \in \mathcal{E} : \alpha(e,s) < 1\}$. In a similar way, for each $e \in \mathcal{E}$, the set of states in which link e is not fully operational is defined as $\mathcal{S}(e) := \{s \in \mathcal{S} : \alpha(e,s) < 1\}$. Certainly, for a nominal state $\mathcal{E}(s) = \emptyset$.

In any non-nominal state $s \in \mathcal{S}$ the available link capacities y^s are decreased in comparison with the nominal link capacities y^0, and applying the nominal flows x^0_{dp} in that state might cause link overloads and lead to traffic losses. Therefore, the path flows, in general, have to be changed in those states. For all $s \in \mathcal{S}$, $d \in \mathcal{D}$, $p \in \mathcal{P}(d)$, variable x^s_{dp} denotes the size of the path flow of demand d on path p in state s. Since in some severe states the network can loose a significant part of its total link capacity, in such states it is reasonable to accept a partial realization of demands. Thus, for all $d \in \mathcal{D}$, $s \in \mathcal{S}$, $h(d,s)$ is the volume of demand d that must be realized in state s; $h(d,s) \leq h(d)$.

Note that link capacities, demand volumes and path flows are expressed in the same units (e.g., Gbps).

2.2 Optimization models

In this section we define MILP optimization problem models, starting from the benchmark model followed by the models related to the LTCC strategy.

2.2.1 Unrestricted Reconfiguration

Unrestricted Reconfiguration (UR) is a traffic protection and routing mechanism applicable to communication networks operating under the assumption of multiple partial link availability. With UR, in a given state the traffic demands are restored from scratch in the available capacity. The optimization problem corresponding to UR can be represented as the following LP problem model using the link-path formulation and state-dependent path flow variables x^s_{dp}.

Problem UR($\mathcal{P}, \mathcal{S}$):

$$F = \min \textstyle\sum_{e \in \mathcal{E}} \xi(e) y_e \tag{2.1a}$$

$$\textstyle\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}(d)} \delta(e,d,p) x^s_{dp} \leq \alpha(e,s) y_e, \quad e \in \mathcal{E}, s \in \mathcal{S} \tag{2.1b}$$

$$\textstyle\sum_{p \in \mathcal{P}(d)} x^s_{dp} \geq h(d,s), \quad d \in \mathcal{D}, s \in \mathcal{S} \tag{2.1c}$$

$$y_e \in \mathbb{R}_+, e \in \mathcal{E};\ x^s_{dp} \in \mathbb{R}_+, d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S}. \tag{2.1d}$$

Objective (2.1a) minimizes the total network cost expressed as the sum of link capacities weighted by their unit capacity costs. Constraint (2.1b) ensures that for all $e \in \mathcal{E}$, $s \in \mathcal{S}$, the available link capacity is not exceeded by the corresponding link loads. Constraint (2.1c) ensures that for all $d \in \mathcal{D}$, $s \in \mathcal{S}$, path flows are sufficient to realize the reduced demand volumes.

Note that in order to solve (2.1) optimally, all elementary paths $\hat{\mathcal{P}}$ should be taken into account. As the number of elementary paths grows exponentially with the graph size, the formulation (2.1) becomes non-compact due to the exponential number of path flow variables x^s_{dp}. Still, UR can be formulated as a compact LP in node-link notion with link flow variables x^s_{ed} instead of path flow variables. Alternatively, it is possible to indirectly consider all elementary paths in (2.1) by using the so called column generation technique (see Section 3.1).

Due to the fact that UR applies no restrictions on flow rearrangement during the change of the network state, the mechanism becomes hardly usable in the networking practice. Yet, UR gives the lower-bound on the network cost and serves well as a computationally efficient benchmark mechanism for comparison while evaluating more specific network protection mechanisms.

2.2.2 Flow Thinning

In its general form, FT is a traffic protection and routing mechanism similar to the UR mechanism, but with additional restrictions on flow rearrangement: state-dependent path flows are thinned with respect to their corresponding nominal path flows. The following LP formulation represents the general optimization problem for FT; it is a general formulation of the problem as it does not specify the FT formula.

Problem FT($\mathcal{P}, \mathcal{S}$):

$$F = \min \textstyle\sum_{e\in\mathcal{E}} \xi(e) y^0_e \tag{2.2a}$$

$$\textstyle\sum_{d\in\mathcal{D}} \sum_{p\in\mathcal{P}(d)} \delta(e,d,p) x^0_{dp} \le y^0_e, \quad e \in \mathcal{E} \tag{2.2b}$$

$$\textstyle\sum_{p\in\mathcal{P}(d)} x^s_{dp} \ge h(d,s), \quad d \in \mathcal{D}, s \in \mathcal{S} \tag{2.2c}$$

$$\textstyle\sum_{d\in\mathcal{D}} \sum_{p\in\mathcal{P}(d)} \delta(e,d,p) x^s_{dp} \le \alpha(e,s) y^0_e, \quad e \in \mathcal{E}, s \in \mathcal{S}(e) \tag{2.2d}$$

$$x^s_{dp} \le x^0_{dp}, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S} \tag{2.2e}$$

$$y^0_e \in \mathbb{R}_+, e \in \mathcal{E};\ x^0_{dp}, x^s_{dp} \in \mathbb{R}_+, d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S}. \tag{2.2f}$$

The formulation (2.2) (unlike the formulation (2.1)) distinguishes the nominal state among all other states and makes use of nominal path flows variables x^0_{dp}. Constraint (2.2b) ensures that nominal flows do not cause overflowing of the nominal link capacity. Constraint (2.2e) ensures thinning: i.e., no state-dependent path flow x^s_{dp} exceeds its corresponding nominal path flows x^0_{dp}. The remaining constraints (2.2c) and (2.2d) are similar to constraints (2.1c) and (2.1b) in the UR formulation, respectively. The only difference in these constraints is that (2.2d) is written down not for all link-state pairs (i.e., $e \in \mathcal{E}, s \in \mathcal{S}$), but only for such link-state pairs in which the considered link is not fully available. The reason is that for any state $s \in \mathcal{S} \setminus \mathcal{S}(e)$ constraint (2.2d) is auxiliary as it is implied by constraints (2.2b) and (2.2e).

Introducing the thinning constraint in FT significantly reduces the number of flows that are adjusted in a transient period, i.e., when the network is moving from some state s_1 to some state s_2, as compared to UR. With FT, the number of adjusted flows is upper-bounded by the number of flows traversing links that are not fully available in either s_1 or s_2, while with UR virtually all flows can be changed.

As flows are not adjusted instantly, some flows can be increased before other flows are decreased, causing potential link overloads. Thus, small number of adjusted flows achieved with FT decreases the chance of link overloads (traffic losses) during transient period. The disadvantage of FT as compared with UR is following: in order to realize the same traffic FT requires in general at least as much total link capacity as UR, and hence network cost F is in general higher than for UR.

2.2.3 Flow thinning formula

The practical usefulness of the general FT presented in Section 2.2.2 is limited, mainly because of potential difficulties with the real-time broadcasting of the current link capacity information and with defining path flow values in the states that are not considered during optimization. The reason is as follows. Suppose that a network is in operational mode, and it uses path flows x^s_{dp} obtained from solving the FT formulation (2.2). In order to properly use these flows, the current network state s should be known to all nodes that are originating nodes of at least one demand path (thus, practically to all network nodes). As state s is defined by the vector of availability coefficients of all links $\alpha(s)$, the vector should be known to all nodes. Otherwise, any differences between the actual network state and the states seen from the originating nodes' perspective could lead to traffic losses. Such differences are inevitable, as it is problematic to disseminate the current availability coefficients of *all* links to *all* originating nodes in real-time. Moreover, when the network finds itself in a state not considered during optimization, no path flows are defined for such a state, which is another source of potential traffic losses.

Using the so called *flow thinning formula* is a way to overcome the aforementioned difficulties. Flow thinning formula is based on the notion of decision rules [5]. With the FT formula, path flows x^s_{dp} do not directly depend on specific state $s \in \mathcal{S}$. Instead, they are defined for all $d \in \mathcal{D}$, $p \in \mathcal{P}(d)$ independently through function F_{dp}, arguments of which, and they are link degradation coefficients $\beta(s)$, describe the state:

$$x^s_{dp} = F_{dp}(\beta(s)). \tag{2.3}$$

Note that formula (2.3) could be defined through the vector of link availability coefficients $\alpha(s)$ as well, still, link degradation coefficients are used here to make the forthcoming formulae and formulations more concise. For given $d \in \mathcal{D}$, $p \in \mathcal{P}(d)$, $s \in \mathcal{S}$, the following quadratic form of the link degradation coefficients is the most general form of the flow thinning formula considered in this thesis:

$$x^s_{dp} = z^0_{dp} - \textstyle\sum_{e \in \widetilde{\mathcal{E}}(d,p)} \beta(e,s) z^e_{dp} + \textstyle\sum_{\{e,e'\} \in \widetilde{\mathcal{E}}^{|2|}(d,p)} \beta(e,s)\beta(e',s) z^{ee'}_{dp}. \tag{2.4}$$

In (2.4), the predefined set $\widetilde{\mathcal{E}}(d,p) \subseteq \mathcal{E}$ (called the range of the flow thinning formula) consists of the links whose degradation coefficients influence the flow on path $p \in \mathcal{P}(d)$ of demand $d \in \mathcal{D}$, and $\widetilde{\mathcal{E}}^{|2|}(d,p)$ is the set of all 2-element subsets of $\widetilde{\mathcal{E}}(d,p)$. For given $d \in \mathcal{D}$, $p \in \mathcal{P}(d)$, vector of variables

$$z_{dp} := (z^0_{dp};\ z^e_{dp}, e \in \widetilde{\mathcal{E}}(d,p);\ z^{ee'}_{dp}, \{e,e'\} \in \widetilde{\mathcal{E}}^{|2|}(d,p)) \tag{2.5}$$

expresses in what extent a certain degradation coefficient $\beta(e,s)$ of link $e \in \mathcal{E}$ in state $s \in \mathcal{S}$ influences state-dependent path flow x^s_{dp}. Variables in z_{dp} are measured in the same units as x^s_{dp}. Note that when the flow thinning formula (2.4) is applied in LP, path flow variables x^s_{dp} become auxiliary because they are defined by other variables.

2.2.4 Variants of flow thinning formula

The formula (2.4) is the most general, and several special cases of it are considered in this thesis. These cases (also called variants) differ by three aspects: *form*, *range*, and *structure*.

2.2.4.1 Form

Two types of the *form of the flow thinning formula* are considered:

- quadratic Q: (2.4);
- affine A: $x^s_{dp} = z^0_{dp} - \sum_{e \in \widetilde{\mathcal{E}}(d,p)} \beta(e,s) z^e_{dp}$.

Clearly, the affine formula is a special case of the quadratic formula. Hence, the quadratic form is in general more effective in terms of traffic protection and requires less protection capacity than the affine form. On the other hand, the optimization process for the quadratic formula is computationally more complex and potentially less efficient than for the affine formula (see Section 3.2.6).

2.2.4.2 Range

Three types of the *range of the flow thinning formula*, that defines set $\widetilde{\mathcal{E}}(d,p) \subseteq \mathcal{E}$, are considered:

- path's links $\mathcal{E}(d,p)$: the formula depends only on the degradation coefficients of the links along the path;
- path's incident links $\mathcal{E}^+(d,p) := \bigcup_{v \in \mathcal{V}(d,p)} \delta(v)$: the formula depends on the degradation coefficients of the links incident to the nodes along the path;
- all links $\mathcal{E}$: the formula depends on the degradation coefficients of all links of the network.

Clearly, the larger the range of the flow thinning formula the more effective traffic protection can be achieved. On the other hand, the range of the formula has strong impact on the implementation feasibility of the FT mechanism, since the all links range $\mathcal{E}$ requires some kind of (impractical) link capacity state signalling mechanism of the flooding kind. The two remaining range variants are much less demanding in this aspect, as the signalling conveys only the information pertaining to the links of the managed tunnel. As far as the efficiency of the optimization process is concerned, the $\mathcal{E}(d,p)$ and $\mathcal{E}$ variants considerably outperform the $\mathcal{E}^+(d,p)$ variant.

2.2.4.3 Structure

Two types of the *structures of the flow thinning formula* are considered:

- general G: $z^0_{dp} \in \mathbb{R}$ and $z_{dp} \in \mathbb{R}$;
- simple S: $z^0_{dp} = x^0_{dp}$ and $z_{dp} \in \mathbb{R}_+$.

Once again, since the simple structure is a special case of the general structure, in terms of traffic protection the latter is in general more effective than the former. However, the simple structure appears to be considerably more safe (and thus more effective) when applied to the states that are not foreseen in the optimization (i.e., the states outside the state list $\mathcal{S}$), because it always thins down the nominal flows for any non-nominal state, which may not be the case for the general structure. Moreover, with the simple formula, the more severe the link capacity degradation appears to be the more flows are thinned.

All combinations of the form, range, and structure result in 12 valid variants of the flow thinning formula presented in Table 2.1.

Table 2.1: Variants of the FT formula.

form	range	str.	formula
A	$\mathcal{E}(d,p)$	S	$x^s_{dp} = x^0_{dp} - \sum_{e \in \mathcal{E}(d,p)} \beta(e,s) z^e_{dp}$
		G	$x^s_{dp} = z^0_{dp} - \sum_{e \in \mathcal{E}(d,p)} \beta(e,s) z^e_{dp}$
	$\mathcal{E}^+(d,p)$	S	$x^s_{dp} = x^0_{dp} - \sum_{e \in \mathcal{E}^+(d,p)} \beta(e,s) z^e_{dp}$
		G	$x^s_{dp} = z^0_{dp} - \sum_{e \in \mathcal{E}^+(d,p)} \beta(e,s) z^e_{dp}$
	$\mathcal{E}$	S	$x^s_{dp} = x^0_{dp} - \sum_{e \in \mathcal{E}} \beta(e,s) z^e_{dp}$
		G	$x^s_{dp} = z^0_{dp} - \sum_{e \in \mathcal{E}} \beta(e,s) z^e_{dp}$
Q	$\mathcal{E}(d,p)$	S	$x^s_{dp} = x^0_{dp} - \sum_{e \in \mathcal{E}(d,p)} \beta(e,s) z^e_{dp} + \sum_{\{e,e'\} \in \mathcal{E}^{\lvert 2 \rvert}(d,p)} \beta(e,s)\beta(e',s) z^{ee'}_{dp}$
		G	$x^s_{dp} = z^0_{dp} - \sum_{e \in \mathcal{E}(d,p)} \beta(e,s) z^e_{dp} + \sum_{\{e,e'\} \in \mathcal{E}^{\lvert 2 \rvert}(d,p)} \beta(e,s)\beta(e',s) z^{ee'}_{dp}$
	$\mathcal{E}^+(d,p)$	S	$x^s_{dp} = x^0_{dp} - \sum_{e \in \mathcal{E}^+(d,p)} \beta(e,s) z^e_{dp} + \sum_{\{e,e'\} \in \mathcal{E}^{+\lvert 2 \rvert}(d,p)} \beta(e,s)\beta(e',s) z^{ee'}_{dp}$
		G	$x^s_{dp} = z^0_{dp} - \sum_{e \in \mathcal{E}^+(d,p)} \beta(e,s) z^e_{dp} + \sum_{\{e,e'\} \in \mathcal{E}^{+\lvert 2 \rvert}(d,p)} \beta(e,s)\beta(e',s) z^{ee'}_{dp}$
	$\mathcal{E}$	S	$x^s_{dp} = x^0_{dp} - \sum_{e \in \mathcal{E}} \beta(e,s) z^e_{dp} + \sum_{\{e,e'\} \in \mathcal{E}^{\lvert 2 \rvert}} \beta(e,s)\beta(e',s) z^{ee'}_{dp}$
		G	$x^s_{dp} = z^0_{dp} - \sum_{e \in \mathcal{E}} \beta(e,s) z^e_{dp} + \sum_{\{e,e'\} \in \mathcal{E}^{\lvert 2 \rvert}} \beta(e,s)\beta(e',s) z^{ee'}_{dp}$

As an example, the following LP problem represents the FT formulation with the flow thinning formula of affine form, simple structure, and path's links range:

Problem FT/A/S/$\mathcal{E}(d,p)$ ($\mathcal{P},\mathcal{S}$):

$$F = \min \textstyle\sum_{e\in\mathcal{E}} \xi(e)y_e^0 \tag{2.6a}$$

$$\textstyle\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}(d)} \delta(e,d,p)x_{dp}^0 \le y_e^0, \quad e\in\mathcal{E} \tag{2.6b}$$

$$\textstyle\sum_{p\in\mathcal{P}(d)} x_{dp}^s \ge h(d,s), \quad d\in\mathcal{D}, s\in\mathcal{S} \tag{2.6c}$$

$$\textstyle\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}(d)} \delta(e,d,p)x_{dp}^s \le \alpha(e,s)y_e^0, \quad e\in\mathcal{E}, s\in\mathcal{S}(e) \tag{2.6d}$$

$$x_{dp}^s = x_{dp}^0 - \textstyle\sum_{e\in\mathcal{E}(d,p)} \beta(e,s)z_{dp}^e, \quad d\in\mathcal{D}, p\in\mathcal{P}(d), s\in\mathcal{S} \tag{2.6e}$$

$$y_e^0\in\mathbb{R}_+, e\in\mathcal{E};\ x_{dp}^0, x_{dp}^s\in\mathbb{R}_+, d\in\mathcal{D}, p\in\mathcal{P}(d), s\in\mathcal{S}; \tag{2.6f}$$

$$z_{dp}^e\in\mathbb{R}_+, d\in\mathcal{D}, p\in\mathcal{P}(d), e\in\mathcal{E}(d,p). \tag{2.6g}$$

The above formulation differs from the formulation (2.2) in equalities (2.6e), which define the state-dependent path flows using the flow thinning formula. As variables z_{dp}^e are nonnegative, on the right-hand side of (2.6e) a positive quantity is always subtracted from x_{dp}^0, hence the thinning feature holds in the same way as in (2.2e). For all the variants of the flow thinning formula with general structure this is not the case, and thinning constraint (2.2e) should be explicitly added to the formulation.

Note, that introducing flow thinning formula (2.6e) leads to larger linear programs (due to additional variables z_{dp}^e) and potentially worse network cost F (due to stricter thinning) as compared to FT. Still, FT/A/S/$\mathcal{E}(d,p)$ is easier to implement in practice than FT, as flow thinning formula (2.6e) provides a reasonable flow approximation for all possible network states, also for states $s\notin\mathcal{S}$. Moreover, the real-time broadcasting of the current link capacity information also becomes easier, as calculation of the path flow size no longer requires from demands' originating nodes knowing current values of link availability coefficients of all links in the network, but only of the links along the path (note the summation in (2.6e)).

2.3 State polytope

The link availability states are expressed so far through a predefined explicit list $\mathcal{S}$. Then, the numbers of constraints and variables in the problem formulations UR, FT, and FT extended by the flow thinning formula, are proportional to the number of states. Therefore, in practice, the list cannot be too large; in particular its size should not grow exponentially with the size of the network, as otherwise the resulting formulations become non-compact and contain an excessive number of constraints and variables.

This is a well known drawback of optimization problems under uncertainty where many states are used to represent the possible values of uncertain parameters. In stochastic programming, this difficulty is alleviated by sampling relevant subset of states and by predicting future states, which leads to solutions that are good enough with high probability [39, 79, 24].

In this thesis there is no probability associated with the states. Instead, the robust optimization paradigm [6] is adopted and the predefined explicit list of states is replaced with the convex hull of the set of states, which is called the state polytope in what follows. Considering the convex hull of the set of states does not further restrict the problem, as one readily verifies that a problem solution is feasible for a set of states if, and only if, it is feasible for all states in its convex hull (for a formal proof see, e.g., [4]). It turns out that the state polytope enables considering exponentially many different states in a compact manner. The state polytope can be formally constructed as follows.

2.3.1 State polytope description

Let $\mathcal{K} = \{1, 2, \ldots, K\}$, $K \geq 1$, be a set of *link (state) types.* Then a binary vector $u = (u_e^k \in \{0, 1\},\ e \in \mathcal{E},\ k \in \mathcal{K})$ is a *state pattern* (vector) that describes the network state by defining the (state) type of each link, if it satisfies condition

$$\textstyle\sum_{k \in \mathcal{K}} u_e^k = 1,\ e \in \mathcal{E}. \tag{2.7a}$$

The idea is to make link availability coefficients and demand volumes dependent on state patterns (instead of states). For each $k \in \mathcal{K}$, let $a(k)$ and $b(k)$ be, respectively, the *link availability ratio* and the *traffic reduction coefficient* of link type k, where sequence $a = (a(k), k \in \mathcal{K})$ is strictly decreasing and sequence $b = (b(k), k \in \mathcal{K})$ is non-increasing:

$$1 = a(1) > a(2) > \ldots > a(K-1) > a(K) \geq 0 \tag{2.8a}$$

$$0 = b(1) \leq b(2) \leq \ldots \leq b(K-1) \leq b(K) < 1. \tag{2.8b}$$

For state pattern u we define *demand (volume) reduction ratio* as follows:

$$B(u) := 1 - \textstyle\sum_{k \in \mathcal{K}} b(k) U^k. \tag{2.9}$$

For each $d \in \mathcal{D}$, let $H(d)$ be the *reference demand volume* of demand d. Then, for state pattern u link availability coefficients and demand volumes are defined as:

$$\alpha(e,u) := \textstyle\sum_{k\in\mathcal{K}} a(k)u_e^k, \quad e \in \mathcal{E} \tag{2.10a}$$

$$h(d,u) := H(d)B(u), \quad d \in \mathcal{D}. \tag{2.10b}$$

Thus, the reduction of the demand volume with respect to the reference volume for each $k \in \mathcal{K}$ is proportional to the number of links U^k of type k in pattern u and their traffic reduction coefficients $b(k)$. Note that the demand volumes $H(d)$ are thinned uniformly according to $B(u)$, which plays a role of a measure of the amount of unavailable capacity (as a ratio) for state pattern u. Unfortunately, condition (2.9) allows for negative-valued demand reduction ratios. Hence, in order to fulfil the obvious requirement $0 \leq B(u) \leq 1$, the quantities $b(k)$ must satisfy the additional condition:

$$\textstyle\sum_{k=k(0)}^{K} b(k)N(k) + b(k(0)-1)(|\mathcal{E}| - \sum_{k=k(0)}^{K} N(k)) \leq 1, \tag{2.11}$$

where $k(0)$ is the smallest index k for which either of the two following conditions holds:

$$\textstyle\sum_{k}^{K} N(i) = |\mathcal{E}| \tag{2.12a}$$

$$\textstyle\sum_{k)}^{K} N(i) < |\mathcal{E}| \ \wedge \ \sum_{k-1}^{K} N(i) > |\mathcal{E}|. \tag{2.12b}$$

Now, for each $k \in \mathcal{K}$, let $N(k) \geq 0$ be an upper bound on the number of links with type k such that:

$$1 \leq N(k) \leq |\mathcal{E}|,\ k \in \mathcal{K} \tag{2.13a}$$

$$\textstyle\sum_{k\in\mathcal{K}} N(k) \geq |\mathcal{E}|. \tag{2.13b}$$

Then let $\mathcal{B}(N)$ be the set of all state pattern vectors u such that for each $k \in \mathcal{K}$, the number U^k of links of type k is bounded by $N(k)$ (note that $\sum_{k\in\mathcal{K}} U^k = |\mathcal{E}|$):

$$\textstyle\sum_{k\in\mathcal{K}} u_e^k = 1,\ e \in \mathcal{E} \tag{2.14a}$$

$$\textstyle\sum_{e\in\mathcal{E}} u_e^k = U^k,\ k \in \mathcal{K}. \tag{2.14b}$$

$$U^k \leq N(k),\ k \in \mathcal{K}. \tag{2.14c}$$

Finally, given set $\mathcal{K}$ and vectors $N = (N(k), k \in \mathcal{K})$, $a = (a(k), k \in \mathcal{K})$, $b = (b(k), k \in \mathcal{K})$, and $H = (H(d), d \in \mathcal{D})$, we define the set of states $\widehat{\mathcal{B}}(N,a,b,H) = \{s(u) : u \in \mathcal{B}(N)\}$ determined by the set of state patterns $\mathcal{B}(N)$, with each state $s(u) \in \widehat{\mathcal{B}}(N,a,b,H)$ being characterized by link availability coefficients $\alpha(u)$ specified by (2.10a), and by demand volumes $h(u)$ specified by (2.10b).

It turns out that the so defined set of states $\widehat{\mathcal{B}}(N, a, b, H)$ can be used to model a variety of particular state lists [50], for example all combinations of simultaneous degradations of at most $N(1)$ links degraded to availability ratio $a(1)$ and at most $N(2)$ links degraded to availability ratio $a(2)$ and at most $N(3)$ links degraded to availability ratio $a(3)$. Section 4.2 contains numerical study devoted, among others, to the modeling and usage of such state lists.

2.3.2 Real-valued state polytope

The state polytope introduced in the previous section is constructed using binary vectors u. Thus, any LP formulation enhanced with such a state polytope becomes a MILP formulation. Fortunately, it happens that binary vectors u can be replaced by their real-valued counterparts without any further elaboration of the state polytope, which enables solving the LP formulation.

Consider the *state polytope* $\mathcal{Q}(N)$ in the $|\mathcal{E}||\mathcal{K}|$-dimensional space of real-valued vectors with non-negative components defined by conditions (2.14); i.e., $u \in \mathbb{R}_+^{|\mathcal{E}||\mathcal{K}|}$ instead of $u \in \{0,1\}^{|\mathcal{E}||\mathcal{K}|}$. Observe that the vertices of polytope $\mathcal{Q}(N)$ are binary since the coefficient matrix specifying constraints (2.14) is identical to the one of the assignment problem, and the constraint matrix of the latter problem is known to be totally unimodular [40]. Thus, $\mathcal{Q}(N) = \mathrm{conv}(\mathcal{B}(N))$, that is $\mathcal{Q}(N)$ is the convex hull of $\mathcal{B}(N)$. Therefore the following property holds:

Property 2.1. *Optimization of any linear objective function over $\mathcal{B}(N)$ (which is a binary program) can be solved as a linear program with the same objective function over $\mathcal{Q}(N)$: an optimal vertex solution of the latter is an optimal solution of the former.*

Property 2.1 becomes crucial when solving FT optimization problems formulated with a set of states specified by $\widehat{\mathcal{B}}(N, a, b, H)$, as will be demonstrated in Section 3.3.1.

Chapter 3

Solution algorithms

The presented LP problem formulations related to FT and its formula variants are hardly scalable because of their non-compactness. Recall that for multicommodity flow networks the size (the number of variables and constraints) of a compact LP problem is polynomially bounded by the size of the graph, while the size of a non-compact LP problem grows exponentially with that size. There are two sources of non-compactness of the presented LP formulations: the number of paths and the number of states. In case of the FT formulations, both the set of path-flow variables and the set of state-dependent constraints grow exponentially with the size of the network graph. In order to solve the issue of non-compactness from the paths and states points of view, one can apply column (path) generation and row (state) generation, respectively.

3.1 Path Generation

Path Generation (PG, see [1, 43, 56]) is a well-known technique in multicommodity flow networks related to column generation in LP (see [34]). Instead of directly including all elementary paths in the formulations, PG enables considering all of them indirectly while iteratively expanding the list of the paths considered in the problem formulation.

A general idea of PG is the following. Consider the Master Problem (MP) P($\mathcal{P}$) with a limited list of paths. Start from some initial path list $\mathcal{P}$ and iteratively solve problem P($\mathcal{P}$), generate candidate paths (e.g., one per demand) and add them to the path list $\mathcal{P}$, provided that they are worth adding (i.e., may improve the solution). The so called Pricing Problem (PP) is used to generate the best candidate path based on an optimal solution of the current problem dual to P($\mathcal{P}$). Adding a new path to the path list $\mathcal{P}$ of problem P($\mathcal{P}$) results in adding new variables and constraints corresponding to the path to the problem P's formulation.

3.1.1 Dual problem to FT

Derivation of the PP for a given problem P starts from formulating the dual problem D of P. Using the FT formulation (2.2) as an example, let us derive the problem DFT dual to FT. Recall that the dual to the LP problem min $\{\underline{c}^T\underline{x} : A\underline{x} \geq \underline{b} \wedge \underline{x} \geq \underline{0} \wedge \underline{x} \in \mathbb{R}^n\}$ (in matrix notation) is an LP problem max $\{\underline{b}^T\underline{y} : A^T\underline{y} \leq \underline{c} \wedge \underline{y} \geq \underline{0} \wedge \underline{y} \in \mathbb{R}^m\}$; for more details on the dual theory refer to [31, 56]. After adding the dual variables in square brackets to the left of the constraints, the FT formulation (2.2) is as follows:

Problem FT($\mathcal{P}, \mathcal{S}$):

$$F = \min \textstyle\sum_{e\in\mathcal{E}} \xi(e)y_e^0 \tag{3.1a}$$

$$[\pi_e^0 \geq 0]\ \textstyle\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}(d)} \delta(e,d,p)x_{dp}^0 \leq y_e^0, \quad e\in\mathcal{E} \tag{3.1b}$$

$$[\lambda_d^s \geq 0]\ \textstyle\sum_{p\in\mathcal{P}(d)} x_{dp}^s \geq h(d,s), \quad d\in\mathcal{D}, s\in\mathcal{S} \tag{3.1c}$$

$$[\pi_e^s \geq 0]\ \textstyle\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}(d)} \delta(e,d,p)x_{dp}^s \leq \alpha(e,s)y_e^0, \quad e\in\mathcal{E}, s\in\mathcal{S}(e) \tag{3.1d}$$

$$[\sigma_{dp}^s \geq 0]\ x_{dp}^s \leq x_{dp}^0, \quad d\in\mathcal{D}, p\in\mathcal{P}(d), s\in\mathcal{S} \tag{3.1e}$$

$$y_e \in \mathbb{R}_+, e\in\mathcal{E};\ x_{dp}^0, x_{dp}^s \in \mathbb{R}_+, d\in\mathcal{D}, p\in\mathcal{P}(d), s\in\mathcal{S}. \tag{3.1f}$$

Thus the dual problem DFT of FT is:

Problem DFT($\mathcal{P}, \mathcal{S}$):

$$G = \max \textstyle\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}} h(d,s)\lambda_d^s \tag{3.2a}$$

$$[y_e^0 \geq 0]\ \pi_e^0 + \textstyle\sum_{s\in\mathcal{S}(e)} \alpha(e,s)\pi_e^s \leq \xi(e), \ \ e\in\mathcal{E} \tag{3.2b}$$

$$[x_{dp}^0 \geq 0]\ \textstyle\sum_{s\in\mathcal{S}} \sigma_{dp}^s \leq \sum_{e\in\mathcal{E}} \delta(e,d,p)\pi_e^0, \ \ d\in\mathcal{D},\ p\in\mathcal{P}(d) \tag{3.2c}$$

$$[x_{dp}^s \geq 0]\ \lambda_d^s \leq \sigma_{dp}^s + \textstyle\sum_{e\in\mathcal{E}(s)} \delta(e,d,p)\pi_e^s, \ \ d\in\mathcal{D},\ p\in\mathcal{P}(d),\ s\in\mathcal{S} \tag{3.2d}$$

$$\pi_e^0, \pi_e^s \in \mathbb{R}_+, e\in\mathcal{E}, s\in\mathcal{S}(e); \tag{3.2e}$$

$$\lambda_d^s, \sigma_{dp}^s \in \mathbb{R}_+, d\in\mathcal{D}, p\in\mathcal{P}(d), s\in\mathcal{S}. \tag{3.2f}$$

3.1.2 Pricing problem for FT

Now let us formulate the PP for the FT problem. First, let us fix a demand $d\in\mathcal{D}$ and consider a path $q \in \hat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$ that is not yet in the current path list $\mathcal{P}$. Adding path

q to the current path list $\mathcal{P}$ of the dual (3.2) consists in adding both new dual variables σ^s, $s \in \mathcal{S}$, and new constraints (3.2c) and (3.2d). It is possible that for all values of the variables λ^s, $s \in \mathcal{S}$, the optimal dual solution $(\pi^*,\ \lambda^*)$ will violate some of the newly added constraints. If that is the case, $(\pi^*,\ \lambda^*)$ will no longer be the optimal solution as it will be separated from the dual polyhedron of $\mathrm{DFT}(\mathcal{P} \cup \{q\}, \mathcal{S})$ by the violated constraints. The minimum over σ^s, $s \in \mathcal{S}$, of the sum of such violations is expressed as the following quantity:

$$V(q) = \min_{\sigma^s \in \mathbb{R}_+,\, s \in \mathcal{S}} \Big\{ \max \Big\{ \textstyle\sum_{s\in\mathcal{S}} \sigma^s - |q|^0, 0 \Big\} + \textstyle\sum_{s\in\mathcal{S}} \max \Big\{ \lambda_d^{s*} - \sigma^s - |q|^s, 0 \Big\} \Big\}. \quad (3.3)$$

In (3.3), $|q|^0$ is the nominal dual length of path q ($|q|^0 := \sum_{e\in\mathcal{E}(d,q)} \pi_e^{0*}$), and $|q|^s$ is the state-dependent dual length of q for state s ($|q|^s := \sum_{e\in\mathcal{E}(d,q)\cap\mathcal{E}(s)} \pi_e^{s*}$, $s \in \mathcal{S}$).

The goal of PP is to find for a given demand $d \in \mathcal{D}$ path $q \in \hat{\mathcal{P}}(d)$ that maximizes violation $V(q)$. Arguably, the positive maximum violation $V^*(q)$ implies that the corresponding path does not belong to the current path list $\mathcal{P}(d)$. Thus, the path list can be expanded by path q: $\mathcal{P}(d) := \mathcal{P}(d) \cup \{q(d)\}$. In fact, finding a path with the most violation is not necessary, since any path with positive violation is a valid candidate to be added to the path list. Still, adding paths with the maximum violation can significantly accelerate the process of path generation, provided that finding each such path is not significantly more time consuming as compared to finding any positive-violation path that is not yet in the path list. The presented PP that maximizes $V(q)$ can be formulated as the following MILP problem (skipping the superscript * in π^*, λ^*):

Problem PP-FT(d, λ, π):

$$\min \textstyle\sum_{e\in\mathcal{E}} \pi_e^0 u_e + \sum_{s\in\mathcal{S}} \Big(\sum_{e\in\mathcal{E}(s)} \pi_e^s u_e - \lambda_d^s \Big) Y^s \quad (3.4a)$$

$$\textstyle\sum_{e\in\delta(v)} u_e = 1, \quad v \in \{o(d), t(d)\} \quad (3.4b)$$

$$\textstyle\sum_{e\in\delta(v)} u_e = 2r_v, \quad v \in \mathcal{V} \setminus \{o(d), t(d)\} \quad (3.4c)$$

$$u_e \in \mathbb{B},\ e \in \mathcal{E}; \quad r_v \in \mathbb{B},\ v \in \mathcal{V} \setminus \{o(d), t(d)\}; \quad Y^s \in \mathbb{B},\ s \in \mathcal{S}. \quad (3.4d)$$

An optimal solution $(u^*,\ r^*,\ Y^*)$ of (3.4) defines a path $q := \{e \in \mathcal{E} : u_e^* = 1\}$ that maximizes the violation $V(q)$ of (3.3). First, equations (3.4b) and (3.4c) ensure that the obtained path is indeed a valid path between $o(d)$ and $t(d)$. Second, the objective (3.4a) is constructed in such a way, that in any optimal solution $Y^s = 1$ when $|q|^s < \lambda_d^s$ and $Y^s = 0$ when $|q|^s > \lambda_d^s$: as the objective (3.4a) is of minimization type, all the terms

$\sum_{e\in\mathcal{E}} \pi_e^s u_e - \lambda_d^s$ that are positive will be eliminated, while all the negative terms will be kept. When the violation is positive, the so defined path q should be added to DFT.

Observe, that formulation (3.4) is nonlinear due to the multiplication of variables u_e and Y^s in the objective (3.4a). In order to eliminate such bi-linearities let us rewrite the formulation (3.4) in the following way:

Problem PP-FT(d, λ, π):

$$\min \sum_{e\in\mathcal{E}} \pi_e^0 u_e + \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}(s)} \pi_e^s Z_e^s - \sum_{s\in\mathcal{S}} \lambda_d^s Y^s \tag{3.5a}$$

$$\sum_{e\in\delta(v)} u_e = 1, \quad v \in \{o(d), t(d)\} \tag{3.5b}$$

$$\sum_{e\in\delta(v)} u_e = 2r_v, \quad v \in \mathcal{V} \setminus \{o(d), t(d)\} \tag{3.5c}$$

$$Z_e^s \le u_e,\ Z_e^s \le Y^s,\ Z_e^s \ge 0,\ Z_e^s \ge u_e + Y^s - 1, \quad s \in \mathcal{S},\ e \in \mathcal{E}(s) \tag{3.5d}$$

$$u_e \in \mathbb{B},\ e \in \mathcal{E};\quad r_v \in \mathbb{B},\ v \in \mathcal{V} \setminus \{o(d), t(d)\};\ Y^s \in \mathbb{B},\ s \in \mathcal{S}; \tag{3.5e}$$

$$Z_e^s \in \mathbb{R},\ s \in \mathcal{S}, e \in \mathcal{E}(s). \tag{3.5f}$$

In (3.5), new binary variables Z_e^s, $s \in \mathcal{S}$, $e \in \mathcal{E}(s)$, substitute the product $u_e Y^s$ in the objective, while constraints (3.5d) force that $Z_e^s = u_e Y^s$ for all $s \in \mathcal{S}$, $e \in \mathcal{E}(s)$. Note, that as variables u_e and Y^s are binary, variables Z can be continuous (thus, $Z_e^s \in \mathbb{R}$ is assumed).

Certainly, it is possible to modify formulation (3.5) for directed networks (with links in $\mathcal{E}$ being directed) using the node-arc formulation with load conservation equations instead of (3.5b)-(3.5c):

$$\sum_{e\in\delta^-(t(d))} u_e - \sum_{e\in\delta^+(t(d))} u_e = 1 \tag{3.6a}$$

$$\sum_{e\in\delta^+(v)} u_e - \sum_{e\in\delta^-(v)} u_e = 0, \quad v \in \mathcal{V} \setminus \{o(d), t(d)\}. \tag{3.6b}$$

3.1.3 Path Generation algorithm

Using the derived dual (3.2) and pricing (3.5) problems for FT (2.2), the path generation algorithm can be stated in the form of the following pseudocode:

Algorithm 1 PGA($\mathcal{P}_0$)

```
iter ← 0
repeat
    (π*, λ*) ← DFT(P_iter)
    P_{iter+1} ← P_iter
```

5: **for** $d \in \mathcal{D}$ **do**

6: $(q^*, F^*) \leftarrow$ PP-FT(d, π^*, λ^*)

7: **if** $F^* > 0$ **then**

8: $\mathcal{P}_{iter+1} \leftarrow \mathcal{P}_{iter+1} \cup \{q^*\}$

9: **end if**

10: **end for**

11: $iter \leftarrow iter + 1$

12: **until** $\mathcal{P}_{iter} = \mathcal{P}_{iter-1}$

Algorithm 1 (PGA) accepts initial path list $\mathcal{P}_0$ as an input argument. The initial path list should contain at least one path for each demand. The main loop (lines 2-12) of the algorithm starts in line 3 from solving problem (3.2) and obtaining its optimal solution (π^*, λ^*). Next, in lines 5-10, the algorithm tries to find for each demand a candidate path q^* by solving PP-FT (formulation (3.5)). In case of success, (i.e, when the obtained objective function value F^* of PP-FT is positive), path q^* is added to the path list $\mathcal{P}_{iter}$. The main loop is repeated until the path list has been extended by at least one path. Otherwise, the algorithm stops and the final path list is sufficient to solve FT (2.2) optimally. Actually, there is no need to solve FT after PGA as it has already been solved indirectly in the last iteration of the main loop in line 3.

Note that in the implementation of Algorithm 1 one can use a single path list $\mathcal{P}$ instead of consecutive path lists $\mathcal{P}_{iter}$ as there is no real dependence between the path list in a given iteration and the path lists from previous iteration. Also note, that partial solutions of the algorithm obtained after every iteration of the main loop in line 3 are valid, though not optimal, solutions of FT.

The presented algorithm is of a general form and is valid for all the variants of the FT formula described in Section 2.2.4. In order to apply Algorithm 1 for the FT formulation with a certain variant of the FT formula, one should replace the dual problem DFT (in line 3) and the pricing problem PP-FT (in line 6) with their counterparts, appropriately derived for the considered formulation. The following section is dedicated to derivation of pricing problems for formulations with selected variants of the FT formula.

3.2 Pricing problems for FT variants

Pricing problems for formulations with various FT formulae turn out to be more complex as compared to PP for FT. The derivations are similar to the one already presented, yet they consist of some additional steps. Since some of the derivations have the same starting part, that common part is presented first until branching out to specific variants of FT formula is inevitable. Let us start with deriving the pricing problem for FT with affine formula, simple structure, and an arbitrary range (i.e., FT/A/S/$\widetilde{\mathcal{E}}(d,p)$), and then proceed with specific ranges.

3.2.1 PP for FT/A/S/$\widetilde{\mathcal{E}}(d,p)$

The primal and dual LP formulations for FT/A/S/$\widetilde{\mathcal{E}}(d,p)$ $(\mathcal{P},\mathcal{S})$ are as follows:

Problem FT/A/S/$\widetilde{\mathcal{E}}(d,p)$ $(\mathcal{P},\mathcal{S})$:

$$F = \min \textstyle\sum_{e\in\mathcal{E}} \xi(e)y_e^0 \tag{3.7a}$$

$$[\pi_e^0 \geq 0] \;\; \textstyle\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{R}(d,e)} x_{dp}^0 \leq y_e^0, \quad e\in\mathcal{E} \tag{3.7b}$$

$$[\lambda_d^s \geq 0] \;\; \textstyle\sum_{p\in\mathcal{P}(d)} x_{dp}^s \geq h(d,s), \quad d\in\mathcal{D},\ s\in\mathcal{S} \tag{3.7c}$$

$$[\pi_e^s \geq 0] \;\; \textstyle\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}(d)} \delta(e,d,p)x_{dp}^s \leq \alpha(e,s)y_e^0, \quad e\in\mathcal{E},\ s\in\mathcal{S}(e) \tag{3.7d}$$

$$[\varphi_{dp}^s] \;\; x_{dp}^s = x_{dp}^0 - \textstyle\sum_{e\in\widetilde{\mathcal{E}}(d,p)} \beta(e,s)z_{dp}^e, \quad d\in\mathcal{D},\ p\in\mathcal{P}(d),\ s\in\mathcal{S} \tag{3.7e}$$

$$y_e \in \mathbb{R}_+, e\in\mathcal{E};\ x_{dp}^0, x_{dp}^s \in \mathbb{R}_+, d\in\mathcal{D}, p\in\mathcal{P}(d), s\in\mathcal{S}; \tag{3.7f}$$

$$z_{dp}^e \in \mathbb{R}_+, d\in\mathcal{D}, p\in\mathcal{P}(d), e\in\widetilde{\mathcal{E}}(d,p). \tag{3.7g}$$

Problem DFT/A/S/$\widetilde{\mathcal{E}}(d,p)$ $(\mathcal{P},\mathcal{S})$:

$$G = \max \textstyle\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}} h(d,s)\lambda_d^s \tag{3.8a}$$

$$[y_e^0 \geq 0] \;\; \pi_e^0 + \textstyle\sum_{s\in\mathcal{S}(e)} \alpha(e,s)\pi_e^s \leq \xi(e), \quad e\in\mathcal{E} \tag{3.8b}$$

$$[x_{dp}^0 \geq 0] \;\; \textstyle\sum_{s\in\mathcal{S}} \varphi_{dp}^s \leq \sum_{e\in\mathcal{E}(d,p)} \pi_e^0, \quad d\in\mathcal{D},\ p\in\mathcal{P}(d) \tag{3.8c}$$

$$[x_{dp}^s \geq 0] \;\; \lambda_d^s \leq \varphi_{dp}^s + \textstyle\sum_{e\in\mathcal{E}(d,p)\cap\mathcal{E}(s)} \pi_e^s, \quad d\in\mathcal{D},\ p\in\mathcal{P}(d),\ s\in\mathcal{S} \tag{3.8d}$$

$$[z_{dp}^e \geq 0] \;\; \textstyle\sum_{s\in\mathcal{S}} \beta(e,s)\varphi_{dp}^s \geq 0, \quad d\in\mathcal{D},\ p\in\mathcal{P}(d),\ e\in\widetilde{\mathcal{E}}(d,p) \tag{3.8e}$$

$$\pi_e^0, \pi_e^s \in \mathbb{R}_+, e\in\mathcal{E}, s\in\mathcal{S}(e);\ \lambda_d^s \in \mathbb{R}_+,\ d\in\mathcal{D},\ s\in\mathcal{S}; \tag{3.8f}$$

$$\varphi_{dp}^s \in \mathbb{R},\ d\in\mathcal{D},\ p\in\mathcal{P}(d),\ s\in\mathcal{S}. \tag{3.8g}$$

Similarly to the derivation of PP in Section 3.1.2, let us fix a demand d, and consider a path q that is not yet in the current path list $\mathcal{P}$ (i.e., $q \in \hat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$). Adding path q

to the current path list $\mathcal{P}$ of the dual (3.8) consists in adding both new dual variables φ^s, $s \in \mathcal{S}$, and new constraints (3.8c)-(3.8e). It is possible that for all values of the variables φ^s, $s \in \mathcal{S}$, the optimal dual solution $(\pi^*,\ \lambda^*)$ will violate some of the newly added constraints. If that is the case, $(\pi^*,\ \lambda^*)$ will no longer be the optimal solution as it will be separated from the dual polyhedron of DFT/A/S/$\widetilde{\mathcal{E}}(d,p)(\mathcal{P} \cup \{q\}, \mathcal{S})$ by the violated constraints. The minimum over the vector $\varphi := (\varphi^s, s \in \mathcal{S})$ of the sum of such violations is expressed as the following quantity (recall that $|q|^0 = \sum_{e \in \mathcal{E}(d,q)} \pi_e^{0*}$ and $|q|^s = \sum_{e \in \mathcal{E}(d,q) \cap \mathcal{E}(s)} \pi_e^{s*}$, $s \in \mathcal{S}$):

$$
\begin{aligned}
V(q) = \min_{\varphi^s \in \mathbb{R},\, s \in \mathcal{S}} \Big\{ & \max \Big\{ \textstyle\sum_{s \in \mathcal{S}} \varphi^s - |q|^0, 0 \Big\} + \\
& \textstyle\sum_{s \in \mathcal{S}} \max \Big\{ \lambda_d^{s*} - |q|^s - \varphi^s, 0 \Big\} + \\
& \textstyle\sum_{e \in \widetilde{\mathcal{E}}(d,q)} \max \Big\{ -\sum_{s \in \mathcal{S}} \beta(e,s)\varphi^s, 0 \Big\} \Big\}.
\end{aligned} \tag{3.9a}
$$

The following LP allows to evaluate $V(q)$ for a given path $q \in \hat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$:

$$V(q) = \min \Big\{ Z + \textstyle\sum_{s \in \mathcal{S}} Y_s + \sum_{e \in \widetilde{\mathcal{E}}(d,q)} X_e \Big\} \tag{3.10a}$$

$$[g \geq 0] \quad Z \geq \textstyle\sum_{s \in \mathcal{S}} \varphi^s - |q|^0 \tag{3.10b}$$

$$[a_s \geq 0] \quad Y_s \geq \lambda_d^{s*} - |q|^s - \varphi^s, \quad s \in \mathcal{S} \tag{3.10c}$$

$$[b_e \geq 0] \quad X_e \geq -\textstyle\sum_{s \in \mathcal{S}} \beta(e,s)\varphi^s, \quad e \in \widetilde{\mathcal{E}}(d,q) \tag{3.10d}$$

$$Z \in \mathbb{R}_+;\ Y_s \in \mathbb{R}_+,\ s \in \mathcal{S};\ X_e \in \mathbb{R}_+,\ e \in \widetilde{\mathcal{E}}(d,q);\ \varphi^s \in \mathbb{R},\ s \in \mathcal{S}. \tag{3.10e}$$

We should find a $q \in \hat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$ that maximizes $V(q)$. Since $V(p)$ is equal to zero for each path $p \in \mathcal{P}(d)$, and a new path q is added to the current path list $\mathcal{P}(d)$ only when $V(q)$ is greater than zero, $V(q)$ can be maximized over all paths $q \in \widehat{\mathcal{P}}(d)$. Thus, we might solve problem (3.10) with the objective (3.10a) replaced with

$$\max_{q \in \widehat{\mathcal{P}}(d)} \min_{X,Y,Z,\varphi} \Big\{ Z + \textstyle\sum_{s \in \mathcal{S}} Y_s + \sum_{e \in \widetilde{\mathcal{E}}(d,q)} X_e \Big\}, \tag{3.11}$$

where vectors X and Y defined as $X := (X_e,\ e \in \widetilde{\mathcal{E}}(d,q))$ and $Y := (Y_s,\ s \in \mathcal{S})$, respectively.

Unfortunately, formulation (3.10) with objective (3.10a) replaced with (3.11) cannot be solved directly. The reason is that our goal is to maximize the value of $V(q)$ over, in particular, the paths outside the current list, i.e., for $q \in \widehat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$, and this cannot be achieved with the minimization objective as in (3.10a). This issue can be resolved by considering the problem dual to (3.10), formulated using the dual variables specified in the square brackets in constraints (3.10b)-(3.10d):

$$V(q) = \max \Big\{ -g|q|^0 + \textstyle\sum_{s\in\mathcal{S}}(g - \sum_{e\in\widetilde{\mathcal{E}}(d,q)} \beta(e,s)b_e)(\lambda_d^{s*} - |q|^s)) \Big\} \tag{3.12a}$$

$$g \le 1 \tag{3.12b}$$

$$b_e \le 1,\ e \in \widetilde{\mathcal{E}}(d,q) \tag{3.12c}$$

$$\textstyle\sum_{e\in\widetilde{\mathcal{E}}(d,q)} \beta(e,s)b_e \le g,\ s \in \mathcal{S} \tag{3.12d}$$

$$g \in \mathbb{R}_+;\ b_e \in \mathbb{R}_+,\ e \in \widetilde{\mathcal{E}}(d,q). \tag{3.12e}$$

Note that dual variables a_s, $s \in \mathcal{S}$, have been eliminated in (3.12). This is possible because in (3.7) variables x_{dp}^s, $d \in \mathcal{D}$, $p \in \mathcal{P}(d)$, $s \in \mathcal{S}$, are auxiliary due to the equality type of constraint (3.7e). Thus, variables x_{dp}^s can be expressed through x_{dp}^0 and z_{dp}^e, $e \in \widetilde{\mathcal{E}}(d,p)$, provided that x_{dp}^s is kept positive, i.e., $x_{dp}^0 \ge \sum_{e\in\widetilde{\mathcal{E}}(d,p)} \beta(e,s)z_{dp}^e$.

The next step is to introduce binary variables u_e, $e \in \mathcal{E}$, that define the set of links that represent path q: $\mathcal{E}(d,q) := \{e \in \mathcal{E} : u_e = 1\}$. With those variables, nominal and state-dependent dual lengths of q take the form of $|q|^0 = \sum_{e\in\mathcal{E}} \pi_e^0 u_e$ and $|q|^s = \sum_{e\in\mathcal{E}(s)} \pi_e^s u_e$, $s \in \mathcal{S}$, respectively. Therefore, objective (3.12a) can be rewritten as follows (note that from now on the superscript * in π^* and λ^* is skipped):

$$V(u) = \max\ \Big\{ -g(\textstyle\sum_{e\in\mathcal{E}} \pi_e^0 u_e) + \sum_{s\in\mathcal{S}} \Big(g - \sum_{e'\in\widetilde{\mathcal{E}}(d,q)} \beta(e',s)b_{e'})(\lambda_d^s - \sum_{e\in\mathcal{E}(s)} \pi_e^s u_e)\Big\}. \tag{3.13}$$

Finally, adding path-defining constraints (3.4b)-(3.4c) (recall that $\delta(v)$ in those constraints denotes the set of all links incident to node $v \in \mathcal{V}$), treating u_e, $e \in \mathcal{E}$, as variables, and transforming objective (3.13) leads to the following formulation of the PP for FT/A/S/$\widetilde{\mathcal{E}}(d,p)$ (formulated as (3.7)):

Problem PP-FT/A/S/$\widetilde{\mathcal{E}}(d,q)(d,\lambda,\pi)$:

$$\begin{aligned} V = \max \Big\{ &-\textstyle\sum_{e\in\mathcal{E}} \pi_e^0 g u_e + (\sum_{s\in\mathcal{S}} \lambda_d^s) g - \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}(s)} \pi_e^s g u_e + \\ &\textstyle\sum_{s\in\mathcal{S}} \sum_{e'\in\widetilde{\mathcal{E}}(d,q)} \sum_{e\in\mathcal{E}(s)} \beta(e',s)\pi_e^s b_{e'} u_e - \\ &\textstyle\sum_{s\in\mathcal{S}} \sum_{e'\in\widetilde{\mathcal{E}}(d,q)} \lambda_d^s \beta(e',s) b_{e'} \Big\} \end{aligned} \tag{3.14a}$$

$$\textstyle\sum_{e\in\delta(v)} u_e = 1, \quad v \in \{o(d), t(d)\} \tag{3.14b}$$

$$\textstyle\sum_{e\in\delta(v)} u_e = 2r_v, \quad v \in \mathcal{V} \setminus \{o(d), t(d)\} \tag{3.14c}$$

$$g \le 1 \tag{3.14d}$$

$$b_{e'} \leq 1,\ e' \in \widetilde{\mathcal{E}}(d,q) \tag{3.14e}$$

$$\textstyle\sum_{e' \in \widetilde{\mathcal{E}}(d,q)} \beta(e',s) b_{e'} \leq g,\ s \in \mathcal{S} \tag{3.14f}$$

$$g \in \mathbb{R}_+;\ \ b_{e'} \in \mathbb{R}_+,\ e' \in \widetilde{\mathcal{E}}(d,q); \tag{3.14g}$$

$$u_e \in \mathbb{B},\ e \in \mathcal{E};\ \ r_v \in \mathbb{B},\ v \in \mathcal{V}. \tag{3.14h}$$

An optimal solution (u^*, r^*) of the above formulation defines (through constraints (3.14b)-(3.14c)) the path q, and more specifically, the set of links and nodes that q consists of: $\mathcal{E}(d,q) := \{e \in \mathcal{E} : u_e^* = 1\}$ and $\mathcal{V}(d,q) := \{v \in \mathcal{V} : r_v^* = 1\}$. The so obtained path q should be added to (3.7) if $V(u) > 0$.

Note that constraints (3.14b)-(3.14c) ensure that any feasible set $\mathcal{U} := \{e \in \mathcal{E} : u_e = 1\}$ contains an elementary path connecting $o(d)$ and $t(d)$ and, possibly, a set of disjoint isolated loops. As variables r_v, $v \in \mathcal{V}$ are binary, the path itself is necessarily elementary (note that making these variables integer-valued would allow the path to visit a certain node several times and thus to contain loops). The reason isolated loops may appear in optimal PP solutions is a tradeoff in the number of elements in the set $\mathcal{U}$: the less elements the lower link capacities are, and the more elements the more flexible are flow thinning formulae (3.7e).

Paths with isolated loops are not only meaningless in network terms, but potentially may also prolong the path generation process. In order to eliminate the isolated loops in $\{e \in \mathcal{E} : u_e = 1\}$ consider the bi-directed version $\mathcal{G}' = (\mathcal{V}, \mathcal{A})$ of the original undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In $\mathcal{G}'$, each undirected link $e \in \mathcal{E}$ is substituted with two oppositely directed arcs $a'(e)$ and $a''(e)$, hence defining the set of directed arcs as $\mathcal{A} := \{(v,w),(w,v) : \{v,w\} \in \mathcal{E}\}$. Adding aggregated arc-flow variables $f_a \in \mathbb{R}_+$, $a \in \mathcal{A}$, together with the following constraints to (3.14) will force that the isolated loops will no longer appear in the feasible solutions:

$$\textstyle\sum_{a \in \delta^-(t(d))} f_a = \sum_{v \in \mathcal{V} \setminus \{o(d),t(d)\}} r_v \tag{3.15a}$$

$$\textstyle\sum_{a \in \delta^+(o(d))} f_a = \sum_{a \in \delta^-(o(d))} f_a \tag{3.15b}$$

$$\textstyle\sum_{a \in \delta^+(v)} f_a = \sum_{a \in \delta^-(v)} f_a + r_v,\ \ v \in \mathcal{V} \setminus \{o(d), t(d)\} \tag{3.15c}$$

$$f_{a'(e)} + f_{a''(e)} \leq (|\mathcal{V}| - 2) u_e,\ \ e \in \mathcal{E} \tag{3.15d}$$

$$f_a \in \mathbb{R}_+,\ a \in \mathcal{A}, \tag{3.15e}$$

where $\delta^-(v)$, $v \in \mathcal{V}$, denotes the set of all arcs incoming to node v, and $\delta^+(v)$ denotes the set of all arcs outgoing from node v. Note that the number of constraints in (3.15) does not depend on the number of paths or states, thus adding these constraints to the PP formulations should not considerably worsen the computational efficiency of the PP's.

Formulation (3.14) is still incomplete for two reasons: it contains products of variables in the objective function (such as gu_e, $b_{e'}u_e$), and the range of the flow thinning formula $\widetilde{\mathcal{E}}(d,q)$ is arbitrary. A proper MILP formulation requires specifying the range of the flow thinning formula and its representation by means of optimization variables. The products of variables can be appropriately linearized only after the range of the flow thinning formula is fixed. The following sections contain complete and linearized MILP formulations of PP for different thinning formula ranges.

3.2.2 PP for FT/A/S/$\mathcal{E}(d,p)$

The proper PP formulation for the range $\widetilde{\mathcal{E}}(d,p) = \mathcal{E}(d,p)$ is as follows:

$$V = \max \Big\{ -\textstyle\sum_{e\in\mathcal{E}} \pi_e^0 G_e + (\sum_{s\in\mathcal{S}} \lambda_d^s) g - \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}(s)} \pi_e^s G_e +$$
$$\textstyle\sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}(s)} \sum_{e'\in\mathcal{E}} \pi_e^s \beta(e',s) T_{e'e} - \sum_{s\in\mathcal{S}} \sum_{e'\in\mathcal{E}} \lambda_d^s \beta(e',s) B_{e'} \Big\} \tag{3.16a}$$
$$\textstyle\sum_{e\in\delta(v)} u_e = 1,\ v \in \{s(d), t(d)\} \tag{3.16b}$$
$$\textstyle\sum_{e\in\delta(v)} u_e = 2r_v,\ v \in \mathcal{V} \setminus \{s(d), t(d)\} \tag{3.16c}$$
$$g \le 1 \tag{3.16d}$$
$$b_e \le 1,\ e \in \mathcal{E} \tag{3.16e}$$
$$\textstyle\sum_{e\in\mathcal{E}} \beta(e,s) B_e \le g,\ s \in \mathcal{S} \tag{3.16f}$$
$$B_e \le u_e, B_e \le b_e, B_e \ge b_e + u_e - 1,\ \ e \in \mathcal{E} \tag{3.16g}$$
$$G_e \le g, G_e \le u_e, G_e \ge g + u_e - 1,\ \ e \in \mathcal{E} \tag{3.16h}$$
$$T_{e'e} \le u_{e'}, T_{e'e} \le u_e, T_{e'e} \le b_{e'}, T_{e'e} \ge b_{e'} + u_{e'} + u_e - 2,\ \ e', e \in \mathcal{E} \tag{3.16i}$$
$$\text{isolated-loops-elimination constraints (3.15a)} - \text{(3.15e)} \tag{3.16j}$$
$$g \in \mathbb{R}_+;\ \ b_e, B_e, G_e \in \mathbb{R}_+,\ e \in \mathcal{E};\ \ T_{e'e} \in \mathbb{R}_+,\ e', e \in \mathcal{E}; \tag{3.16k}$$
$$u_e \in \mathbb{B},\ e \in \mathcal{E};\ \ r_v \in \mathbb{B},\ v \in \mathcal{V} \setminus \{s(d), t(d)\}. \tag{3.16l}$$

In (3.16), the extra variables are used to eliminate the product of variables (similarly to what was done in (3.5)) and have the following meaning: $B_e = b_e u_e$, $G_e = g b_e$, $T_{e'e} = b_{e'} u_{e'} u_e$.

3.2.3 PP for FT/A/S/$\mathcal{E}^+(d,p)$

For this variant, the PP formulation is obtained using the following equality which is valid for an arbitrary vector of link-dependent quantities $(A(e),\ e \in \mathcal{E})$, where $\mathcal{E}(d,q) = \{e \in \mathcal{E} : u_e = 1\}$ and $\mathcal{V}(d,q) = \{v \in \mathcal{V} : r_v = 1\}$:

$$\textstyle\sum_{e\in\mathcal{E}^+(d,q)} A(e) = \sum_{v\in\mathcal{V}} \sum_{e\in\delta(v)} A(e)r_v - \sum_{e\in\mathcal{E}} A(e)u_e, \quad s \in \mathcal{S}.$$

The above equality allows replacing the (undefined beforehand) summation over the set $\mathcal{E}^+(d,q)$ with an expression that depends on variables r_v and u_e. Still, the MILP formulation of PP becomes more complex than its counterpart (3.16) for the simpler range $\mathcal{E}(d,p)$:

$$\begin{aligned}
V = \max \Big\{ & -\textstyle\sum_{e\in\mathcal{E}} \pi_e^0 G_e + (\sum_{s\in\mathcal{S}} \lambda_d^s)g - \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}(s)} \pi_e^s G_e + \\
& \textstyle\sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}(s)} \Big(\sum_{v\in\mathcal{V}} \sum_{e'\in\delta(v)} \pi_e^s \beta(e',s) R_{e'ev} - \sum_{e'\in\mathcal{E}} \pi_e^s \beta(e',s) T_{e'e} \Big) - \\
& \textstyle\sum_{s\in\mathcal{S}} \Big(\sum_{v\in\mathcal{V}} \sum_{e'\in\delta(v)} \lambda_d^s \beta(e',s) H_{e'v} - \sum_{e'\in\mathcal{E}} \lambda_d^s \beta(e',s) B_{e'} \Big) \Big\}
\end{aligned} \tag{3.17a}$$

$$\textstyle\sum_{e\in\delta(v)} u_e = 1,\ v \in \{s(d), t(d)\} \tag{3.17b}$$

$$\textstyle\sum_{e\in\delta(v)} u_e = 2r_v,\ v \in \mathcal{V} \setminus \{s(d), t(d)\} \tag{3.17c}$$

$$g \le 1 \tag{3.17d}$$

$$b_e \le 1,\ e \in \mathcal{E} \tag{3.17e}$$

$$\textstyle\sum_{v\in\mathcal{V}} \sum_{e\in\delta(v)} \beta(e,s) H_{ev} - \sum_{e\in\mathcal{E}} \beta(e,s) B_e \le g,\ s \in \mathcal{S} \tag{3.17f}$$

$$B_e \le u_e, B_e \le b_e, B_e \ge b_e + u_{e'} - 1, \quad e \in \mathcal{E} \tag{3.17g}$$

$$H_{ev} \le b_e, H_{ev} \le r_v, H_{ev} \ge b_e + r_v - 1, \quad v \in \mathcal{V},\ e \in \delta(v) \tag{3.17h}$$

$$G_e \le g, G_e \le u_e, G_e \ge g + u_e - 1, \quad e \in \mathcal{E} \tag{3.17i}$$

$$T_{e'e} \le u_{e'}, T_{e'e} \le u_e, T_{e'e} \le b_{e'}, T_{e'e} \ge b_{e'} + u_{e'} + u_e - 2, \quad e', e \in \mathcal{E} \tag{3.17j}$$

$$\begin{aligned}
& R_{e'ev} \le b_{e'}, R_{e'ev} \le u_e, H_{e'e} \le r_v, H_{e'ev} \ge b_{e'} + u_e + r_v - 2, \\
& \qquad\qquad v \in \mathcal{V},\ e' \in \delta(v),\ e \in \mathcal{E}
\end{aligned} \tag{3.17k}$$

$$\text{isolated-loops-elimination constraints (3.15a)} - \text{(3.15e)} \tag{3.17l}$$

$$g \in \mathbb{R}_+;\ b_e \in \mathbb{R}_+,\ e \in \mathcal{E};\quad u_e \in \mathbb{B},\ e \in \mathcal{E}; \tag{3.17m}$$

$$r_v \in \mathbb{B},\ v \in \mathcal{V} \setminus \{s(d), t(d)\}; \tag{3.17n}$$

$$B_e \in \mathbb{R}_+,\ e \in \mathcal{E};\quad G_e \in \mathbb{R}_+,\ e \in \mathcal{E};\quad T_{e'e} \in \mathbb{R}_+,\ e', e \in \mathcal{E}; \tag{3.17o}$$

$$H_{ev} \in \mathbb{R}_+,\ e \in \mathcal{E},\ v \in \mathcal{V};\quad R_{e'ev},\ e', e \in \mathcal{E},\ v \in \mathcal{V}. \tag{3.17p}$$

In (3.17), the extra real-valued variables B_e, G_e, $T_{e'e}$, H_{ev}, $R_{e'ev}$ are used to eliminate the products of pairs of the original variables that appear in the objective function through constraints (3.17g)-(3.17k). Note that $T_{e'e}$ and $R_{e'ev}$ express the products of three original variables.

3.2.4 PP for FT/A/S/$\mathcal{E}$

For this variant, the PP formulation becomes simpler than (3.16) since now variables $B_{e'}$ are equal to $b_{e'}$ rather than to $b_{e'}u_{e'}$, and variables $T_{e'e}$ express the product $b_{e'}u_e$ rather than $b_{e'}u_{e'}u_e$.

$$V = \max \Big\{ - \textstyle\sum_{e\in\mathcal{E}} \pi_e^0 G_e + (\sum_{s\in\mathcal{S}} \lambda_d^s) g - \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}(s)} \pi_e^s G_e + \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}(s)} \sum_{e'\in\mathcal{E}} \pi_e^s \beta(e',s) T_{e'e} - \sum_{s\in\mathcal{S}} \sum_{e'\in\mathcal{E}} \lambda_d^s \beta(e',s) b_{e'} \Big\} \tag{3.18a}$$

$$\textstyle\sum_{e\in\delta(v)} u_e = 1,\ v \in \{s(d), t(d)\} \tag{3.18b}$$

$$\textstyle\sum_{e\in\delta(v)} u_e = 2r_v,\ v \in \mathcal{V} \setminus \{s(d), t(d)\} \tag{3.18c}$$

$$g \le 1 \tag{3.18d}$$

$$b_e \le 1,\ e \in \mathcal{E} \tag{3.18e}$$

$$\textstyle\sum_{e\in\mathcal{E}} \beta(e,s) b_e \le g,\ s \in \mathcal{S} \tag{3.18f}$$

$$G_e \le g, G_e \le u_e, G_e \ge g + u_e - 1,\quad e \in \mathcal{E} \tag{3.18g}$$

$$T_{e'e} \le u_e, T_{e'e} \le b_{e'}, T_{e'e} \ge b_{e'} + u_e - 1,\quad e', e \in \mathcal{E} \tag{3.18h}$$

$$g \in \mathbb{R}_+;\quad b_e \in \mathbb{R}_+,\ e \in \mathcal{E};\quad G_e \in \mathbb{R}_+,\ e \in \mathcal{E};\quad T_{e'e} \in \mathbb{R}_+,\ e', e \in \mathcal{E} \tag{3.18i}$$

$$u_e \in \mathbb{B},\ e \in \mathcal{E};\quad r_v \in \mathbb{B},\ v \in \mathcal{V} \setminus \{s(d), t(d)\}. \tag{3.18j}$$

Note that isolated-loops-elimination constraints (3.15a)-(3.15e) are not required in this formulation as the loops in the optimal path will not appear due to the maximal (all links) range of the flow thinning formula.

3.2.5 PP for FT/A/G/$\widetilde{\mathcal{E}}(d,p)$

Now let us consider problem formulation (2.2) extended by the flow thinning formula with affine form, general structure and an arbitrary range, which results in the following formulation:

Problem FT/A/G/$\widetilde{\mathcal{E}}(d,p)$ $(\mathcal{P},\mathcal{S})$:

$$F = \min \textstyle\sum_{e\in\mathcal{E}} \xi(e) y_e^0 \tag{3.19a}$$

$$[\pi_e^0 \geq 0] \quad \textstyle\sum_{d\in\mathcal{D}} \sum_{p\in\mathcal{R}(d,e)} x_{dp}^0 \leq y_e^0, \quad e \in \mathcal{E} \tag{3.19b}$$

$$[\lambda_d^s \geq 0] \quad \textstyle\sum_{p\in\mathcal{P}(d)} x_{dp}^s \geq h(d,s), \quad d \in \mathcal{D},\ s \in \mathcal{S} \tag{3.19c}$$

$$[\pi_e^s \geq 0] \quad \textstyle\sum_{d\in\mathcal{D}} \sum_{p\in\mathcal{P}(d)} \delta(e,d,p) x_{dp}^s \leq \alpha(e,s) y_e^0, \quad e \in \mathcal{E},\ s \in \mathcal{S}(e) \tag{3.19d}$$

$$[\varphi_{dp}^s] \quad x_{dp}^s = z_{dp}^0 - \textstyle\sum_{e\in\widetilde{\mathcal{E}}(d,p)} \beta(e,s) z_{dp}^e, \quad d \in \mathcal{D},\ p \in \mathcal{P}(d),\ s \in \mathcal{S} \tag{3.19e}$$

$$[\sigma_{dp}^s \geq 0] \quad x_{dp}^s \leq x_{dp}^0, \quad d \in \mathcal{D},\ p \in \mathcal{P}(d),\ s \in \mathcal{S} \tag{3.19f}$$

$$y_e \in \mathbb{R}_+, e \in \mathcal{E};\ x_{dp}^0, x_{dp}^s \in \mathbb{R}_+, d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S}; \tag{3.19g}$$

$$z_{dp}^0, z_{dp}^e \in \mathbb{R}, e \in \widetilde{\mathcal{E}}(d,p), d \in \mathcal{D}, p \in \mathcal{P}(d). \tag{3.19h}$$

Note that thinning constraint (3.19f) cannot be removed from the formulation as thinning is not guaranteed by constraint (3.19e) due to the general structure of the flow thinning formula. Recall that in case of simple structure (formulation (3.7)) thinning constraint was auxiliary and has been removed, thus leading to a simpler problem formulation.

It turns out that PP for such formulations (3.19) (for different ranges) are more complex than for their counterparts described in this section. Moreover, as the initial formulation (2.2) has been changed, the derivation process (similar to the one described in Section 3.2.1) must be repeated starting all the way from the formulation of the dual to (3.19):

Problem DFT/A/G/$\widetilde{\mathcal{E}}(d,p)$ $(\mathcal{P},\mathcal{S})$:

$$G = \max \textstyle\sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}} h(d,s) \lambda_d^s \tag{3.20a}$$

$$[y_e^0 \geq 0] \quad \pi_e^0 + \textstyle\sum_{s\in\mathcal{S}(e)} \alpha(e,s) \pi_e^s \leq \xi(e), \quad e \in \mathcal{E} \tag{3.20b}$$

$$[x_{dp}^0 \geq 0] \quad \textstyle\sum_{s\in\mathcal{S}} \sigma_{dp}^s \leq \sum_{e\in\mathcal{E}(d,p)} \pi_e^0, \quad d \in \mathcal{D},\ p \in \mathcal{P}(d) \tag{3.20c}$$

$$[x_{dp}^s \geq 0] \quad \lambda_d^s - \varphi_{dp}^s \leq \sigma_{dp}^s + \textstyle\sum_{e\in\mathcal{E}(d,p)\cap\mathcal{E}(s)} \pi_e^s, \quad d \in \mathcal{D},\ p \in \mathcal{P}(d),\ s \in \mathcal{S} \tag{3.20d}$$

$$[z_{dp}^e] \quad \textstyle\sum_{s\in\mathcal{S}} \beta(e,s) \varphi_{dp}^s = 0, \quad d \in \mathcal{D},\ p \in \mathcal{P}(d),\ e \in \widetilde{\mathcal{E}}(d,p) \tag{3.20e}$$

$$[z_{dp}^0] \quad \textstyle\sum_{s\in\mathcal{S}} \varphi_{dp}^s = 0, \ \ d \in \mathcal{D},\ \ p \in \mathcal{P}(d) \tag{3.20f}$$

$$\pi_e^0, \pi_e^s \in \mathbb{R}_+,\ e \in \mathcal{E},\ s \in \mathcal{S}(e); \tag{3.20g}$$

$$\lambda_d^s, \sigma_{dp}^s \in \mathbb{R}_+, \varphi_{dp}^s \in \mathbb{R},\ d \in \mathcal{D},\ p \in \mathcal{P}(d),\ s \in \mathcal{S}. \tag{3.20h}$$

As before, let us fix a demand $d \in \mathcal{D}$, and consider a path $q \in \widehat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$. Adding path q to the current path list $\mathcal{P}$ of the dual (3.20) consists in adding both new dual variables φ^s, $s \in \mathcal{S}$, and σ^s, $s \in \mathcal{S}$, and new constraints (3.20c)-(3.20f). It is possible that for all values of the variables φ^s and σ^s the optimal dual solution (π^*, λ^*) will violate some of the newly added constraints. If that is the case, (π^*, λ^*) will no longer be the optimal solution as it will be separated from the dual polyhedron of DFT/G/S/$\widetilde{\mathcal{E}}(d,p)(\mathcal{P}\cup\{q\},\mathcal{S})$ by the violated constraints. The minimum of the sum of such violations taken over all feasible dual variables $\sigma := (\sigma^s,\ s \in \mathcal{S})$ and $\varphi := (\varphi^s,\ s \in \mathcal{S})$ is expressed through the following quantity (recall that $|q|^0 = \sum_{e\in\mathcal{E}(d,q)} \pi_e^{0*}$ and $|q|^s = \sum_{e\in\mathcal{E}(d,q)\cap\mathcal{E}(s)} \pi_e^{s*}$, $s \in \mathcal{S}$):

$$\begin{aligned} V(q) = \min_{\sigma\geq 0,\varphi} \Big\{ & \max\Big\{\textstyle\sum_{s\in\mathcal{S}} \sigma^s - |q|^0, 0\Big\} + \\ & \textstyle\sum_{s\in\mathcal{S}} \max\Big\{\lambda_d^s - \varphi^s - \sigma^s - |q|^s, 0\Big\} + \\ & \textstyle\sum_{e\in\widetilde{\mathcal{E}}(d,q)} |\sum_{s\in\mathcal{S}} \beta(e,s)\varphi^s| \ + \ |\sum_{s\in\mathcal{S}} \varphi^s|\Big\}. \end{aligned} \tag{3.21a}$$

The following LP formulation can be used to evaluate $V(q)$ for a given path $q \in \hat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$:

$$V(q) = \min\Big\{ Z + \textstyle\sum_{s\in\mathcal{S}} Y_s + \sum_{e\in\widetilde{\mathcal{E}}(d,q)}(X_e' + X_e'') + W + W' \Big\} \tag{3.22a}$$

$$[g \geq 0] \;\; Z \geq \textstyle\sum_{s\in\mathcal{S}} \sigma^s - |q|^0 \tag{3.22b}$$

$$[a_s \geq 0] \;\; Y_s \geq \lambda_d^s - \varphi^s - \sigma^s - |q|^s, \;\; s \in \mathcal{S} \tag{3.22c}$$

$$[b_e' \geq 0] \;\; X_e' \geq \textstyle\sum_{s\in\mathcal{S}} \beta(e,s)\varphi^s, \;\; e \in \widetilde{\mathcal{E}}(d,q) \tag{3.22d}$$

$$[b_e'' \geq 0] \;\; X_e'' \geq -\textstyle\sum_{s\in\mathcal{S}} \beta(e,s)\varphi^s, \;\; e \in \widetilde{\mathcal{E}}(d,q) \tag{3.22e}$$

$$[c' \geq 0] \;\; W' \geq \textstyle\sum_{s\in\mathcal{S}} \varphi^s \tag{3.22f}$$

$$[c'' \geq 0] \;\; W'' \geq -\textstyle\sum_{s\in\mathcal{S}} \varphi^s \tag{3.22g}$$

$$\sigma^s \in \mathbb{R}_+,\ s \in \mathcal{S};\ \varphi^s \in \mathbb{R},\ s \in \mathcal{S}; \tag{3.22h}$$

$$Z, W', W'' \in \mathbb{R}_+;\ Y_s \in \mathbb{R}_+,\ s \in \mathcal{S};\ X_e', X_e'' \in \mathbb{R}_+,\ e \in \widetilde{\mathcal{E}}(d,q). \tag{3.22i}$$

While we should find a $q \in \hat{\mathcal{P}}(d)\setminus\mathcal{P}(d)$ that maximizes $V(q)$, the above formulation cannot be used directly to solve problem $\max_{q\in\widehat{\mathcal{P}}(d)} V(q)$ for the same reason as in Section 3.2.1. Thus, proceeding as in that section, let us formulate the problem dual to (3.22), using the dual variables specified in the square brackets in constraints (3.22b)-(3.22g):

$$V(q) = \max\Big\{ -g|q|^0 + \textstyle\sum_{s\in\mathcal{S}} a_s(\lambda^s - |q|^s) \Big\} \tag{3.23a}$$

$$g \leq 1 \tag{3.23b}$$

$$a_s \leq g, \;\; s \in \mathcal{S} \tag{3.23c}$$

$$b_e' \leq 1,\ b_e'' \leq 1, \;\; e \in \widetilde{\mathcal{E}}(d,q) \tag{3.23d}$$

$$c' \leq 1, \; c'' \leq 1 \tag{3.23e}$$

$$a_s = c' - c'' + \textstyle\sum_{e \in \widetilde{\mathcal{E}}(d,q)} \beta(e,s)(b'_e - b''_e), \quad s \in \mathcal{S} \tag{3.23f}$$

$$g, c', c'' \in \mathbb{R}_+; \; a_s \in \mathbb{R}_+, \; s \in \mathcal{S}; \; b'_e, b''_e \in \mathbb{R}_+, \; e \in \widetilde{\mathcal{E}}(d,q), \tag{3.23g}$$

which is as follows after the elimination of auxiliary variables a_s, $s \in \mathcal{S}$:

$$V(q) = \max \Big\{ -g|q|^0 + $$

$$\textstyle\sum_{s \in \mathcal{S}} (c' - c'' + \sum_{e \in \widetilde{\mathcal{E}}(d,q)} \beta(e,s)(b'_e - b''_e))(\lambda^s - |q|^s) \Big\} \tag{3.24a}$$

$$g \leq 1 \tag{3.24b}$$

$$0 \leq c' - c'' + \textstyle\sum_{e \in \widetilde{\mathcal{E}}(d,q)} \beta(e,s)(b'_e - b''_e) \leq g, \quad s \in \mathcal{S} \tag{3.24c}$$

$$b'_e \leq 1, \; b''_e \leq 1, \quad e \in \widetilde{\mathcal{E}}(d,q) \tag{3.24d}$$

$$c' \leq 1, \; c'' \leq 1 \tag{3.24e}$$

$$g, c', c'' \in \mathbb{R}_+; \; b'_e, b''_e \in \mathbb{R}_+, \; e \in \widetilde{\mathcal{E}}(d,q). \tag{3.24f}$$

Finally, after adding to (3.24) path-defining constraints for undirected links (3.4b)-(3.4c) and isolated-loops-elimination constraints (3.15a)-(3.15e), treating u_e, $e \in \mathcal{E}$ as variables, fixing the range of the flow thinning formula to $\widetilde{\mathcal{E}}(d,p) = \mathcal{E}(d,p)$, and performing proper linearization we obtain the following MILP formulation of the PP for FT/G/S/$\mathcal{E}(d,p)$:

$$V = \max \Big\{ -\textstyle\sum_{e \in \mathcal{E}} \pi^0_e G_e + \sum_{s \in \mathcal{S}} \lambda^s (c' - c'') + \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}} \beta(e,s) \lambda^s (B'_e - B''_e) +$$

$$\textstyle\sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}} \beta(e,s) \lambda^s (B'_e - B''_e) +$$

$$\textstyle\sum_{s \in \mathcal{S}} \sum_{e' \in \mathcal{E}(s)} \sum_{e \in \mathcal{E}} \beta(e,s) \pi^s_{e'} (T''_{e'e} - T'_{e'e}) +$$

$$\textstyle\sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}(s)} \pi^s_e (K''_e - K'_e) \Big\} \tag{3.25a}$$

$$\textstyle\sum_{e \in \delta(v)} u_e = 1, \; v \in \{s(d), t(d)\} \tag{3.25b}$$

$$\textstyle\sum_{e \in \delta(v)} u_e = 2r_v, \; v \in \mathcal{V} \setminus \{s(d), t(d)\} \tag{3.25c}$$

$$g \leq 1 \tag{3.25d}$$

$$0 \leq c' - c'' + \textstyle\sum_{e \in \mathcal{E}} \beta(e,s)(B'_e - B''_e) \leq g, s \in \mathcal{S} \tag{3.25e}$$

$$c' \leq 1, \; c'' \leq 1 \tag{3.25f}$$

$$G_e \leq g, G_e \leq u_e, G_e \geq g + u_e - 1, \quad e \in \mathcal{E} \tag{3.25g}$$

$$B'_e \leq u_e, B'_e \leq b'_e, B'_e \geq b'_e + u_e - 1, \quad e \in \mathcal{E} \tag{3.25h}$$

$$B''_e \leq u_e, B''_e \leq b''_e, B''_e \geq b''_e + u_e - 1, \quad e \in \mathcal{E} \tag{3.25i}$$

$$T'_{e'e} \leq u_{e'}, T'_{e'e} \leq u_e, T'_{e'e} \leq b'_e, T'_{e'e} \geq b'_e + u_{e'} + u_e - 2, \; e', e \in \mathcal{E} \tag{3.25j}$$

$$T''_{e'e} \leq u_{e'}, T''_{e'e} \leq u_e, T''_{e'e} \leq b''_e, T''_{e'e} \geq b''_e + u_{e'} + u_e - 2,\ e', e \in \mathcal{E} \quad (3.25\text{k})$$

$$K'_e \leq c', K'_e \leq u_e, K'_e \geq c' + u_e - 1, \ \ e \in \mathcal{E} \quad (3.25\text{l})$$

$$K''_e \leq c'', K_e \leq u_e, K''_e \geq c'' + u_e - 1, \ \ e \in \mathcal{E} \quad (3.25\text{m})$$

isolated-loops-elimination constraints (3.15a) – (3.15e) (3.25n)

$$B'_e, B''_e, G_e, K'_e, K''_e \in \mathbb{R}_+,\ e \in \mathcal{E} \quad (3.25\text{o})$$

$$T'_{e'e}, T''_{e'e} \in \mathbb{R}_+,\ e', e \in \mathcal{E} \quad (3.25\text{p})$$

$$g, c', c'' \in \mathbb{R}_+;\ b'_e, b''_e \in \mathbb{R}_+,\ e \in \mathcal{E} \quad (3.25\text{q})$$

$$u_e \in \mathbb{B},\ e \in \mathcal{E};\ r_v \in \mathbb{B},\ v \in \mathcal{V} \setminus \{s(d), t(d)\}. \quad (3.25\text{r})$$

In the above formulation, the extra variables that are used to eliminate the products (bi-linearities and tri-linearities) of the original variables have the following meaning: $G_e = gu_e$, $B'_e = b'_e u_e$, $B''_e = b''_e u_e$, $T'_{e'e} = b'_e u_e u_{e'}$, $T''_{e'e} = b''_e u_e u_{e'}$, $K'_e = c' u_e$, $K''_e = c'' u_e$. Note that constraints (3.24d) are intentionally missing in (3.25) because they transform into $B'_e \leq 1$, $B''_e \leq 1$, $e \in \mathcal{E}$ after adding u_e variables and thus these constraints become redundant as they are guaranteed by constraints (3.25h)-(3.25i).

Similarly to simple structure variants, PP formulations for the two remaining cases of the flow thinning formula range are analogous to (3.25). As before, the PP formulation for $\widetilde{\mathcal{E}}(d,p) = \mathcal{E}^+(d,p)$ is more complex (and more time consuming as far as problem solving computation time is concerned) while the PP formulation for $\widetilde{\mathcal{E}}(d,p) = \mathcal{E}$ is less complex (and less time consuming) than (3.25).

3.2.6 PP for FT/Q/S/$\widetilde{\mathcal{E}}(d,p)$

The formulations of pricing problems for the FT formulae containing the quadratic terms as in the general formula FT/Q/G/$\widetilde{\mathcal{E}}(d,p)$ (2.4) or its counterpart with simple structure FT/Q/S/$\widetilde{\mathcal{E}}(d,p)$ can be derived analogously as for the affine variants. Yet, the resulting PP formulations require much more variables and become computationally inefficient. For example, products of the form $b_{e''} u_{e''} u_{e'} u_e$ appearing in the non-linear formulations would have to be eliminated, and this would involve $|\mathcal{E}|^3$ auxiliary variables. To resolve this issue and at the same time keep the advantages of the quadratic form, in the numerical study we simply use the sets of paths generated for the affine counterparts of the quadratic problem in question (see Section 4.1.4).

3.3 Solution for state polytope

Let us consider the FT/A/S/$\mathcal{E}(d,p)$ variant of the problem (see formulation (2.6)) as an example and solve it for the set of states $\widehat{\mathcal{B}}(N,a,b,H)$, i.e., for all states of the form $s(u)$, $u \in \mathcal{B}(N)$. This, as shown below, can be achieved through an iterative procedure based on the state and path generation algorithms.

3.3.1 Feasibility tests

Consider a given state list $\mathcal{S}$ being a sub-list of $\widehat{\mathcal{B}}(N,a,b,H)$, and let $(y^0(\mathcal{S}), x^0(\mathcal{S}), z(\mathcal{S}))$ be a feasible solution of problem (2.6) for $\mathcal{S}$. Clearly, in order to see whether this solution is feasible for all states in $\widehat{\mathcal{B}}(N,a,b,H)$ one can check if the following constraints (corresponding to constraints (2.6c), (2.6d), and (2.6e), respectively) are satisfied for each $u \in \mathcal{B}(N)$:

$$h(d,u) - \textstyle\sum_{p\in\mathcal{P}(d)} x_{dp}(u) \le 0, \quad d \in \mathcal{D} \tag{3.26a}$$

$$\textstyle\sum_{d\in\mathcal{D}} \sum_{p\in\mathcal{R}(d,e)} x_{dp}(u) - \alpha(e,u) y_e^0(\mathcal{S}) \le 0, \quad e \in \mathcal{E} \tag{3.26b}$$

$$-x_{dp}(u) \le 0, \quad d \in \mathcal{D},\ p \in \mathcal{P}(d). \tag{3.26c}$$

Note that flows $x^s(\mathcal{S})$ in (3.26) are determined by $x^0(\mathcal{S})$ and $z(\mathcal{S})$ through equations (2.6e): $x_{dp}(u) := x^0_{dp}(\mathcal{S}) - \sum_{e\in\mathcal{E}(d,p)} \beta(e,u) z^e_{dp}(\mathcal{S})$, for all $d \in \mathcal{D}$, $p \in \mathcal{P}(d)$, $u \in \mathcal{B}(N)$.

Satisfiability of constraints (3.26a)-(3.26c) can be checked separately by formulating a corresponding binary program (called *feasibility test*) of maximizing the left-hand side of the appropriate inequality in (3.26) over $\mathcal{B}(N)$, where $\beta(e,u) = 1 - \alpha(e,u)$, and $\alpha(e,u)$ and $h(d,u)$ are expressed with u as follows (cf. (2.9) and (2.10)):

$$\alpha(e,u) := \textstyle\sum_{k\in\mathcal{K}} a(k) u_e^k, \quad e \in \mathcal{E} \tag{3.27a}$$

$$h(d,u) := H(d)\Big(1 - \textstyle\sum_{k\in\mathcal{K}} b(k) \sum_{e\in\mathcal{E}} u_e^k\Big), \quad d \in \mathcal{D}. \tag{3.27b}$$

The so defined feasibility tests can be written in a concise form as:

$$T(d) = \max_{u\in\mathcal{B}(N)} \Big\{ h(d,u) - \textstyle\sum_{p\in\mathcal{P}(d)} x_{dp}(u) \Big\}, \quad d \in \mathcal{D} \tag{3.28a}$$

$$T(e) = \max_{u\in\mathcal{B}(N)} \Big\{ \textstyle\sum_{d\in\mathcal{D}} \sum_{p\in\mathcal{R}(d,e)} x_{dp}(u) - \alpha(e,u) y_e^0(\mathcal{S}) \Big\}, \quad e \in \mathcal{E} \tag{3.28b}$$

$$T(d,p) = \max_{u\in\mathcal{B}(N)} \Big\{ - x_{dp}(u) \Big\}, \quad d \in \mathcal{D},\ p \in \mathcal{P}(d). \tag{3.28c}$$

Let us consider test (3.28a) for some $d \in \mathcal{D}$ and let u^* be an optimal solution for the test. Clearly, if the value $T(d)$ obtained from this test is positive, then $\sum_{p\in\mathcal{P}(d)} x_{dp}(u^*) <$

$h(d, u^*)$, which means that demand d is not satisfied by the considered feasible solution $y^0(\mathcal{S})$, $x^0(\mathcal{S})$, $z(\mathcal{S})$ of FT/A/S/$\mathcal{E}(d,p)(\mathcal{P},\mathcal{S})$. The same holds for the other two tests: if $T(e) > 0$, then $\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{R}(d,e)} x_{dp}(u^*) > \alpha(e,u^*)y_e^0(\mathcal{S})$ (i.e., link e is overflowed); and if $T(d,p) > 0$, then $x_{dp}(u^*) < 0$ (i.e., the flow value is infeasible).

Thus, if either of the conditions

$$\max_{d\in\mathcal{D}} T(d) > 0 \;\text{ or }\; \max_{e\in\mathcal{E}} T(e) > 0 \;\text{ or }\; \max_{d\in\mathcal{D},\, p\in\mathcal{P}(d)} T(d,p) > 0 \tag{3.29}$$

holds, then the solution $y^0(\mathcal{S})$, $x^0(\mathcal{S})$, $z(\mathcal{S})$ is infeasible for the set of states $\widehat{\mathcal{B}}(N,a,b,H)$ (and *vice versa*).

The feasibility tests (3.28) can be turned onto LP problems since, according to Property 2.1, the binary set of feasible solutions $\mathcal{B}(N)$ can simply be substituted with its continuous counterpart $\mathcal{Q}(N)$. The appropriate LP problem formulations corresponding to feasibility tests (3.28a)-(3.28c) are given below. In them, N, a, b, H are vectors of parameters describing state polytope $\mathcal{Q}(N)$ (see Section 2.3.1); $y^0(\mathcal{S})$, $x^0(\mathcal{S})$, $z(\mathcal{S})$ are vectors of parameters obtained from solving FT/A/S/$\mathcal{E}(d,p)$ for a given state list $\mathcal{S}$; and u, U, α, β, h, x are vectors of variables.

Problem FEAS-TEST-DEMAND $(y^0(\mathcal{S}), x^0(\mathcal{S}), z(\mathcal{S}), d)$**:**

$$T(d) = \max\left\{h_d - \textstyle\sum_{p\in\mathcal{P}(d)} x_{dp}\right\} \tag{3.30a}$$

$$\textstyle\sum_{k\in\mathcal{K}} u_e^k = 1, \quad e\in\mathcal{E} \tag{3.30b}$$

$$\textstyle\sum_{e\in\mathcal{E}} u_e^k \le N(k), \quad k\in\mathcal{K} \tag{3.30c}$$

$$U^k = \textstyle\sum_{e\in\mathcal{E}} u_e^k, \quad k\in\mathcal{K} \tag{3.30d}$$

$$\beta_e = 1 - \textstyle\sum_{k\in\mathcal{K}} a(k)u_e^k, \quad e\in\mathcal{E} \tag{3.30e}$$

$$h_d = H(d)(1 - \textstyle\sum_{k\in\mathcal{K}} b(k)U^k) \tag{3.30f}$$

$$x_{dp} = x_{dp}^0(\mathcal{S}) - \textstyle\sum_{e\in\mathcal{E}(d,p)} \beta_e z_{dp}^e(\mathcal{S}), \quad p\in\mathcal{P}(d) \tag{3.30g}$$

$$u_e^k, U^k, \beta_e \in \mathbb{R}_+,\ e\in\mathcal{E},\ k\in\mathcal{K}; \tag{3.30h}$$

$$h_d \in \mathbb{R}_+;\ \ x_{dp}\in\mathbb{R},\ p\in\mathcal{P}(d), \tag{3.30i}$$

Problem FEAS-TEST-LINK $(y^0(\mathcal{S}), x^0(\mathcal{S}), z(\mathcal{S}), e)$**:**

$$T(e) = \max\left\{\textstyle\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{R}(d,e)} x_{dp} - \alpha_e y_e^0(\mathcal{S})\right\} \tag{3.31a}$$

$$\textstyle\sum_{k\in\mathcal{K}} u_{e'}^k = 1, \quad e'\in\mathcal{E} \tag{3.31b}$$

$$\textstyle\sum_{e'\in\mathcal{E}} u_{e'}^k \le N(k), \quad k\in\mathcal{K} \tag{3.31c}$$

$$\alpha_e = \textstyle\sum_{k\in\mathcal{K}} a(k)u_e^k \tag{3.31d}$$

$$\beta_{e'} = 1 - \textstyle\sum_{k\in\mathcal{K}} a(k)u^k_{e'}, \quad e' \in \mathcal{E} \tag{3.31e}$$

$$x_{dp} = x^0_{dp}(\mathcal{S}) - \textstyle\sum_{e'\in\mathcal{E}(d,p)} \beta_{e'} z^{e'}_{dp}(\mathcal{S}), \quad d \in \mathcal{D},\ p \in \mathcal{P}(d) \tag{3.31f}$$

$$u^k_{e'}, \beta_{e'} \in \mathbb{R}_+,\ e' \in \mathcal{E},\ k \in \mathcal{K}; \tag{3.31g}$$

$$\alpha_e \in \mathbb{R}_+;\ \ x_{dp} \in \mathbb{R},\ d \in \mathcal{D},\ p \in \mathcal{P}(d), \tag{3.31h}$$

Problem FEAS-TEST-FLOW $(y^0(\mathcal{S}), x^0(\mathcal{S}), z(\mathcal{S}), d, p)$**:**

$$T(d,p) = \max \Big\{ - x_{dp} \Big\} \tag{3.32a}$$

$$\textstyle\sum_{k\in\mathcal{K}} u^k_e = 1, \quad e \in \mathcal{E} \tag{3.32b}$$

$$\textstyle\sum_{e\in\mathcal{E}} u^k_e \le N(k), \quad k \in \mathcal{K} \tag{3.32c}$$

$$\beta_e = 1 - \textstyle\sum_{k\in\mathcal{K}} a(k)u^k_e, \quad e \in \mathcal{E} \tag{3.32d}$$

$$x_{dp} = x^0_{dp}(\mathcal{S}) - \textstyle\sum_{e\in\mathcal{E}(d,p)} \beta_e z^e_{dp}(\mathcal{S}) \tag{3.32e}$$

$$u^k_e, \beta_e \in \mathbb{R}_+,\ e \in \mathcal{E},\ k \in \mathcal{K};\ \ x_{dp} \in \mathbb{R}. \tag{3.32f}$$

3.3.2 State Generation algorithm

The algorithm for solving FT/A/S/$\mathcal{E}(d,p)$ (for now, for a fixed path list $\mathcal{P}$) by state generation based on the introduced feasibility tests is given below.

Algorithm 2 SGA($\mathcal{S}_0$)

```
iter ← 0
repeat
    (y⁰(S_iter), x⁰(S_iter), z(S_iter)) ← FT/A/S/E(d,p)(S_iter)
    S_iter+1 ← S_iter
    for d ∈ D do
        (u(d), T(d)) ← FEAS-TEST-DEMAND(y⁰(S_iter), x⁰(S_iter), z(S_iter), d)
        if T(d) > 0 then
            S_iter+1 ← S_iter+1 ∪ {s(u(d))}
        end if
    end for
    for e ∈ E do
        (u(e), T(e)) ← FEAS-TEST-LINK(y⁰(S_iter), x⁰(S_iter), z(S_iter), e)
        if T(e) > 0 then
            S_iter+1 ← S_iter+1 ∪ {s(u(e))}
        end if
```

16: **end for**

17: **for** $d \in \mathcal{D}$ **do**

18: **for** $p \in \mathcal{P}(d)$ **do**

19: $(u(d,p), T(d,p)) \leftarrow$ FEAS-TEST-FLOW$(y^0(\mathcal{S}_{iter}), x^0(\mathcal{S}_{iter}), z(\mathcal{S}_{iter}), d, p)$

20: **if** $T(d,p) > 0$ **then**

21: $\mathcal{S}_{iter+1} \leftarrow \mathcal{S}_{iter+1} \cup \{s(u(d,p))\}$

22: **end if**

23: **end for**

24: **end for**

25: $iter \leftarrow iter + 1$

26: **until** $\mathcal{P}_{iter} = \mathcal{P}_{iter-1}$

Algorithm 2 requires an initial state list $\mathcal{S}_0$ as an input argument, for example a reasonable subset of $\widehat{\mathcal{B}}(N,a,b,H)$. However, it is also possible to consider selected states outside of $\widehat{\mathcal{B}}(N,a,b,H)$ by simply putting them on the initial state list. The main loop (lines 2-26) starts in line 3 from solving FT/A/S/$\mathcal{E}(d,p)$ (see formulation (2.6)) for the initial state list and obtaining its optimal solution ($y^0(\mathcal{S}_0)$, $x^0(\mathcal{S}_0)$, $z(\mathcal{S}_0)$) (here and in what follows the superscript * that indicates optimality is skipped). Next, in lines 5-10 for each demand $d \in \mathcal{D}$ feasibility test (3.30) is solved, and its optimal solution is stored as ($u(d)$, $T(d)$). In case of the positive result of the test (i.e., if $T(d) > 0$), the obtained state $s(u(d))$ is added to the state list for the next main iteration $\mathcal{S}_{iter+1}$. In a similar way the state list $\mathcal{S}_{iter+1}$ can potentially be extended by solving feasibility test (3.31) in lines 11-16 and feasibility test (3.32) in lines 17-24. If at least one state has been added to the state list $\mathcal{S}_{iter+1}$, the main loop is repeated for the extended state list. Otherwise, the algorithm stops (after some n iterations) and the current vectors $y^0(\mathcal{S}_n)$, $x^0(\mathcal{S}_n)$, $z(\mathcal{S}_n)$ computed in line 3 form an optimal solution of FT/A/S/$\mathcal{E}(d,p)(\mathcal{P},\mathcal{S}_n)$, where $\mathcal{S}_n$ contains all the states from $\widehat{\mathcal{B}}(N,a,b,H)$ and the states from the initial state list outside $\widehat{\mathcal{B}}(N,a,b,H)$ (if any).

3.3.3 Combination of state generation and path generation

In order to find the true minimum of FT/A/S/$\mathcal{E}(d,p)$ (when all paths are considered) for a given state polytope $\widehat{\mathcal{B}}(N,a,b,H)$, one can combine the state generation algorithm SGA with the path generation algorithm PGA. The resulting iterative procedure, referred to as the SGA+PGA algorithm, is as follows.

Algorithm 3 SGA+PGA($\mathcal{S}_0, \mathcal{P}_0$)

1: $\mathcal{P}_0 \leftarrow \mathcal{P}_0 \cup \mathrm{PGA}(\mathcal{S}_0, \mathcal{P}_0)$
2: $iter \leftarrow 0$
3: **while** true **do**
4:     $\mathcal{S}_{iter+1} \leftarrow \mathcal{S}_{iter} \cup \mathrm{SGA}(\mathcal{S}_{iter}, \mathcal{P}_{iter})$
5:     **if** $\mathcal{S}_{iter} = \mathcal{S}_{iter+1}$ **then**
6:         **break**
7:     **end if**
8:     $\mathcal{P}_{iter+1} \leftarrow \mathcal{P}_{iter} \cup \mathrm{PGA}(\mathcal{S}_{iter+1}, \mathcal{P}_{iter})$
9:     **if** $\mathcal{P}_{iter} = \mathcal{P}_{iter+1}$ **then**
10:         **break**
11:     **end if**
12:     $iter \leftarrow iter + 1$
13: **end while**

The above algorithm takes an initial state list $\mathcal{S}_0$ and an initial path list $\mathcal{P}_0$ as input arguments. The algorithm starts from preliminary extension of the initial path list by solving PGA. In the main loop (lines 3-13) first SGA is solved for the current state and path lists. In consequence, state list $\mathcal{S}_{iter+1}$ is in general extended with the new states returned by SGA. The algorithm stops in case no new states have been added. Next, in a similar way PGA is solved for the (potentially extended) state list and current path list. In consequence, path list $\mathcal{P}_{iter+1}$ is in general extended with the new paths returned by PGA. Similarly, the algorithm stops in case no new paths have been added.

Despite the fact that the while loop in line 3 is of infinite type, the algorithm will always stop in either line 6 or line 10 after a finite number of steps since in each iteration at least one path and one state are added, and the number of paths and states are finite. Moreover, although both the number of paths and the number of states grow exponentially with the size of the network, a polynomial number of iterations can be expected, just like when using the simplex algorithm for non-compact linear problems.

3.3.4 Extension of state generation algorithm to quadratic FT

So far, solving the problem based on state generation algorithm and feasibility tests was presented for a chosen FT formula, namely FT/A/S/$\mathcal{E}(d,p)$. Solving the problem for the FT formula variants with affine form and other ranges and structures is not specifically different, it only necessitates some accurate adjustments in the feasibility tests. However, solving the problem for the FT formula variant with quadratic form requires separate attention, as certain terms of the objective functions of the feasibility test problems become bi-linear. More precisely, since the path flows for a given point $u \in \mathcal{Q}(N)$ are expressed as

$$x_{dp}(u) = x^0_{dp}(\mathcal{S}) - \textstyle\sum_{e\in\mathcal{E}(d,p)} \beta(e,u) z^e_{dp}(\mathcal{S}) + \textstyle\sum_{\{e,e'\}\in\mathcal{E}^{|2|}(d,p)} \beta(e,u)\beta(e',u) z^{ee'}_{dp}(\mathcal{S}),$$

products of the optimization variables composing vector u (of the form $u^k_e u^{k'}_{e'}$) will appear, because

$$\beta(e,u) = 1 - \textstyle\sum_{k\in\mathcal{K}} a(k) u^k_e \;\text{ and }\; \beta(e',u) = 1 - \textstyle\sum_{k'\in\mathcal{K}} a(k') u^{k'}_{e'},$$

and thus

$$\begin{aligned}\beta(e,u)\beta(e',u) &= (1 - \textstyle\sum_{k\in\mathcal{K}} a(k) u^k_e)(1 - \textstyle\sum_{k'\in\mathcal{K}} a(k') u^{k'}_{e'}) = \\ &= 1 - \textstyle\sum_{k\in\mathcal{K}} a(k) u^k_e - \textstyle\sum_{k'\in\mathcal{K}} a(k') u^{k'}_{e'} + \textstyle\sum_{k,k'\in\mathcal{K}} a(k)a(k') u^k_e u^{k'}_{e'}.\end{aligned}$$

Using the above formulae, the linear (bi-linear) formulations of the feasibility tests (3.28) adjusted for quadratic FT can be easily obtained from formulations (3.30)-(3.32). In the so obtained adjustments, auxiliary variables x_{dp} can be moved to the respective objective functions; in effect, the adjusted tests involve formulations in continuous variables with quadratic (in general neither convex nor concave) objective functions and linear constrains, and as such can be treated by a quadratic programming solver. It turns out, however, that this would lead to an excessive number of SGA iterations. Therefore, it is more efficient to use a Mixed-Integer Quadratic Programming (MIQP) solver for the modified feasibility tests with the state-polytope-defining variables u assumed to be binary – this version of SGA is used in the numerical study in Chapter 4.

Chapter 4

Numerical study

This chapter discusses the results of a numerical study of the optimization problems and solution algorithms presented in Chapters 2-3. The study consists of three parts. The first part (Section 4.1) illustrates the efficiency of path generation algorithm. The second part (Section 4.2) is devoted to SGA+PGA. The third part (Section 4.3) summarizes the cost efficiency of the FT mechanisms.

All reported computations were executed on a PC-class computer (Windows 10 64-bit, 8 GB RAM, Processor Intel Core i5-3210M, 4 logical processors, 2.5GHz) using CPLEX [12] optimization software package (version 12.4.0.0 in the first part of the numerical study and 12.8.0.0 in the second part).

4.1 Path generation algorithm study

In the first part of the numerical study we analyze computational efficiency (in terms of the computation time) of the PGA and related pricing problems described in Section 3.1. We also analyze traffic efficiency (in terms of the network cost) of different variants of the FT formula described in Section 2.2.

4.1.1 Network instance

This part of the study was carried out for the network described in the SNDlib library of communication network instances (sndlib.zib.de, see [44]) under the name *polska*. The network, depicted in Figure 4.1, is composed of $|V| = 12$ nodes, $|E| = 18$ (undirected) links, and $|D| = 66$ (undirected) demands. In the study we used the traffic data from

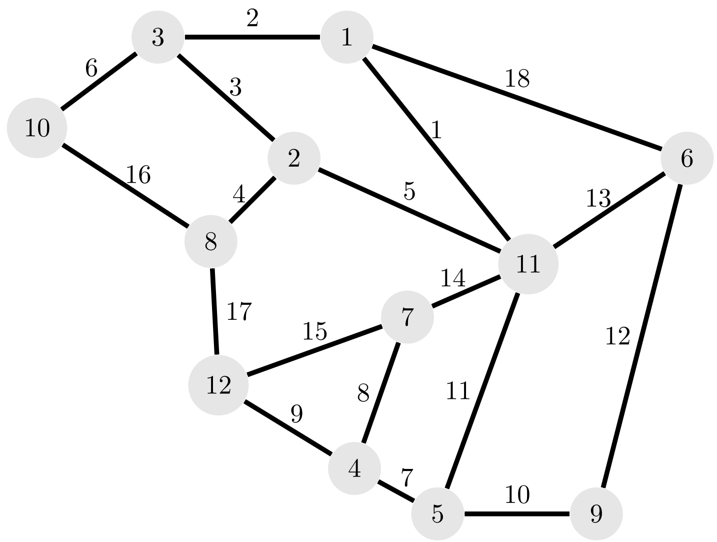

Figure 4.1: *polska* network topology.

the first *polska* instance (i.e., *polska*–D-B-M-N-C-A-N-N) stored in SNDlib. The assumed link unit capacity cost $\xi(e)$ was computed as the link's <module_cost> divided by <module_capacity> for the first pair of these values. Thus, the entire vector ξ of the unit link capacity costs is as follows: 1.0645, 1.7548, 1.0064, 1.2000, 1.7548, 1.5290, 1.3419, 1.1677, 1.3419, 1.6129, 2.0903, 2.0903, 1.6129, 1.0645, 1.9677, 0.9161, 1.2581, 1.8968 (the order of the links is the same as in the SNDlib file).

We consider the following three link availability state scenarios:

(SL) Single link degradation scenario: SL contains the nominal state $s(0)$ (all links fully available), and all states with exactly one degraded link (note that SL contains $|\mathcal{E}|+1$ states). The traffic demands in state $s(0)$, $h(d,s(0))$, $d \in \mathcal{D}$, are specified in SNDlib. In SL, 100% traffic protection is assumed, i.e., $h(d,s) = h(d,s(0))$, $d \in \mathcal{D}$, for each single link degradation state s. Link degradation coefficients of the affected links are assumed to be $\beta(e,s) = 0.5$.

(DL) Double link degradation scenario: DL contains SL and all states with exactly two degraded links (hence DL contains $|\mathcal{E}|(|\mathcal{E}|-1)/2+|\mathcal{E}|+1$) states). Link degradation coefficients in the double link degradation states are assumed to be $\beta(e,s) = 0.4$. For the double link degradation states the demand volumes are reduced to 95% of $h(d,s(0))$ for each $d \in \mathcal{D}$.

(TL) Triple link degradation scenario: TL contains SL and DL, and all states with exactly three degraded links (TL contains $|\mathcal{E}|(|\mathcal{E}|-1)(|\mathcal{E}|-2)/6 + |\mathcal{E}|(|\mathcal{E}|-1)/2 + |\mathcal{E}| +$

1 states). Link degradation coefficients of the degraded links in the triple link degradation states are assumed to be $\beta(e, s) = 0.3$ and the demand volumes are reduced to 90% of $h(d, s(0))$ for each $d \in \mathcal{D}$.

The assumed form of the state scenarios is convenient for studying computational efficiency of the PGA-based optimization process for mesh network topologies. The scenarios are easy to generate, do not refer to any particular network operating conditions, and are sufficiently demanding for comparing PGA's computational efficiency for different variants of the FT formula. In consequence, although the *polska* instance corresponds to a long-distance optical transmission network, and not, for example, to a particular FSO-based MAN (for which FT is destined), its tractable size, mesh topology similar to MAN, and description available in SNDlib makes it suitable for the conducted experiments. It provides a representative (for a medium size MAN) illustration for quantitative relations between computational times for different FT variants (that otherwise can be qualitatively deduced from the corresponding master and pricing problems formulations).

4.1.2 Results for UR and FT

Table 4.1 compares results obtained by means of the PG algorithm for UR and FT. Recall that UR (see [43, 56]) is a traffic protection mechanism concept that allows restoring the traffic demands from scratch using the entire capacity of links available in a given network link availability state. Although UR is hardly implementable in practice, it serves as a benchmark mechanism since it is the least constrained mechanism and thus the cost of the network protected by means of UR provides a lower bound for any other protection mechanism. The results are obtained for FT by solving problem (2.2) and for UR by solving problem (2.1).

For each state scenario (SL, DL, TL), the rows in Table 4.1 contain the results for UR and FT, and the consecutive columns describe:

- F^0: cost of unprotected network dimensioned for $h(d, s(0))$, $d \in \mathcal{D}$,
- F^*: cost of the optimal solution resulting from the PGA,
- ΔF^0: cost increase with respect to unprotected network ($\Delta F^0 = \frac{F^* - F^0}{F^0} \times 100\%$),
- ΔF^*: cost increase of FT with respect to UR ($\Delta F^* = \frac{F^*(FT) - F^*(UR)}{F^*(UR)} \times 100\%$),

– $|\mathcal{P}^*|$: final number of paths in tha path list,

– $|\mathcal{P}^u|$: number of paths used in the final solution (i.e., paths with strictly positive optimal value of x^0_{dp}),

– number of iterations (iter), total computation time (total), computation time per iteration (t/iter), computation time spent in the MP (master problem) per iteration (t/MP), and pricing problem computation time per iteration (t/PP).

Table 4.1: Results for UR and FT.

		F^0	F^*	ΔF^0[%]	ΔF^*[%]	$\|\mathcal{P}^*\|$	$\|\mathcal{P}^u\|$	iter	total	t/iter	t/MP	t/PP
SL	UR	30275	35858	18	–	189	149	3	2s	0.7s	0.1s	0.6s
	FT	30275	40236	33	12	259	123	9	6s	0.6s	0.1s	0.5s
DL	UR	30275	38087	26	–	235	217	3	23s	7.6s	3s	4.6s
	FT	30275	40093	32	5	269	143	8	1m12s	8.9s	5.5s	3.4s
TL	UR	30275	36630	21	–	249	249	4	4m13s	1m3s	45s	18s
	FT	30275	37200	23	2	294	156	8	24m3s	3m	2m28s	32s

The results show that the lower bound for the cost of protection (achieved with UR) for the considered network instance is between 18% (SL) and 26% (DL). Certainly, FT requires more capacity, but the cost increase as compared to UR is not significant (between 2% for TL and 12% for SL). The number of generated paths is lower for UR than for FT, but the number of the paths used in the final (optimal) solution is higher (which could be expected). For both UR and FT the computation times are negligible for SL and low for DL. For TL, the algorithm applied to UR is still quite fast, while for FT it starts to be more time consuming, both for MP and for PP. We note here that the PG algorithm for UR works differently than for FT (and, for that matter, for affine FT and quadratic FT): in the former case in each iteration a new path is considered for each demand $d \in \mathcal{D}$ and each state $s \in \mathcal{S}$ (see [25]), while for FT only one demand path is considered. This is why UR requires less iterations than FT. Yet, even for FT the number of iterations is small and the PG algorithm converges very quickly.

4.1.3 Results for affine FT

Table 4.2 gives the results for FT/A, i.e., for FT with the affine thinning formula. The results for FT/A are obtained by solving problem (2.6) with the thinning formula (2.6e) replaced by its appropriate variant taken from Table 2.1. For each state scenario, all six combinations of the range and the structure considered for the flow thinning formula of form A (see Section 2.2.4) are examined (and specified in columns "range" and "structure"). Now, column "ΔF^*" expresses the increase of the network cost for a given variant of FT/A with respect to the corresponding solution F^* of FT given in Table 4.1. The meaning of the remaining columns is the same as before.

Regarding the network cost F^* achievable by the considered FT/A formula variants, we first observe that in the SL case, the minimal cost (i.e., the cost of FT) is achieved already with the simplest form of the FT/A formula, that is with the FT/A/S/$\mathcal{E}(d,p)$ variant. The reason is as follows. Suppose ${x^0_{dp}}^*$ and ${x^s_{dp}}^*$ are optimal for FT. Then the flow thinning formula

$$x^s_{dp} = x^0_{dp} - \textstyle\sum_{e\in\mathcal{E}(d,p)} \beta(e,s) z^e_{dp}, \text{ where } z^e_{dp} := \frac{{x^0_{dp}}^* - {x^{s(e)}_{dp}}^*}{\beta(e,s)},\ e \in \mathcal{E}(d,p),$$

will give the optimal values of x^s_{dp}, that is those found for FT. In the formula, $s(e)$ denotes the particular state $s \in \mathcal{S}$ in which link e is affected with the degradation coefficient $\beta(e, s(e)) > 0$ (recall that in the considered example $\beta(e, s(e)) = 0.5$); in the remaining states, i.e., for each $s \in \mathcal{S} \setminus \{s(e)\}$, $\beta(e,s) = 0$.

For DL and TL the cost is no longer minimal. For range $\mathcal{E}(d,p)$ the increase of F^* with repsect to the cost of FT, given by ΔF^*, is quite high and is the same for both structures G (general) and S (simple) – it is equal 18% for DL and 13% for TL. For the two other ranges $\mathcal{E}^+(d,p)$ and $\mathcal{E}$ the difference between G and S becomes visible. For $\mathcal{E}^+(d,p)$ the considered increase for G equals 7% and is roughly 2 times smaller than for S, while for $\mathcal{E}$ it is only 4% and is 3.25 (TL) to 3.6 (DL) times smaller then for S.

Note that for $\mathcal{E}^+(d,p)$ Table 4.2 reports just one computation time (t/MP). The reason is that this particular range case was not treated by the path generation algorithm because of the excessive computation time (of the order of hours) required for the pricing problem. Thus, instead of using path generation we solved the appropriate variant of the master problem (2.6) once using the path list $\mathcal{P}^*$ generated for $\mathcal{E}$. For the same reason the number

Table 4.2: Results for FT/A.

	range	structure	F^*	$\Delta F^*[\%]$	$\lvert\mathcal{P}^*\rvert$	$\lvert\mathcal{P}^u\rvert$	iter	total	t/iter	t/MP	t/PP
SL	$\mathcal{E}(d,p)$	G	40236	0	275	125	8	8m2s	1m	1s	59s
		S	40236	0	276	123	9	2m22s	16s	0.2s	15.8s
	$\mathcal{E}^+(d,p)$	G	40236	0	283	128	–	0.3s	–	–	–
		S	40236	0	274	124	–	0.3s	–	–	–
	$\mathcal{E}$	G	40236	0	283	125	9	6m28s	43s	0.1s	42.9s
		S	40236	0	274	124	9	2m3s	14s	0.2s	13.8s
DL	$\mathcal{E}(d,p)$	G	47174	18	285	161	9	10m13s	1m8s	3s	1m5s
		S	47174	18	283	161	9	4m31s	30s	2s	28s
	$\mathcal{E}^+(d,p)$	G	43543	9	216	159	–	40s	–	–	–
		S	47174	18	187	161	–	5s	–	–	–
	$\mathcal{E}$	G	42172	5	216	155	8	13m18s	1m39s	25s	1m14s
		S	47174	18	187	161	7	2m21s	20s	1.6s	18.4s
TL	$\mathcal{E}(d,p)$	G	41863	13	232	161	10	25m36s	3m34s	27s	3m7s
		S	41863	13	223	162	9	12m37s	1m24s	18s	1m6s
	$\mathcal{E}^+(d,p)$	G	39986	7	191	145	–	4m20s	–	–	–
		S	41863	13	165	160	–	1m6s	–	–	–
	$\mathcal{E}$	G	38707	4	191	133	7	32m39s	4m39s	2m1s	2m38s
		S	41863	13	165	160	6	6m27s	1m5s	20s	45s

of iterations is not reported in column "iter". Although this makes the optimization approach heuristic, the obtained solutions are clearly near-optimal.

The number of paths generated by PGA (column "$|\mathcal{P}^*|$") and the number of paths used in the optimal solution (column "$|\mathcal{P}^u|$") do not exhibit any particular properties, except that sometimes $|\mathcal{P}^u|$ is considerably smaller than $|\mathcal{P}^*|$ (even more than two times for SL and $\mathcal{E}(d,p)$).

The number of iterations performed by the PG algorithm varies from 6 to 10 and this is a reasonable number indicating fast convergence. The computation time spent solving PP (pricing problem) is also reasonable although pricing new paths requires solving MILP problems. The time spent solving MP (master problem) is typically smaller than the time required for pricing. Certainly, the computation time increases with the number of states in $\mathcal{S}$, but even for TL the total computation times are acceptable.

It is also worth noting that the case FT/A/S gives the same value of F^* for all the three ranges, and this means that the solution for $\mathcal{E}^+(d,p)$ is optimal, as it is always less than or equal to the optimal solution for $\mathcal{E}(d,p)$, and greater than or equal to the optimal solution for $\mathcal{E}$. Let us also note that in all the considered scenarios the costs of FT/A/S/$\mathcal{E}(d,p)$ and FT/A/G/$\mathcal{E}(d,p)$ happen to be equal. Yet, this is not always true. We have found some (randomly generated) state scenarios where the costs for these two cases are different. The results for these state scenarios are presented in Table 4.3. The first three columns in the table describe the randomly generated state scenarios: in each state $s \in \mathcal{S}$, each link $e \in \mathcal{E}$ is affected with degradation coefficient "β" with probability "prob", and remains fully available otherwise. Traffic demands are kept at the nominal level: $h(d,s) = h(d)$. Still, although existent, the difference between the simple and general structure solutions is small: of the order of 1.5%.

Table 4.3: Results for FT/A/S/$\mathcal{E}(d,p)$ and FT/A/G/$\mathcal{E}(d,p)$ on random states.

$\lvert\mathcal{S}\rvert$	prob	β	structure	F^*	ΔF	total
10	0.25	0.4	G	44512	–	17m23s
			S	45213	1.6%	5m5s
10	0.5	0.5	G	57852	–	26m47s
			S	58824	1.7%	6m31s

4.1.4 Results for quadratic FT

Table 4.4 shows the results for FT/Q, i.e., for FT with the quadratic flow thinning formula. As for the FT/A case, all six combinations of range/structure (see Section 2.2.4) are considered for each state scenario. All the FT/Q cases were directly optimized through solving an appropriate version of the master problem (2.6) for the sets of paths obtained for the corresponding FT/A cases – this is due to excessive pricing time for FT/Q. (Therefore, Table 4.4 has less columns as compared with Table 4.2.) This near-optimal procedure has already been applied for the $\mathcal{E}^+(d,p)$ range of the FT/Q.

Clearly, the FT/Q cost values given in Table 4.4 for SL are the same as for FT/A (and, for that matter, for FT), since FT/A is a special case of FT/Q and the costs for FT/A and FT are the same (as explained in Section 4.1.3). Note that the corresponding total solution times are much shorter since no pricing is involved.

However, for DL and TL, the cost of the FT/Q solutions is considerably smaller than for the corresponding FT/A solutions. In fact, the case FT/Q/G/$\mathcal{E}$ (general quadratic form with full range) indicates virtually the same cost as FT (arbitrary flow thinning): the cost increase equals 0.03% for DL, and 0.2% for TL. At the same time, the simplified quadratic form with full range (FT/Q/S/$\mathcal{E}$) is only marginally worse: cost increase 0.5% for DL, and 1.3% for TL. Compared to $\mathcal{E}$, the costs obtained with range $\mathcal{E}^+(d,p)$ are not much larger. For DL the cost increase is 2.6% for FT/Q/G and 3.0% for FT/Q/S, while for TL it is equal to 4.5% for FT/Q/G and 6.1% for FT/Q/S. For $\mathcal{E}(d,p)$ further cost increase is observed, reaching 7.6% for FT/Q/S and TL.

As far as computation time is concerned, the SL and DL cases are optimized very quickly, yet the computation time becomes substantially longer for TL, the reason being the excessive number of variables and constraints in the LP formulation.

4.1.5 Implementation issues and suggested formula

The flow thinning mechanism assumes that the capacity of each tunnel is controlled at its source node by a packet admission control mechanism based on the on-line knowledge of the currently available link capacities. Therefore, some signalling mechanism for interchanging information concerning the current state of link capacities must be applied. In the case of FT, and of FT/Q and FT/A with the full range $\mathcal{E}$, this requires some kind of a flooding protocol since the source nodes of the tunnels need to be aware of the

Table 4.4: Results for FT/Q.

	range	structure	F^*	$\Delta F^*[\%]$	$\lvert\mathcal{P}^*\rvert$	$\lvert\mathcal{P}^u\rvert$	total
SL	$\mathcal{E}(d,p)$	G	40236	0	275	125	0.2s
		S	40236	0	276	124	0.3s
	$\mathcal{E}^+(d,p)$	G	40236	0	283	124	0.2s
		S	40236	0	274	124	0.2s
	$\mathcal{E}$	G	40236	0	283	124	0.2s
		S	40236	0	274	124	0.3s
DL	$\mathcal{E}(d,p)$	G	41939	4.6	285	176	8s
		S	42034	4.8	283	176	12s
	$\mathcal{E}^+(d,p)$	G	41152	2.6	216	165	6s
		S	41278	3.0	187	164	9s
	$\mathcal{E}$	G	40104	0.03	216	144	3s
		S	40285	0.5	187	147	10s
TL	$\mathcal{E}(d,p)$	G	39939	7.4	232	202	15m59s
		S	40044	7.6	223	202	11m36s
	$\mathcal{E}^+(d,p)$	G	38864	4.5	191	164	38h25m27s
		S	39456	6.1	165	147	1h5m14s
	$\mathcal{E}$	G	37269	0.2	191	134	15h59m39s
		S	37705	1.3	165	132	7h58m31s

current availability state of *all* links. Since flooding signalling takes time, the capacity adjustment process may lead to unprecise tunnel capacity control and, in consequence, to traffic losses. Moreover, flooding may be excessively complex to implement. This actually means that feasibility of FT, FT/Q/$\mathcal{E}$ and FT/A/$\mathcal{E}$ is problematic.

On the other hand, in the case of FT/Q and FT/A with the $\mathcal{E}(d,p)$ range, the signalling in question is simple and fast as it pertains solely to the links of the tunnel: when availability of a link is changed, an appropriate message is propagated backwards to the source node of each tunnel traversing the considered link, resulting in efficient and timely message delivery for link availability state monitoring. For the $\mathcal{E}^{+}(d,p)$ range the signalling is similar, only more information is propagated.

Another issue is how the tunnel capacity control performs for the states not considered in optimization, i.e., the states not in the set $\mathcal{S}$. In this case the most cost effective flow thinning mechanism, i.e., FT, is not satisfactory, as it does not provide consistent means of tunnel control. Thus, since inappropriate setting of tunnel capacity can lead to link overloads, the FT mechanism is risky in this aspect. On the contrary, the FT/Q and FT/A mechanisms can simply apply the flow thinning formula also for the unforeseen states, additionally modifying the tunnel capacity to 0 or x^0_{dp} when necessary, i.e., when the value obtained from the FT formula is below 0 or above x^0_{dp}, respectively. (Traffic efficiency of such extended flow thinning is addressed in Section 8.2 of [50] for FT/A/G.). We may also expect that the larger the range of the FT formula, the better the approximation of the proper tunnel capacity in the states not considered in optimization – this, however, still needs to be verified. Here, simple structure S of the FT formula seems more safe than general structure G because the former does not allow to exceed the nominal tunnel capacity in any state.

FT is the most general flow thinning mechanism and therefore it results in the lowest cost of link capacity. In fact, in this aspect FT is quite close to the UR benchmark mechanism, which provides the lower bound on the link capacity cost for *any* protection/restoration mechanism (see [56]). Tables 4.1, 4.2 and 4.4 show that the network cost achieved with FT/Q/G with range $\mathcal{E}$ is almost the same as the cost of FT, while the cost for FT/Q/G with range $\mathcal{E}^{+}(d,p)$ is higher but to a reasonable extent (4.5% of cost increase with respect to FT at most). The network cost achieved with FT/Q/G with the smallest range $\mathcal{E}(d,p)$ is up to 7.4% higher than the cost for FT. As far as FT/A is concerned, the network cost increase is significantly higher than for FT/Q.

Having in mind that the optimization of flow thinning is performed off-line, the total computation times are acceptable in all the considered cases apart from FT/Q with the ranges $\mathcal{E}$ and $\mathcal{E}^+(d,p)$, but even that, however, is not a big issue taking into account that the results shown in the tables were obtained using a plain laptop, and the efficiency of the optimization algorithm could be improved (if really needed) by exploiting specific properties of its master problem.

The above observations (summarized in Table 4.5) suggest that FT/Q/G/$\mathcal{E}^+(d,p)$, i.e., the FT mechanism with the general quadratic flow thinning formula encompassing all links incident to the nodes the tunnel, is a reasonable traffic protection mechanism to be considered for implementation in the network.

Table 4.5: Summary of the results.

	range	structure	implementation	states not in $\mathcal{S}$	capacity cost	optimization time
UR	$\mathcal{E}$	–	infeasible	not covered	very low	very short
FT	$\mathcal{E}$	–	problematic	not covered	low	medium
FT/A	$\mathcal{E}(d,p)$	G	feasible	covered	high	medium
		S	feasible+	covered	high	short
	$\mathcal{E}^+(d,p)$	G	feasible	covered+	medium	medium-
		S	feasible	covered+	high	medium
	$\mathcal{E}$	G	flooding	covered++	medium	medium
		S	flooding	covered++	high	short+
F/Q	$\mathcal{E}(d,p)$	G	feasible	covered	medium	medium-
		S	feasible+	covered	medium	medium-
	$\mathcal{E}^+(d,p)$	G	feasible	covered+	low	excessive
		S	feasible	covered+	low	medium
	$\mathcal{E}$	G	flooding	covered++	low	excessive
		S	flooding	covered++	low	excessive

4.2 State generation algorithm study

The second part of the numerical studies illustrates efficiency of the SGA+PGA combined iterative optimization procedure described in Section 3.3 from the computational efficiency and the optimized network cost viewpoints for different variants of the flow thinning formula.

4.2.1 Network instance

This part of the study was performed using an FSO network instance designed for Paris metropolitan area using realistic data – population distribution data to calculate the traffic matrix, and historical weather data to calculate typical FSO link degradation ratios. The considered instance (referred to as PMAN) was introduced in [38]. The network (depicted in Figure 4.2) is composed of $|\mathcal{V}| = 12$ nodes and $|\mathcal{E}| = 21$ undirected links. Its set of traffic demands consists of $|\mathcal{D}| = 66$ undirected demands; the reference demand volumes, expressed in Gbps, are given in the form of a traffic matrix depicted in Table 4.6. The link capacities are expressed in Gbps as well, while the unit link capacity cost is assumed to be equal to 1 cost unit per 1 Gbps for all the links (i.e., $\xi(e) = 1$, $e \in \mathcal{E}$).

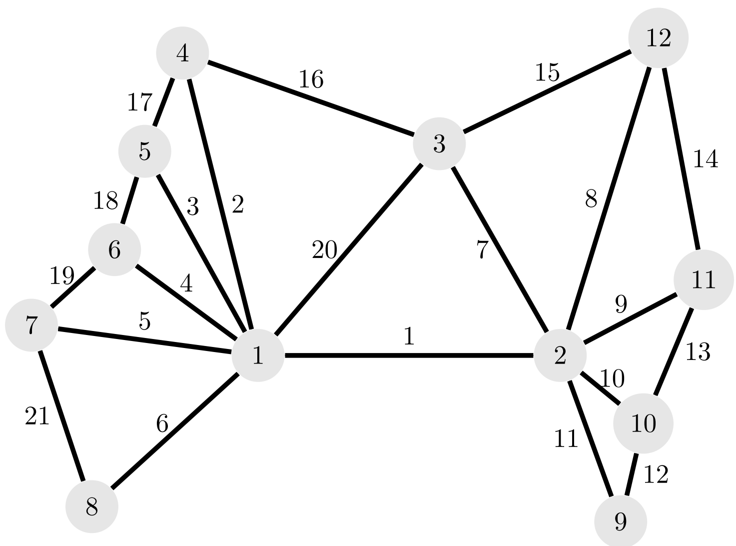

Figure 4.2: PMAN network topology.

It is important to note that although in reality traffic demands (modelling the Internet traffic) are directed and FSO links are realized by means of full-duplex FSO systems, each providing two oppositely directed links of the same capacity, the use of undirected links and undirected demands is correct provided the traffic matrix is symmetric (as

Table 4.6: Traffic matrix [Gbps].

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
(1) Paris1	–	74.38	5.40	4.26	4.99	5.02	5.24	4.74	5.41	5.26	4.87	3.82
(2) Paris2	–	–	5.40	4.26	4.99	5.02	5.24	4.74	5.41	5.26	4.87	3.82
(3) Saint-Denis	–	–	–	0.23	0.34	0.35	0.36	0.33	0.38	0.37	0.29	0.23
(4) Argenteuil	–	–	–	–	0.27	0.28	0.29	0.26	0.30	0.30	0.22	0.01
(5) Colombes	–	–	–	–	–	0.36	0.37	0.34	0.39	0.37	0.34	0.26
(6) Courbevoie	–	–	–	–	–	–	0.37	0.34	0.39	0.38	0.34	0.26
(7) Nanterre	–	–	–	–	–	–	–	0.34	0.40	0.39	0.35	0.28
(8) Versailles	–	–	–	–	–	–	–	–	0.29	0.28	0.23	0.18
(9) Vitry-Sur-Seine	–	–	–	–	–	–	–	–	–	0.31	0.26	0.20
(10) Creteil	–	–	–	–	–	–	–	–	–	–	0.25	0.20
(11) Montreuil	–	–	–	–	–	–	–	–	–	–	–	0.21
(12) Aulnay-Sous-Bois	–	–	–	–	–	–	–	–	–	–	–	–

in PMAN). If that is the case, we can first assume undirected demands with the values specified in the upper part (i.e., above the diagonal) of the traffic matrix (as in Table 4.6) and dimension the network using the undirected network model. Then, in the network dimensioning post-processing phase, we can bring back the directions of the demands for the upper-part demands and create they directed path flows by directing the optimal (undirected) nominal path flows x^0 accordingly. In this way each link $e = \{v(e), w(e)\}$ will carry a set of (directed) flows in the $v \to w$ direction, and a set of flows in the $w \to v$ direction. The resulting directed link loads (let us denote them by $Y(v(e), w(e))$ and $Y(w(e), v(e)))$ sum up to y_e^0 (the optimal nominal undirected capacity of link e). After that, in the same way we create directed path flows for the directed demands from the lower part (below the diagonal) of the traffic matrix that are symmetrical (but oppositely directed) to their counterparts from the upper part. For link e this will result in link loads $Y'(w(e), v(e))$ and $Y'(v(e), w(e))$. Clearly, $Y'(w(e), v(e)) = Y(v(e), w(e))$ and $Y'(v(e), w(e)) = Y(w(e), v(e))$. Therefore, if we realize loads $Y(v(e), w(e))$ and $Y'(v(e), w(e))$ on arc $(v(e), w(e)$, and loads $Y(w(e), v(e))$ and

$Y'(w(e), v(e))$ on arc $(w(e), v(e)$, these loads will be carried on full-duplex link e with capacity y_e^0 in each direction.

In fact, the optimization model presented in this paper can be easily modified to directly consider full-duplex links and directed demands. But since for symmetric traffic optimal solutions of such a modified model are equivalent to the solutions described above, we prefer to use the above model as it requires two times less flow variables.

4.2.2 Efficiency of SGA+PGA

Table 4.7 presents the results of solving FT/A/S/$\mathcal{E}(d,p)$ using SGA+PGA optimization procedure. Analogous results for FT/A/G/$\mathcal{E}(d,p)$ are presented in Table 4.8. We examine six sets of states $\widehat{\mathcal{B}}(N,a,b,H)$ (rows 1-6 in Table 4.7). Recall that, considering for example row no 3, the parameters $N = (21,1,2)$, $a = (1, 0.75, 0.75)$, and $b = (0,0,0.05)$ imply that the corresponding state set includes all combinations of simultaneous degradation of $U^2 \leq 1$ links that are degraded to availability ratio 0.75, and $U^3 \leq 2$ links that are degraded also to availability ratio 0.75 (then $U^1 := 21 - U^2 - U^3$ links are fully available). When state (U^1, U^2, U^3) with respective number of links of each class is observed, the traffic of each demand $d \in \mathcal{D}$ is equal to $H(d)(1 - 0.05U^3)$ (see (2.10b), (2.9)), thus the reference demand volumes are reduced only when $U^3 > 0$. The reference demand values $H(d)$, $d \in \mathcal{D}$ are taken from Table 4.6 assuming the lexicographical ordering of the node pairs $(1,2), (1,3), \ldots, (11,12)$. It is worth noting that the considered link availability ratios (i.e., 1.0, 0.75 and 0.5), correspond to the modulation and coding schemes applicable to FSO links; see the discussion in [38] based on [70, 33].

In the calculations, the initial path lists for the PG algorithm contain only one path per demand (66 paths in total) – the shortest path with respect to the link unit costs (thus, these are the shortest paths with respect to the number of hops, as $\xi(e) = 1$, $e \in \mathcal{E}$). The initial list of states $\mathcal{S}^0$ contains the full-availability state (all links fully available), and all states with exactly one degraded link with link availability coefficient $\alpha(e,s) = 0.75$; thus, there are 22 states in total. For all the states in the initial list 100% traffic protection is assumed, i.e., $h(d,s) = H(d)$, $d \in \mathcal{D}$, $s \in \mathcal{S}^0$. Note that assigning the entire demand volume $H(d)$ for each demand d to its shortest path will result in the optimally dimensioned network when protection is not considered. For the examined network this cost turns out to be $F^* = 246.38$.

Table 4.7: Results of SGA+PGA for FT/A/S/$\mathcal{E}(d, p)$.

$N(k)$	$a(k)$	$b(k)$	F^*	$\Delta\mathcal{P}$	$\Delta\mathcal{S}$	iter	t [s]	t/PGA [s]	t/SGA [s]
(21, 1)	(1, .75)	(0, 0)	312.36	215	0	0	161	160	1
(21, 1, 1)	(1, .75, .75)	(0, 0, .05)	325.72	262	168	3	1477	737	740
(21, 1, 2)	(1, .75, .75)	(0, 0, .05)	326.83	300	287	3	4288	803	3485
(21, 1, 2, 1)	(1, .75, .75, .5)	(0, 0, .05, .1)	432.30	358	541	3	5853	1197	4656
(21, 1, 2, 2)	(1, .75, .75, .5)	(0, 0, .05, .1)	448.58	340	534	10	12296	2685	9611
(21, 1, 2, 3)	(1, .75, .75, .5)	(0, 0, .05, .1)	449.98	307	465	6	7534	1592	5942

Table 4.8: Results of SGA+PGA for FT/A/G/$\mathcal{E}(d, p)$.

$N(k)$	$a(k)$	$b(k)$	F^*	$\Delta\mathcal{P}$	$\Delta\mathcal{S}$	iter	t [s]	t/PGA [s]	t/SGA [s]
(21, 1)	(1, .75)	(0, 0)	312.36	211	0	0	408	407	1
(21, 1, 1)	(1, .75, .75)	(0, 0, .05)	325.72	286	190	4	4544	1433	3111
(21, 1, 2)	(1, .75, .75)	(0, 0, .05)	326.83	372	385	12	25891	5241	20650
(21, 1, 2, 1)	(1, .75, .75, .5)	(0, 0, .05, .1)	432.30	353	570	4	18668	2619	16049
(21, 1, 2, 2)	(1, .75, .75, .5)	(0, 0, .05, .1)	448.58	342	613	8	31459	3987	27472
(21, 1, 2, 3)	(1, .75, .75, .5)	(0, 0, .05, .1)	449.98	328	577	10	24725	4130	20595

The meaning of the parameters in the tables is as follows:

– $N(k)$, $a(k)$, $b(k)$: state polytope parameters

– F^*: cost of the optimal solution

– $\Delta|\mathcal{P}|$: number of generated paths

– $\Delta|\mathcal{S}|$: number of generated states

– number of iterations of SGA+PGA (iter), total computation time (t[s]); computation time spent in PGA (t/PGA[s]); computation time spent in SGA (t/SGA[s]).

Tables 4.7 and 4.8 show that the SGA+PGA is computational efficient. It requires up to $10 - 12$ iterations and generates a reasonable number of paths and states. In both tables the first row describes the state set $\widehat{\mathcal{B}}(N, a, b, H)$ which is equal to the initial state set $\mathcal{S}^0$; therefore no iterations of SGA+PGA are needed since the algorithm stops after the first execution of PGA. We notice that computation time spent in SGA is in most cases larger than in PGA, and the total computation times are typically of the order of hours. Still, those computation times are acceptable considering the size of the problems and the fact that the problems are not supposed to be solved online but rather one time per the network management cycle.

In Table 4.9 we compare performance of FT/A/S and FT/A/G for two ranges: $\mathcal{E}(d,p)$ and $\mathcal{E}^+(d,p)$. In the table, the asterisks in the computation time columns of $\mathcal{E}^+(d,p)$ range denote that the result was obtained through solving the appropriate master problem only once for the final path lists and state lists obtained for the range $\mathcal{E}(d,p)$ (recall that a similar sub-optimal heuristic approach was used in Section 4.1.3). Note that the network cost for both FT/A/S variants in Table 4.9 is the same.

Table 4.9: Network cost and computation time of SGA+PGA for FT variants.

$N(k)$	$a(k)$	$b(k)$	$\mathcal{E}(d,p)$				$\mathcal{E}^+(d,p)$			
			FT/A/S		FT/A/G		FT/A/S		FT/A/G	
			F^*	t [s]	F^*	t [s]	F^*	t [s]*	F^*	t [s]*
(21, 1)	(1, .75)	(0, 0)	312.36	161	312.36	408	312.36	1	312.36	1
(21, 1, 1)	(1, .75, .75)	(0, 0, .05)	325.72	1477	325.72	4544	325.72	23	317.59	57
(21, 1, 2)	(1, .75, .75)	(0, 0, .05)	326.83	4288	326.83	25891	326.83	31	317.91	307
(21, 1, 2, 1)	(1, .75, .75, .5)	(0, 0, .05, .1)	432.30	5853	432.30	18668	432.30	171	409.38	1553
(21, 1, 2, 2)	(1, .75, .75, .5)	(0, 0, .05, .1)	448.58	12296	448.58	31459	448.58	150	415.27	1757
(21, 1, 2, 3)	(1, .75, .75, .5)	(0, 0, .05, .1)	449.98	7534	449.98	24725	449.98	80	416.99	896

A general conclusion is that optimization of FT/A/G is more time consuming than FT/A/S, especially in the case of the $\mathcal{E}^+(d,p)$ range, but its cost can be noticeably lower than that of FT/A/S (there is an up to 8% cost difference between FT/A/S and FT/A/G for the $\mathcal{E}^+(d,p)$ range).

Table 4.10 presents results of SGA extended to the quadratic FT formula. The calculations were performed for fixed path lists obtained from the respective final path lists in the solutions presented in Table 4.9. As expected, the quadratic formula leads to less expensive networks in terms of the total link capacity cost (by about 1-4%, see the F^* values in the second row in Tables 4.9 and 4.10). However, due to excessive computation time, it was possible to solve the optimization problem only for the first two state polytopes. Note that the cost values in the first rows of Tables 4.9 and 4.10 are the same, as this case considers only single link degradations. Note also, that the reason why the network cost ($F^* = 313.99$) for the FT/Q/S/$\mathcal{E}^+(d,p)$ case is smaller than the cost ($F^* = 316.44$) for the FT/Q/G/$\mathcal{E}^+(d,p)$ case is that the path lists for the quadratic FT were not optimized but just taken from the corresponding affine FT solutions.

Table 4.10: Results of SGA for FT/Q variants.

$N(k)$	$a(k)$	$b(k)$	$\mathcal{E}(d,p)$				$\mathcal{E}^+(d,p)$			
			FT/Q/S		FT/Q/G		FT/Q/S		FT/Q/G	
			F^*	t [s]	F^*	t [s]	F^*	t [s]	F^*	t [s]
(21, 1)	(1, .75)	(0, 0)	312.36	4	312.36	6	312.36	5	312.36	7
(21, 1, 1)	(1, .75, .75)	(0, 0, .05)	317.53	54	316.52	145	313.99	184	316.44	483

4.3 Network cost efficiency

Below, in Table 4.11, we briefly summarize the cost efficiency of the FT mechanisms, in particular in comparison with UR. The comparison encompasses UR, FT, FT/A/G and FT/A/S, and is based on SL, DL, and TL lists of states, described in Section 4.1.1. Note that the state polytope model is not used here since the SGA+PGA algorithm is not applicable to UR and FT. The percentages in the FT column express the increase of the network cost for FT in comparison with the cost for UR, while the percentages in the FT/A/G/$\mathcal{E}$ column show the increase of the cost for FT/A/G/$\mathcal{E}$ in comparison with the cost for FT. In all other columns, the percentages compare the reported cost to the cost for FT/G/A/$\mathcal{E}$.

Table 4.11: Network cost for FT/A variants for PMAN.

	UR	FT	$\mathcal{E}$		$\mathcal{E}^+(d,p)$		$\mathcal{E}(d,p)$	
			FT/A/G	FT/A/S	FT/A/G	FT/A/S	FT/A/G	FT/A/S
SL	392.90	416.63(6%)	416.63(0%)	416.63(0%)	416.63(0%)	416.63(0%)	416.63(0%)	416.63(0%)
DL	362.80	379.10(4%)	385.13(2%)	407.15(6%)	390.82(1%)	407.15(6%)	407.15(6%)	407.15(6%)
TL	325.03	333.60(3%)	337.46(1%)	350.17(4%)	339.12(1%)	350.17(4%)	350.17(4%)	350.17(4%)

A general conclusion is that FT/A/G/$\mathcal{E}^+(d,p)$ performs very well in terms of the network cost with respect to FT (the cost for FT provides a lower bound for all FT variants). Also, the cost for FT is not much larger than the cost for UR, and the cost for FT/A/S (the same for all formula ranges) is acceptable.

Finally, recall that the decrease of the network cost observed when the FT/A formula is substituted by its FT/Q counterpart for the cases reported in Tables 4.9 and 4.10 was between 1-4%. However, such a decrease can be more prominent, as illustrated in Table 4.12.

Table 4.12: Network cost for FT/Q variants for *polska*.

	$\mathcal{E}$		$\mathcal{E}^+(d,p)$		$\mathcal{E}(d,p)$	
	FT/Q/G	FT/Q/S	FT/Q/G	FT/Q/S	FT/Q/G	FT/Q/S
DL	40104 (+5%)	40285 (+17%)	41152 (+6%)	41278 (+14%)	41939 (+12%)	42034 (+12%)
TL	37269 (+4%)	37705 (+11%)	38864 (+3%)	39456 (+6%)	39939 (+5%)	40044 (+5%)

The above table shows the results obtained for *polska* network described in Section 4.1.1. In Table 4.12 all six variants of the FT/Q formula are considered for the DL and TL scenarios. Each entry of the table gives the network cost achieved with the formula specified for its column and the scenario specified for its row, together with the percentage (given in parenthesis) of the increase of the network cost when the corresponding FT/A formula is applied. This time the gain from FT/Q can be quite substantial, ranging from 4 to 17%.

Chapter 5

Summary

Logical Tunnel Capacity Control (LTCC) is a traffic routing and protection strategy designed for communications networks characterized by frequent link capacity fluctuations. It introduces a novel Flow Thinning (FT) mechanism of controlling the size of the flows assigned to network tunnels. This thesis studies optimization problem models and optimization solution algorithms of designing the network that uses the LTCC strategy and implements the FT mechanism. The models and the algorithms are essential for evaluating the efficiency and for effective deployment of the LTCC strategy.

The FT mechanism is based on the FT formula, which might vary with respect to the form, structure and range. The thesis is a comprehensive study of the network design problems corresponding to the resulting variants of the FT formula. Additionally, it studies two classes of problem formulations that differ with respect to the approach of describing the availability of the links. While the direct approach enumerates all the considered network states, the indirect approach is based on the notion of the so called state polytope, which enables considering (potentially) exponentially many network states in a compact way.

The presented optimization models are, in general, hardly solvable directly by optimization software, due to, in particular, a huge number of network paths and network states and a potentially complex and non-linear FT formula. Therefore, the thesis presents and studies the application of optimization solution algorithms that are based on the path generation and, if the state-polytope-based approach is used, state generation approaches. The thesis provides a detailed study of the resulting Path Generation Algorithm (PGA) and State Generation Algorithm (SGA. The formal difficulty of PGA results from the fact that the Pricing Problem (PP), being the key component of PGA, is flow-thinning-formula-dependent. Therefore, a different PP had to be derived for each variant of the FT formula. The key components of SGA are, in turn, feasibility tests, which were also derived in this thesis. The thesis presents how the combination of PGA and SGA enables solving the network design problem models, in particular, the ones based on the state polytope concept, to optimality.

The effectiveness of the LTCC strategy was analyzed through extensive numerical studies. They used a real urban network instance based on the FSO technology, for which the LTCC strategy is especially well suited. The link availability description was prepared using historical weather data. The goal of the studies was to examine the efficiency of the developed optimization models and algorithms with respect to the computation time and the solution quality. They also aim at evaluating the performance of the LTCC traffic routing and protection strategy, measured, in particular, by means of the total network cost, and compared to the idealistic reference strategy of unrestricted reconfiguration.

A general conclusion derived from the numerical studies is that the proposed traffic routing and protection strategy turns out to be a reasonable candidate to be used in real communications networks. It appears that the flow thinning formula and the state polytope notion provide much flexibility in network design, as they provide means to loo k for a balance between implementation feasibility, routing robustness, network cost, and optimization time.

The use of the flow thinning formulae for tunnel flow adjustment is relatively simple to implement using existing network protocols. The signalling is fast and simple when the path's links and path's incident links ranges of the flow thinning formula are applied, while the use of the all links range makes the signalling problematic.

When the robustness of the applied strategy is of the utmost importance, one should use the flow thinning formula with the simple structure and with the ranges as large as possible. The models with the general structure are less robust as they do not circumvent exceeding the nominal tunnel capacity. Still, any flow thinning formula is robust by its nature, as even for some unforeseen state the traffic is inevitably thinned at the originating nodes according to a consistent formula dependent on the link degradation coefficients. Moreover, the robustness could be improved even further by applying the state polytope model.

Obviously, the use of the flow thinning formula with the quadratic form, the general structure and the all links range leads to the most efficient traffic handling and thus requires the least network capacity cost. In this case the cost is clearly very low, being only several percent higher than the benchmark mechanism of the unrestricted reconfiguration.

The numerical results show that the elaborated solution algorithms (i.e., PGA, SGA, and SGA+PGA) are computationally efficient, despite the necessity to solve binary-type MIP pricing problems in the path generation. The optimization times varied from less than a second to tens of hours (about one hour on average) on a plain laptop, which is acceptable as the optimization is supposed to be performed off-line. Clearly, in terms of the optimization time the most challenging optimization problem models are those with the flow thinning formula of the quadratic form and with the path's incident links range, as the number of optimization variables and constraints needed to formulate those problem variants may be excessive. In such cases, it is definitely worth to consider the heuristic approach applied in the thesis. With this approach, the most severe variants are not treated by PGA. Instead, a single master problem is being solved with the predefined path lists, taken from the optimal solution of the corresponding less severe variants.

To summarize, the recommended flow thinning formula to be considered for implementation is the flow thinning formula with quadratic form, the general structure, and the path's incident links range, i.e., formula FT/Q/G/$\mathcal{E}^+(d,p)$. It is worth mentioning that although the achieved results are acceptable, there is still a lot of room for further enhancements. Some ideas for future work are adding modularity of link capacity, improving computational efficiency of the most challenging flow thinning formula variants, testing effectiveness of the heuristic approach, to name a few.

Index

Bibliography

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] C. Barnhart, C. A. Hane, and P. H. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.

[3] A. Ben-Tal, S. Boyd, and A. Nemirovski. Extending scope of robust optimization: Comprehensive robust counterparts of uncertain problems. *Mathematical Programming*, 107(1–2):63–89, 2006.

[4] A. Ben-Tal, L. El Ghaoui, and A. S. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

[5] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

[6] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[7] J. F. Benders. Partitioning procedures for solving mixed variable programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[8] D. Bertsimas and M. Sim. The Price of Robustness. *Operations Research*, 52(1):35–53, 2004.

[9] Y. Bi and A. Tang. Uncertainty-aware optimization for network provisioning and routing. In *Proc. 53rd Annual Conference on Information Sciences and Systems (CISS 2019)*, pages 1–6, 2019.

[10] C. Chekuri. Routing and network design with robustness to changing or uncertain traffic demands. *SIGACT News*, 38:106–129, 09 2007.

[11] T. Cinkler, P. Laborczi, and A. Horvath. Protection through thirfty configuration. In *Proc. 16th International Teletraffic Congress (ITC 1999)*. Elsevier Science B.V., 1999.

[12] IBM ILOG Cplex. V12. 1: User's manual for CPLEX. *International Business Machines Corporation*, 46(53):157, 2009.

[13] G. Dahl and M. Stoer. A cutting plane algorithm for multicommodity survivable network design problems. *INFORMS Journal on Computing*, 10:1–11, 1998.

[14] F. D'Andreagiovanni, D. Nace, M. Pióro, M. Poss, M. Shehaj, and A.Tomaszewski. On robust FSO network dimensioning. In *Proc. 9th International Workshop on Resilient Networks Design and Modeling (RNDM 2017)*. Alghero, Italy, September 2017.

[15] J. Desrosiers and M. E. Lübbecke. *Branch-Price-and-Cut Algorithms*. John Wiley & Sons, Ltd, 2011.

[16] A. Forsgren and M. Prytz. *Telecommunications Network Design*. Springer US, Boston, MA, 2006.

[17] Y. Fouquet, D. Nace, M. Pióro, and M. Poss. An optimization framework for traffic restoration in optical wireless networks with partial link failures. *Optical Switching and Networking*, 23:108–117, 2017.

[18] Y. Fouquet, D. Nace, M. Pióro, and I. Shinko. Elastic routing: A distributed variant, implementation issues, and numerical results. In *Proc. 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC 2013)*, pages 98–104, 2013.

[19] B. Gabor, D. Orincsay, and A. Kern. Surviving multiple network failures using shared backup path protection. In *Proc. 9th IEEE International Symposium on Computers and Communications (ISCC 2003)*, 2003.

[20] A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[21] W. D. Grover. *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking.* Prentice Hall, 2003.

[22] A. Gupta, P. Anand, R. Khajuria, and S. Bhagat. A survey of free space optical communication network channel over optical fiber cable communication. *International Journal of Computer Applications*, 105:975–8887, November 2014.

[23] K. Holmberg and D. Yuan. A multicommodity network-flow problem with side constraints on paths solved by column generation. *INFORMS Journal on Computing*, 15(1):42–57, 2003.

[24] L. Huang, M. Chen, and Y. Liu. Learning-aided stochastic network optimization with state prediction. *IEEE/ACM Transactions on Networking*, 26(4):1810–1820, 2018.

[25] I. Kalesnikau. Comparing two pricing methods for path generation in a resilient network design problem. *Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne*, 8-9:882–885, 2017.

[26] I. Kalesnikau. Flow Thinning Simulation System (FTSS): NS-3 implementation and simulation results. Technical report, Institute of Telecommunications, Warsaw University of Technology, 2020.

[27] I. Kalesnikau, M. Pióro, M. Poss, D. Nace, and A. Tomaszewski. A robust optimization model for affine/quadratic flow thinning: A traffic protection mechanism for networks with variable link capacity. *Networks*, 75(4):420–437, 2020.

[28] I. Kalesnikau, M. Pióro, J. Rak, H. Ivanov, E. Fitzgerald, and E. Leitgeb. Enhancing resilience of FSO networks to adverse weather conditions. *IEEE Access*, 9:123541–123565, 2021.

[29] A. M. C. A. Koster, A. Zymolka, M. Jäger, and R. Hülserman. Demand-wise shared protection for meshed optical networks. *Journal of Network and System Management*, 13(1):35–55, 2005.

[30] C. Larsson. *Design of Modern Communication Networks: Methods and Applications.* Academic Press, Inc., USA, 1st edition, 2014.

[31] L. Lasdon. *Optimization Theory for Large Systems.* MacMillan, 1970.

[32] H. Luss. On equitable resource allocation problems: A lexicographical minimax approach. *Operations Research*, 47:361–378, 1999.

[33] J. S. Malhotra, M. Kumar, and A. K. Sharma. Performance Comparison of PS-QPSK and PM-QPSK Modulation Schemes in High Capacity Long Haul DWDM Optical Communication Link. *International Journal of Engineering Sciences*, 2(5):154–159, 2013.

[34] M. Minoux. *Mathematical Programming: Theory and Algorithms.* John Wiley & Sons, 1986.

[35] P. G. Mogre, M. Hollic, and R. Steimetz. QoS in wireless mesh networks: challenges, pitfalls, and roadmap to its realization. In *Proc. 17th International workshop on Network and operating systems support for digital audio & video (NOSSDAV'07).* Urbana-Champaign, IL, USA, June 2007.

[36] M. Mycek and A. Tomaszewski. Implementation options for the LTCC (Logical Tunnel Capacity Control) strategy. Technical report, Institute of Telecommunications, Warsaw University of Technology, 2020.

[37] D. Nace and J. B. Orlin. Lexicographically minimum and maximum load linear programming problems. *Operations Research*, 55(1):182–187, 2007.

[38] D. Nace, M. Pióro, M. Poss, F. D'Andreagiovanni, I. Kalesnikau, M. Shehaj, and A. Tomaszewski. An optimization model for robust FSO network dimensioning. *Optical Switching and Networking*, 32(0):25–40, 2019.

[39] M. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems.* Morgan & Claypool, 2010.

[40] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley & Sons, 1988.

[41] N. Nishida. Practical use of operations research for production planning. *IFAC Proceedings Volumes*, 24(12):33–36, 1991.

[42] W. Ogryczak, M. Pióro, and A. Tomaszewski. Telecommunications network design and max-min optimization problem. *Journal of Telecommunications and Information Technology*, 4, January 2005.

[43] S. Orlowski and M. Pióro. Complexity of column generation in network design with path-based survivability mechanisms. *Networks*, 59(1):132–147, 2012.

[44] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski. SNDlib 1.0 – survivable network design library. *Networks*, 55(3):276–286, 2010.

[45] B. Ostrowski, M. Pióro, A. Tomaszewski, and E. Fitzgerald. Resilience through multicast – an optimization model for multi-hop wireless sensor networks. *Ad Hoc Networks*, 107:102239, 2020.

[46] B. Ostrowski, M. Pióro, A. Tomaszewski, M. Nowak, and I. Kalesnikau. Throughput vs. resilience in multi-hop wireless sensor networks with periodic packet traffic. In *Proc. 16th International Conference on the Design of Reliable Communication Networks (DRCN 2020)*, pages 1–7, 2020.

[47] A. Ouorou and J.-Ph. Vial. A model for robust capacity planning for telecommunications networks under demand uncertainty. In *Proc. 6th International Workshop on Design and Reliable Communication Networks (DRCN 2007)*, pages 1–4, 2007.

[48] M. Pióro. Fair routing and related optimization problems. In *Proc. 15th International Conference on Advanced Computing and Communications (ADCOM 2007)*, pages 229–235, 2007.

[49] M. Pióro, E. Fitzgerald, I. Kalesnikau, D. Nace, and J. Rak. *Optimization of Wireless Networks for Resilience to Adverse Weather Conditions*, pages 523–556. Springer International Publishing, Cham, 2020.

[50] M. Pióro, Y. Fouquet, D. Nace, and M. Poss. Optimizing flow thinning protection in multicommodity networks with variable links capacity. *Operations Research*, 64(2):273–289, 2016. Best paper award in INFORMS Telecom and Network Analytics 2016-2018.

[51] M. Pióro, I. Kalesnikau, and M. Poss. An optimization model for affine flow thinning – a traffic protection mechanism for FSO networks. In *Proc. 9th International Workshop on Resilient Networks Design and Modeling (RNDM 2017)*, 2017.

[52] M. Pióro, I. Kalesnikau, and M. Poss. Path generation for affine flow thinning. In *Proc. 8th International Network Optimization Conference (INOC 2017).* Lisbon, Portugal, February 2017.

[53] M. Pióro, I. Kalesnikau, and M. Poss. Path generation for affine flow thinning. *Electronic Notes in Discrete Mathematics*, 64:355–364, February 2018.

[54] M. Pióro, I. Kalesnikau, and M. Poss. Practical aspects of Flow Thinning optimization. In *Proc. 10th International Workshop on Resilient Networks Design and Modeling (RNDM 2018)*, pages 1–8. Longyearbyen, Norway, September 2018.

[55] M. Pióro, I. Kalesnikau, and M. Poss. An optimization model for quadratic flow thinning – a traffic protection mechanism for FSO networks. *Optical Switching and Networking*, 31(0):168–182, 2019.

[56] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks.* Morgan-Kaufmann, 2004.

[57] M. Pióro, D. Nace, and Y. Fouquet. On protected traffic routing in wireless networks with partial multiple link failures. In *Proc. 8th International Conference on Broadband and Wireless Computing (BWCCA 2013).* Compiegne, France, October 2013.

[58] M. Pióro, D. Nace, and Y. Fouquet. An optimization model for communication networks with partial multiple link failures. In *Proc. 5th International Workshop on Reliable Networks Design and Modeling (RNDM 2013), co-located with ICUMT 2013 Congress.* Almaty, Kazakhstan, October 2013.

[59] M. Pióro, A. Tomaszewski, and A. Capone. Maximization of multicast periodic traffic throughput in multi-hop wireless networks with broadcast transmissions. *Ad Hoc Networks*, 77:119–142, 2018.

[60] M. Poss and C. Raack. Affine recourse for the robust network design problem: Between static and dynamic routing. *Networks*, 61(2):180–198, 2013.

[61] P. Pourhejazy and O. K. Kwon. The new generation of operations research methods in supply chain optimization: A review. *Sustainability*, 8(10), 2016.

[62] J. Rak. *Resilient Routing in Communication Networks*. Springer International Publishing, 2015.

[63] M. Shehaj. *Robust dimensioning of wireless optical networks with multiple partial link failures*. PhD thesis, Université de Technologie de Compiègne, February 2020.

[64] M. Shehaj, I. Kalesnikau, D. Nace, M. Pióro, and E. Qafzezi. Modeling transmission degradation on FSO links caused by weather phenomena for WMN optimization. In *Proc. 11th International Workshop on Resilient Networks Design and Modeling (RNDM 2019)*, pages 1–7, 2019.

[65] M. Shehaj, D. Nace, I. Kalesnikau, and M. Pióro. Dimensioning of hybrid FSO/fiber networks. In *Proc. 2nd International Balkan Conference on Communications and Networking (BalkanCom 2018)*. Podgorica, Montenegro, June 2018.

[66] M. Shehaj, D. Nace, I. Kalesnikau, and M. Pióro. Link dimensioning of hybrid FSO/fiber networks resilient to adverse weather conditions. *Computer Networks*, 161:1–13, 2019.

[67] I. Shinko, D. Nace, and Y. Fouquet. A study on a distributed rerouting scheme. In *Proc. 27th International Conference on Advanced Information Networking and Applications Workshops (AINA 2013)*, pages 1461–1466, 2013. Barcelona, Spain.

[68] E. A. Silver. Operations research in inventory management: A review and critique. *Operations Research*, 29(4):628–645, 1981.

[69] J. Strand, A. L. Chiu, and R. Tkach. Issues for routing in the optical layer. *IEEE Communications Magazine*, 39(2):81–87, 2001.

[70] A. Svensson. An Introduction to Adaptive QAM Modulation Schemes for Known and Predicted Channels. *Proceedings of the IEEE*, 95(12):2322–2336, 2007.

[71] J. Teng, Y. Hu, Y. Zhang, and M. Chen. Network optimization under traffic uncertainties based on SDN. In *Proc. Human Centered Computing*, pages 371–382, Cham, 2019. Springer International Publishing.

[72] A. Tomaszewski. *Optimisation of Telecommunications Transport Networks.* Prace Naukowe – Politechnika Warszawska: Elektronika. Oficyna Wydawnicza Politechniki Warszawskiej, 2013.

[73] A. Tomaszewski, M. Pióro, and M. Żotkiewicz. On the complexity of resilient network design. *Networks*, 55:109–118, 2010.

[74] C. S. K. Vadrevu, R. Wang, M. Tornatore, C. U. Martel, and B. Mukherjee. Degraded service provisioning in mixed-line-rate wdm backbone networks using multipath routing. *IEEE/ACM Transactions on Networking*, 22(3):840–849, 2014.

[75] Y. Wang, T. H. Cheng, and M. Ma. Priority and maximum revenue based routing and wavelength assignment for all-optical WDM networks. In *Proc. IEEE International Conference on Research, Innovation and Vision for the Future (RIVF 2007)*, pages 135–139, 2007.

[76] R. Wessäly. *Dimensioning survivable capacitated networks.* Cuvillier Verlag Gottingen, 2000.

[77] R. Wessäly, S. Orlowski, A. Zymolka, A. M. C. A. Koster, and C. G. Gruber. Demand-wise shared protection revisited: A new model for survivable network design. In *Proc. 2nd International Network Optimization Conference (INOC 2005)*, pages 100–105, 2005.

[78] S. Yang and F. A. Kuipers. Traffic uncertainty models in network planning. *IEEE Communications Magazine*, 52(2):172–177, 2014.

[79] N. Youssef. *Stochastic analysis via robust optimization.* PhD thesis, January 2016.