



HAL
open science

An optimization model for quadratic flow thinning – a traffic protection mechanism for FSO networks

Michal Pióro, Ilya Kalesnikau, Michael Poss

► To cite this version:

Michal Pióro, Ilya Kalesnikau, Michael Poss. An optimization model for quadratic flow thinning – a traffic protection mechanism for FSO networks. *Optical Switching and Networking*, 2019, 31, pp.168-182. 10.1016/j.osn.2018.10.002 . hal-01927518

HAL Id: hal-01927518

<https://hal.science/hal-01927518>

Submitted on 30 Mar 2021

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

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

An optimization model for quadratic flow thinning

– a traffic protection mechanism for FSO networks

Michał Pióro^{a,b,*}, Ilya Kalesnikau^a, Michael Poss^c

^a*Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland*

^b*Department of Electrical and Information Technology, Lund University, Lund, Sweden*

^c*UMR CNRS 5506 LIRMM, Université de Montpellier, 161 rue Ada, Montpellier, France*

Abstract

Flow thinning (FT) is an active traffic protection mechanism destined for communication networks with variable capacity of links, for example wireless networks. In FT, end-to-end traffic demands are equipped with dedicated logical tunnels (for example MPLS tunnels) whose maximal capacities are subject to thinning in order to follow the fluctuations of the currently available link capacities. It follows that for each demand the instantaneous traffic realized between its end nodes must accommodate to the current total capacity available on its dedicated tunnels. In the paper we develop an optimization model for network dimensioning for an important modification of FT with a feasible implementation, the so called Quadratic Flow Thinning (QFT), and present a solution algorithm based on the path generation method. We derive relevant pricing problems for basic variants of the QFT mechanism, and present a numerical study that illustrates efficiency of the optimization algorithm, as well as compares the network cost for different variants of QFT.

Keywords: Resilient networks, free space optics, variable link capacity, linear and mixed-integer programming, multicommodity flows, quadratic and affine routing, path generation.

1. Introduction

Flow thinning (FT) is an active traffic protection mechanism for communication networks whose links experience fluctuations in available capacity. Since in general the links composing such networks do not achieve their nominal (maximum) capacity simultaneously, in a typical network state only some links are fully available while on each of the remaining links only a fraction of its maximum capacity is usable. The considered fluctuations are common in wireless networks, in particular in Wireless Mesh Networks (WMN). Thus, an interesting application of FT are WMNs carrying IP packet traffic over the MPLS tunnels realized along the free space optical (FSO) links [1]. The FSO links are set up by means of laser beams between transponders in the line of sight, commonly placed on top of high buildings in a city [2].

*Corresponding author

Email addresses: m.pioro@tele.pw.edu.pl (Michał Pióro), ikalesni@mion.elka.pw.edu.pl (Ilya Kalesnikau), michael.poss@lirmm.fr (Michael Poss)

The concept of FT was introduced in [3]. FT copes with the phenomenon of variable link capacity by using logical tunnels for the end-to-end traffic demands. Each end-to-end traffic demand is assigned a set of such logical tunnels whose total capacity is dedicated to carry the demand's traffic. The nominal (i.e., maximum) capacity of the tunnels, supported by the nominal (maximum) link capacity, is subject to state-dependent thinning to account for variable capacity of the links, fluctuating below the maximum. The current capacity of the tunnels is controlled at their source nodes on the basis of the knowledge of the current link availability state. In consequence, the capacity available on the tunnels is also fluctuating below their nominal levels, and hence the instantaneous traffic sent between the demand's end nodes must accommodate to the current total capacity available on its dedicated tunnels.

In the paper we deal with a (potentially) practical version of FT where the capacity of each tunnel is thinned according to a tunnel-dependent quadratic function whose arguments are link availability coefficients (for a given link such a coefficient is the fraction of the maximal link capacity available in a given link availability state), and not in an arbitrary way as in the original FT mechanism. We consider a general form of the quadratic flow thinning formula (and call the resulting mechanism *Quadratic Flow Thinning* – QFT), and develop a network dimensioning optimization model for QFT (and, for that matter, its variants, including the so called *Affine Flow Thinning* (AFT) considered in [3, 4, 5]) – a fairly complicated issue not yet studied. As the optimization process is based on the path generation approach we derive a relevant pricing problem and present a numerical study illustrating efficiency of the resulting optimization algorithm as such, and effectiveness, in terms of the network cost, of QFT as compared with effectiveness of FT and a well known benchmark protection mechanism called Global Rerouting (called also Unrestricted Reconfiguration, see [6, 7]). All the derivations and results of the paper related to QFT are original and have not been published before.

We mention here that FT is a generalization of the *Demand-Wise Shared Protection* (DWSP) strategy [8, 9] devised for multiple total link failures (multiple total link failures are sometimes called shared risk link groups – SRLG). In fact, FT mechanism is a non-trivial extension of DWSP since thinning becomes trivial in the SRLG case: the affected tunnels are entirely lost and non-affected tunnels are preserved. Since the network dimensioning problem for DWSP is \mathcal{NP} -hard (as shown in [10]), the problem is \mathcal{NP} -hard also for FT, and hence, most likely, for QFT.

The paper is organized as follows. We start with introducing basic notation in Section 2, followed by Section 3 where a non-compact linear programming formulation of the network dimensioning problem with FT and of its dual are presented, together with a solution algorithm based on path-generation. In Section 4, we reformulate the dimensioning problem for quadratic flow thinning (the variant of FT crucial to this paper) and discuss important variants of quadratic flow thinning dealt with in the paper, including affine flow thinning. Then, in Section 5, which is crucial to the paper's considerations, we derive the pricing problems (used for path generation) for quadratic flow thinning and its variants (some derivations related to Section 5 are moved to Appendix A). Next, in Section 6, we present and discuss the results of an extensive numerical study illustrating the efficiency of the optimization algorithms and effectiveness of various variants of flow thinning. Finally, we give concluding remarks and comment on future work in Section 7.

2. Notation

Although the considerations of the paper are valid for both directed and undirected network graphs, for the space saving reason the presented optimization model will assume the undirected case. Thus, the network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is composed of the set of nodes \mathcal{V} and the set of links \mathcal{E} , where each (undirect) link represents an undirected pair $\{v, w\}$ of nodes for some $v, w \in \mathcal{V}$, $v \neq w$.

The maximal (referred to as *nominal* in the following) capacity of link $e \in \mathcal{E}$ is denoted by y_e^0 and its unit capacity cost by $\xi(e)$. It follows that the network cost (to be minimized) is expressed as $F = \sum_{e \in \mathcal{E}} \xi(e)y_e^0$. Yet, as in our network model the capacity of links can vary, typically not all links reach their nominal capacity simultaneously. Thus, at any time instant, nominal link capacities y_e^0 , $e \in \mathcal{E}$, are achieved only for a subset of links, while for the remaining links their capacity is reduced with respect to the nominal value. To handle this, we consider a set of link *availability states* \mathcal{S} . Each state $s \in \mathcal{S}$ is described by link *availability coefficients* $\alpha(e, s)$ ($0 \leq \alpha(e, s) \leq 1$), $e \in \mathcal{E}$. By definition, the capacity of link e available in state s is equal to $\alpha(e, s)y_e^0$. In the following we will also use the notion of link *failure coefficients* defined as $\beta(e, s) := 1 - \alpha(e, s)$. The subset of states s in \mathcal{S} for which a particular link $e \in \mathcal{E}$ is not fully available ($\alpha(e, s) < 1$) will be denoted by $\mathcal{S}(e)$; symmetrically, the subset of links in \mathcal{E} that are not fully available in state $s \in \mathcal{S}$ will be denoted by $\mathcal{E}(s)$. The nominal link capacities, described by the vector $y^0 := (y_e^0, e \in \mathcal{E})$, are optimization variables.

Traffic demands are represented by the set \mathcal{D} . Each demand $d \in \mathcal{D}$ is characterized by an undirected pair of the demand's end nodes $\{o(d), t(d)\}$, and given traffic volumes $h(d, s)$ to be realized in each of the considered link availability states $s \in \mathcal{S}$. Traffic volumes and link capacities are expressed in the same units (e.g., Mbps). Each demand d is equipped with a set of allowable paths $\mathcal{P}(d)$ (called the *path-set*) composed of selected elementary paths between $o(d)$ and $t(d)$. Only the paths in $\mathcal{P}(d)$ can be used to realize the demand (traffic) volumes and each such path $p \in \mathcal{P}(d)$ carries the *nominal flow* x_{dp}^0 . More precisely, each x_{dp}^0 specifies the nominal capacity (expressed in the same units as link capacity and demand volume) reserved on the tunnel realized along path $p \in \mathcal{P}(d)$. The set of all allowable paths is denoted by $\mathcal{P} := \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$. The maximum path-sets, i.e., path-sets $\mathcal{P}(d)$ containing all the elementary paths between $o(d)$ and $t(d)$, are denoted by $\widehat{\mathcal{P}}(d)$, $d \in \mathcal{D}$, and $\widehat{\mathcal{P}} := \bigcup_{d \in \mathcal{D}} \widehat{\mathcal{P}}(d)$. Clearly, the number of paths in $\widehat{\mathcal{P}}$ grows exponentially with the size of the network graph.

Since we consider elementary paths, we can identify each of them with the set $\mathcal{E}(d, p)$ of the links it traverses ($\mathcal{E}(d, p) \subseteq \mathcal{E}$). Similarly, $\mathcal{V}(d, p)$ will denote the set of nodes traversed by the path. For a given path-set $\mathcal{P}(d)$, its subset containing a given link e will be denoted with $\mathcal{R}(d, e)$, i.e., $\mathcal{R}(d, e) := \{p \in \mathcal{P}(d) : e \in \mathcal{E}(d, p)\}$, $d \in \mathcal{D}$, $e \in \mathcal{E}$. The sets of allowable paths $\mathcal{P}(d)$ are parameters in the problem formulations considered in the following.

In general, not all nominal path-flows x_{dp}^0 , $d \in \mathcal{D}$, $p \in \mathcal{P}(d)$, can be realized when the network is in a state $s \in \mathcal{S}$, since for some links the available capacity $\alpha(e, s)y_e^0$ can be decreased with respect to their nominal link capacity y_e^0 . To account for that, the nominal flows are thinned (decreased) to accommodate to the reduced link capacity. Still, the thinned flows must be sufficient to carry the demands $h(d, s)$, $d \in \mathcal{D}$. The thinned nominal path-flows for state $s \in \mathcal{S}$ are denoted by x_{dp}^s . These flows are reserved on the allowable paths for the duration of the state. As only thinning of nominal flows is allowed, the following

inequalities must hold:

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

In the following the path-flow vectors $x^0 := (x_{dp}^0, d \in \mathcal{D}, p \in \mathcal{P}(d))$ and $x^s := (x_{dp}^s, d \in \mathcal{D}, p \in \mathcal{P}(d))$, $s \in \mathcal{S}$, will be treated, together with the capacity vector y^0 , as optimization variables. For specifying the optimization variables we will use the following abbreviations: \mathbb{R} – the set of all real numbers (for continuous variables unconstrained in sign), \mathbb{R}_+ – the set of non-negative real numbers (for non-negative continuous variables), \mathbb{B} – the two-element set $\{0, 1\}$ (for binary variables).

3. FTOP – flow thinning optimization problem

3.1. Formulation of FTOP and its dual

For a given set of allowable paths $\mathcal{P} = \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$, problem FTOP is denoted by $P(\mathcal{P})$. It is represented by the non-compact link-path linear programming (LP) formulation (1) involving variables y^0, x^0 , and x^s ($s \in \mathcal{S}$). In the following formulation the quantities indicated in the square brackets are dual variables that will be used in (2).

Problem $P(\mathcal{P})$:

$$F(\mathcal{P}) = \min \sum_{e \in \mathcal{E}} \xi(e) y_e^0 \tag{1a}$$

$$[\pi_e^0 \geq 0] \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d,e)} x_{dp}^0 \leq y_e^0, \quad e \in \mathcal{E} \tag{1b}$$

$$[\lambda_d^s \geq 0] \sum_{p \in \mathcal{P}(d)} x_{dp}^s \geq h(d, s), \quad d \in \mathcal{D}, s \in \mathcal{S} \tag{1c}$$

$$[\pi_e^s \geq 0] \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d,e)} x_{dp}^s \leq \alpha(e, s) y_e^0, \quad e \in \mathcal{E}, s \in \mathcal{S}(e) \tag{1d}$$

$$[\sigma_{dp}^s \geq 0] x_{dp}^s \leq x_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S} \tag{1e}$$

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

Objective (1a) minimizes the cost of links. The first capacity constraint (1b) makes sure that the nominal link loads do not exceed the nominal link capacities. Next, the demand constraint (1c) assures that in each state $s \in \mathcal{S}$, the thinned nominal flows are sufficient to realize the volume of each demand $d \in \mathcal{D}$ assumed for this state. Then, the second capacity constraint (1d) ensures that the capacity of each link $e \in \mathcal{E}$ available in each state $s \in \mathcal{S}(e)$ is not exceeded (recall that $\mathcal{S}(e)$ denotes the set of states in which $\alpha(e, s) < 1$). Finally, inequalities (1e) ensure that in case of link failures the nominal flows are actually thinned.

Note that constraint (1d) is, for each link $e \in \mathcal{E}$, written down only for $s \in \mathcal{S}(e)$, and not for all $s \in \mathcal{S}$. This is because for any state $s \in \mathcal{S} \setminus \mathcal{S}(e)$ (in which the capacity of the considered link e is fully available, $\alpha(e, s) = 1$) constraint (1d) is implied by (1b) and (1e).

The following formulation, crucial for its solution algorithm based on path generation, describes the dual to FTOP (1):

Problem $D(\mathcal{P})$:

$$W(\mathcal{P}) = \max \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} h(d, s) \lambda_d^s \tag{2a}$$

$$[y_e^0 \geq 0] \pi_e^0 + \sum_{s \in \mathcal{S}(e)} \alpha(e, s) \pi_e^s \leq \xi(e), \quad e \in \mathcal{E} \tag{2b}$$

$$[x_{dp}^0 \geq 0] \quad \sum_{s \in \mathcal{S}} \sigma_{dp}^s \leq \sum_{e \in \mathcal{E}(d,p)} \pi_e^0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (2c)$$

$$[x_{dp}^s \geq 0] \quad \lambda_d^s \leq \sigma_{dp}^s + \sum_{e \in \mathcal{E}(d,p) \cap \mathcal{E}(s)} \pi_e^s, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S} \quad (2d)$$

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

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

Above, for each dual constraint its corresponding primal variable is given in the square brackets.

3.2. Solving FTOP through path generation

Although in the following we will assume that the number of states in \mathcal{S} is polynomial with the number of links in \mathcal{E} , the link-path LP formulation (1) of FTOP remains noncompact because of exponentially many routing paths in the maximum path-set $\widehat{\mathcal{P}}$. In fact, we cannot hope for finding a compact LP formulation since, as shown in [3], FTOP is \mathcal{NP} -hard. Since in practice formulation (1) cannot be solved directly (as we are not able to include all elementary paths in the path-sets), in order to consider all paths in $\widehat{\mathcal{P}}$ we need to apply path generation (PG, see [11, 6, 7]) – a classical technique in multicommodity flow networks related to column generation in linear programming (cf. [12]). With PG, starting from some initial path-sets $\mathcal{P}(d)$, $d \in \mathcal{D}$, we iteratively generate new paths, one per demand, and add to the path-sets those paths that may improve the solution. Generating new paths is done by solving an appropriate *pricing problem* (PP) using, as parameters, an optimal solution (λ^*, π^*) of the current dual $D(\mathcal{P})$. Adding new paths to the path-sets to the problem $P(\mathcal{P})$ consists in adding the corresponding variables and constraints to the problem formulation, i.e., we generate new columns and rows in (1).

PP for FTOP is obtained as follows (for a detailed derivation see Section 4 of [3]). Consider a given demand $d \in \mathcal{D}$ and a given path $q \in \widehat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$. Introducing such a new path q to the dual (2) introduces both new (we call them *local* in the following) dual variables σ^s , $s \in \mathcal{S}$, and new constraints (2c)–(2d). It can happen that some of the new constraints will be violated by the optimal dual solution (λ^*, π^*) of $D(\mathcal{P})$ for any non-negative values of the local variables σ^s , $s \in \mathcal{S}$. In such a case, those constraints in the set of constraints (2c)–(2d) that are violated for a given vector σ^s , $s \in \mathcal{S}$, will separate (λ^*, π^*) from the dual polyhedron of $D(\mathcal{P} \cup \{q\})$. The minimum of the sum of such violations is expressed through the quantity

$$P(q) = \min_{\sigma^s \in \mathbb{R}_+, s \in \mathcal{S}} \left\{ \max \left\{ \sum_{s \in \mathcal{S}} \sigma^s - |q|^0, 0 \right\} + \sum_{s \in \mathcal{S}} \max \left\{ \lambda_d^{s*} - \sigma^s - |q|^s, 0 \right\} \right\} \quad (3)$$

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

Thus, the pricing problem PP (defined for each demand $d \in \mathcal{D}$) consists in maximizing $P(q)$ over $q \in \widehat{\mathcal{P}}(d)$. If the resulting maximum is attained by a path $q(d) \in \widehat{\mathcal{P}}(d)$ and the maximum is greater than 0 (note that this implies $q(d) \notin \mathcal{P}(d)$), then path $q(d)$ is added to the problem: $\mathcal{P}(d) := \mathcal{P}(d) \cup \{q(d)\}$. For undirected networks the so posed PP can be stated by means of the following mixed-integer (MIP) formulation (where $\delta(v)$ denotes the set of links incident to $v \in \mathcal{V}$ and the superscript $*$ in π^*, λ^* is skipped).

$$P(u) = \min \left\{ \sum_{e \in \mathcal{E}} \pi_e^0 u_e + \sum_{s \in \mathcal{S}} \left(\sum_{e \in \mathcal{E}(s)} \pi_e^s u_e - \lambda_d^s \right) Y^s \right\} \quad (4a)$$

$$\begin{aligned} \sum_{e \in \delta(v)} u_e = 1, \quad v \in \{o(d), t(d)\}; \quad \sum_{e \in \delta(v)} u_e = 2w_v, \quad v \in \mathcal{V} \setminus \{o(d), t(d)\} \quad (4b) \\ u_e \in \mathbb{B}, e \in \mathcal{E}; \quad w_v \in \mathbb{B}, v \in \mathcal{V} \setminus \{o(d), t(d)\}; \quad Y^s \in \mathbb{B}, s \in \mathcal{S}. \end{aligned}$$

Equations (4b) define, through variables u_e (and w_v) equal to 1, a routing path q linking $o(d)$ and $t(d)$. Note that in any optimal solution, $Y^s = 0$ when the length, with respect to π^s , of the constructed path q (i.e., $|q^s|$) is greater than λ_d^s , and $Y^s = 1$ when $|q^s| < \lambda_d^s$. This is easily seen from the form of the objective function (4a), which in order to be minimized should have all the terms $\sum_{e \in \mathcal{E}} \pi_e^s u_e - \lambda_d^s$ cancelled (through $Y^s = 0$) when positive, and maintained (through $Y^s = 1$) when negative. It turns out that an optimal solution u^*, W^*, Y^*, L of (4) defines a path $q(d) := \{e \in \mathcal{E} : u_e^* = 1\}$ maximizing the value $P(q)$ defined by (3). Hence, $q(d)$, $d \in \mathcal{D}$, should be added to FTOP when $L^* > 0$.

To get rid of bi-linearities $u_e Y^s$ in the objective function (4a) we can introduce binary variables Z_e^s , $s \in \mathcal{S}, e \in \mathcal{E}(s)$, rewrite the objective as

$$\sum_{e \in \mathcal{E}} \pi_e^0 u_e + \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}(s)} \pi_e^s Z_e^s - \sum_{s \in \mathcal{S}} \lambda_d^s Y^s, \quad (5)$$

and add the constraints

$$Z_e^s \leq u_e, \quad Z_e^s \leq Y^s, \quad Z_e^s \geq 0, \quad Z_e^s \geq u_e + Y^s - 1, \quad s \in \mathcal{S}, e \in \mathcal{E}(s). \quad (6)$$

For binary u_e and Y^s these constraints force the equalities $Z_e^s = u_e Y^s$, $s \in \mathcal{S}, e \in \mathcal{E}(s)$ (hence, variables Z can formally be assumed continuous).

The PP counterpart for directed networks (with constraints (4b) replaced by the standard load-conservation equations) is formulated in [3].

4. Flow thinning formulae

A solution of FTOP minimizes the cost of network links using optimized state-dependent path-flow patterns x^s that can be thinned with respect to the nominal path-flow pattern x^0 in an arbitrary way (i.e., obeying only the inequality in (1e)). In this section we will assume that the nominal path-flows are thinned according to a fixed *flow thinning formula* of a given form depending on the link failure coefficients.

4.1. General quadratic formula (Q/G)

The most general form of the flow thinning formula considered in this paper is expressed as a *quadratic form* of the link failure coefficients. It defines the flow on path $p \in \tilde{\mathcal{P}}(d)$ of demand $d \in \mathcal{D}$ in state $s \in \mathcal{S}$ as follows:

$$x_{dp}^s = z_{dp}^0 - \sum_{e \in \tilde{\mathcal{E}}(d,p)} \beta(e, s) z_{dp}^e + \sum_{\{e, e'\} \in \tilde{\mathcal{E}}^{[2]}(d,p)} \beta(e, s) \beta(e', s) z_{dp}^{ee'}. \quad (7)$$

Above, the set $\tilde{\mathcal{E}}(d, p) \subseteq \mathcal{E}$ is a (predefined) subset of links whose failure coefficients influence the thinning formula for a given path $p \in \mathcal{P}(d)$, $d \in \mathcal{D}$. (In (7), $\tilde{\mathcal{E}}^{[2]}(d, p)$ denotes the set of all 2-element subsets of $\tilde{\mathcal{E}}(d, p)$.) For such a path, the quantities

$$z_{dp} := (z_{dp}^0; z_{dp}^e, e \in \tilde{\mathcal{E}}(d, p); z_{dp}^{ee'}, \{e, e'\} \in \tilde{\mathcal{E}}^{[2]}(d, p))$$

will be used as optimization variables which are continuous and unrestricted in sign. When optimized (and fixed) they will define flow thinning formulae: for each path $p \in \mathcal{P}(d)$, $d \in \mathcal{D}$, such a formula is a quadratic form in $\beta(e, s)$, $e \in \tilde{\mathcal{E}}(d, p)$, determining the path's flow for each state $s \in \mathcal{S}$. Note that since z_{dp} are unconstrained in sign, we could as well add the first sum on the right-hand side of (7) instead of subtracting it. We used the subtraction just to make the formula analogous to the simple flow thinning formula (S) introduced in Section 4.2.2.

Inserting the thinning formula (7) into FTOP results in the following LP formulation:

Problem P(\mathcal{P}):

$$F(\mathcal{P}) = \min \sum_{e \in \mathcal{E}} \xi(e) y_e^0 \quad (8a)$$

$$[\pi_e^0 \geq 0] \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d, e)} x_{dp}^0 \leq y_e^0, \quad e \in \mathcal{E} \quad (8b)$$

$$[\lambda_d^s \geq 0] \sum_{p \in \mathcal{P}(d)} x_{dp}^s \geq h(d, s), \quad d \in \mathcal{D}, s \in \mathcal{S} \quad (8c)$$

$$[\pi_e^s \geq 0] \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d, e)} x_{dp}^s \leq \alpha(e, s) y_e^0, \quad e \in \mathcal{E}, s \in \mathcal{S}(e) \quad (8d)$$

$$[\varphi_{dp}^s] \quad x_{dp}^s = z_{dp}^0 - \sum_{e \in \tilde{\mathcal{E}}(d, p)} \beta(e, s) z_{dp}^e + \sum_{\{e, e'\} \in \tilde{\mathcal{E}}^{[2]}(d, p)} \beta(e, s) \beta(e', s) z_{dp}^{ee'}, \quad (8e)$$

$$[\sigma_{dp}^s \geq 0] \quad x_{dp}^s \leq x_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S} \quad (8f)$$

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

$$z_{dp}^0, z_{dp}^e, z_{dp}^{ee'} \in \mathbb{R}, d \in \mathcal{D}, p \in \mathcal{P}(d), e \in \tilde{\mathcal{E}}(d, p), \{e, e'\} \in \tilde{\mathcal{E}}^{[2]}(d, p).$$

As in FTOP, the quantities indicated in the square brackets are dual variables.

4.2. Variants of the flow thinning formula

Various specific cases of the thinning formula (7) considered in the following will differ in the three aspects listed below. All twelve combinations implied by these aspects lead to valid flow thinning formulae and will be considered in the numerical study described in Section 6.

4.2.1. Form

Two basic forms of the thinning formula are considered:

- (Q) quadratic (7): $x_{dp}^s = z_{dp}^0 - \sum_{e \in \tilde{\mathcal{E}}(d, p)} \beta(e, s) z_{dp}^e + \sum_{\{e, e'\} \in \tilde{\mathcal{E}}^{[2]}(d, p)} \beta(e, s) \beta(e', s) z_{dp}^{ee'}$.

- (A) affine (no quadratic terms): $x_{dp}^s = z_{dp}^0 - \sum_{e \in \tilde{\mathcal{E}}(d, p)} \beta(e, s) z_{dp}^e$.

4.2.2. Structure

The structure of the thinning formula is twofold:

- (G) general: no constraints on z_{dp}^0 and z_{dp} (as in (7)).

- (S) simple: $z_{dp}^0 = x_{dp}^0$ and nonnegative z_{dp} and, i.e.,

$$x_{dp}^s = x_{dp}^0 - \sum_{e \in \tilde{\mathcal{E}}(d, p)} \beta(e, s) z_{dp}^e + \sum_{\{e, e'\} \in \tilde{\mathcal{E}}^{[2]}(d, p)} \beta(e, s) \beta(e', s) z_{dp}^{ee'}, \quad z_{dp} \geq 0. \quad (9)$$

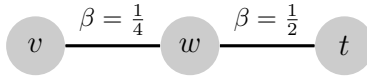


Figure 1: A two-link network.

Above z_{dp} denotes (for $p \in \mathcal{P}(d)$, $d \in \mathcal{D}$) the vector of all z_{dp}^e (in affine formulae) or the vector of all z_{dp}^e and $z_{dp}^{ee'}$ (in quadratic formulae).

With nonnegative coefficients z_{dp} in the simple formula (S), the signs of the summations on the right-hand side of (9) are important. To illustrate this point let us consider a very simple network with two links ($e = 1$ between nodes v and w , and $e = 2$ between nodes w and t) in series depicted in Figure 1, and one demand between nodes v and t . Both links have capacity 1. We consider four states in \mathcal{S} described in the first three columns in Table 1, where the link failure coefficients for a particular state $s \in \mathcal{S}$ are specified in column 2 (for link $e = 1$) and in column 3 (for link $e = 2$). Clearly, the maximum nominal flow for the demand is $x^0 = 1$ (the subscripts in x^0 are skipped as there is only one demand $d = 1$ with only one path $p = 1$ composed of links $e = 1$ and $e = 2$). With the simple affine formula (with $\tilde{\mathcal{E}}(d, p) = \mathcal{E}$) the maximum feasible flows in the remaining states (given in column 4) are achieved with $z^1 = z^2 = 1$ (i.e., $x^s = 1 - \beta(1, s) - \beta(2, s)$). For the simple quadratic formula the maximum flows (given in column 5) are achieved with $z^1 = z^2 = z^{12} = 1$ (i.e., $x^s = 1 - \beta(1, s) - \beta(2, s) + \beta(1, s) \cdot \beta(2, s)$). Note that the maximum flow in state $s = 3$ is greater for the quadratic formula ($x^3 = \frac{3}{8}$) than for the affine formula ($x^3 = \frac{1}{4}$). Note also that for FT (column 6) the maximum flow in state $s = 3$ is equal to $\frac{1}{2}$ (of course, this value cannot be further increased due to capacity constraints). The presented example clearly shows why the sign before the first sum in (9) must be subtracted from x^0 (to have feasible flows), and the second sum added (to maximize the achievable flows).

s	$\beta(1, s)$	$\beta(2, s)$	AFT: x^s	QFT: x^s	FT: x^s
0	0	0	1	1	1
1	$\frac{1}{4}$	0	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$
2	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
3	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{1}{2}$

Table 1: Affine and quadratic flows with simple formula.

4.2.3. Range

The *range of the thinning formula* is the set $\tilde{\mathcal{E}}(d, p) \subseteq \mathcal{E}$. As discussed in [3], typical examples of the sets $\tilde{\mathcal{E}}(d, p)$ are:

- ($\mathcal{E}(d, p)$) path's links: $\tilde{\mathcal{E}}(d, p) = \mathcal{E}(d, p)$ – the formula depends only on the failure coefficients of the links along the path.
- ($\mathcal{E}^+(d, p)$) link's incident to path's nodes: $\tilde{\mathcal{E}}(d, p) = \bigcup_{v \in \mathcal{V}(d, p)} \delta(v)$ (where $\delta(v)$ is the set of links incident with v) – the formula depends on the failure coefficients of the links

incident to the nodes along the path; in the following the set of links $\bigcup_{v \in \mathcal{V}(d,p)} \delta(v)$ will be denoted with $\mathcal{E}^+(d, p)$.

- (\mathcal{E}) all links: $\tilde{\mathcal{E}}(d, p) = \mathcal{E}$ – the formula depends on the failure coefficients of all links.

5. Pricing problems

The counterparts of FTOP corresponding to the thinning formulae described in Section 4.2 are solved through path-generation (see Section 3.2). For that, appropriate pricing problems are required. Below we will derive the pricing problems (PP) for the simple affine formula case (A/S) and for the general affine formula case (A/G). Both of them turn out to be more complicated than their FT counterpart derived in Section 3.2. We will also discuss the pricing problems for the quadratic forms.

5.1. PP for simple affine formula

We start with deriving the pricing problem for the simple affine formula A/S.

5.1.1. A/S – PP derivation for arbitrary range $\tilde{\mathcal{E}}(d, p)$

The optimization problem for the simple affine formula (a counterpart of (8)), i.e.,

Problem P(\mathcal{P}):

$$F(\mathcal{P}) = \min \sum_{e \in \mathcal{E}} \xi(e) y_e^0 \quad (10a)$$

$$[\pi_e^0 \geq 0] \quad \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d,e)} x_{dp}^0 \leq y_e^0, \quad e \in \mathcal{E} \quad (10b)$$

$$[\lambda_d^s \geq 0] \quad \sum_{p \in \mathcal{P}(d)} x_{dp}^s \geq h(d, s), \quad d \in \mathcal{D}, s \in \mathcal{S} \quad (10c)$$

$$[\pi_e^s \geq 0] \quad \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d,e)} x_{dp}^s \leq \alpha(e, s) y_e^0, \quad e \in \mathcal{E}, s \in \mathcal{S}(e) \quad (10d)$$

$$[\varphi_{dp}^s] \quad x_{dp}^s = x_{dp}^0 - \sum_{e \in \tilde{\mathcal{E}}(d,p)} \beta(e, s) z_{dp}^e, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S} \quad (10e)$$

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

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

has the following dual:

Problem D(\mathcal{P}):

$$W(\mathcal{P}) = \max \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} h(d, s) \lambda_d^s \quad (11a)$$

$$[y_e^0 \geq 0] \quad \pi_e^0 + \sum_{s \in \mathcal{S}(e)} \alpha(e, s) \pi_e^s \leq \xi(e), \quad e \in \mathcal{E} \quad (11b)$$

$$[x_{dp}^0 \geq 0] \quad \sum_{s \in \mathcal{S}} \varphi_{dp}^s \leq \sum_{e \in \mathcal{E}(d,p)} \pi_e^0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (11c)$$

$$[x_{dp}^s \geq 0] \quad \lambda_d^s \leq \varphi_{dp}^s + \sum_{e \in \mathcal{E}(d,p) \cap \mathcal{E}(s)} \pi_e^s, \quad s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (11d)$$

$$[z_{dp}^e \geq 0] \quad \sum_{s \in \mathcal{S}} \beta(e, s) \varphi_{dp}^s \geq 0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), e \in \tilde{\mathcal{E}}(d, p) \quad (11e)$$

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

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

As in the derivation of PP for FT in Section 3.2, consider a given demand $d \in \mathcal{D}$ and a given path $q \in \widehat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$. Adding such a new path q to the dual (11) introduces new local variables φ^s , $s \in \mathcal{S}$, and new constraints (11c)–(11e). If the new constraints (11c)–(11e) are violated by the optimal solution (λ^*, π^*) of the dual (11) for any values of the local variables, then the violated constraints will separate (λ^*, π^*) from the dual polyhedron. The minimum of the sum of such violations (taken over all feasible local dual variables) is expressed through the quantity

$$P(q) = \min_{\varphi^s \in \mathbb{R}, s \in \mathcal{S}} \left\{ \max \left\{ \sum_{s \in \mathcal{S}} \varphi^s - |q|^0, 0 \right\} + \sum_{s \in \mathcal{S}} \max \left\{ \lambda_d^{s*} - |q|^s - \varphi^s, 0 \right\} + \sum_{e \in \widetilde{\mathcal{E}}(d,q)} \max \left\{ - \sum_{s \in \mathcal{S}} \beta(e, s) \varphi^s, 0 \right\} \right\}.$$

(recall that $|q|^0 := \sum_{e \in \mathcal{E}(d,q)} \pi_e^{0*}$ and $|q|^s := \sum_{e \in \mathcal{E}(d,q) \cap \mathcal{E}(s)} \pi_e^{s*}$, $s \in \mathcal{S}$) which can be calculated as the solution of the following LP.

$$P(q) = \min \left\{ Z + \sum_{s \in \mathcal{S}} Y_s + \sum_{e \in \widetilde{\mathcal{E}}(d,q)} X_e \right\} \quad (13a)$$

$$[g \geq 0] \quad Z \geq \sum_{s \in \mathcal{S}} \varphi^s - |q|^0 \quad (13b)$$

$$[a_s \geq 0] \quad Y_s \geq \lambda_d^{s*} - |q|^s - \varphi^s, \quad s \in \mathcal{S} \quad (13c)$$

$$[b_e \geq 0] \quad X_e \geq - \sum_{s \in \mathcal{S}} \beta(e, s) \varphi^s, \quad e \in \widetilde{\mathcal{E}}(d, q) \quad (13d)$$

$$Z \in \mathbb{R}_+; Y_s \in \mathbb{R}_+, s \in \mathcal{S}; X_e \in \mathbb{R}_+, e \in \widetilde{\mathcal{E}}(d, q); \varphi^s \in \mathbb{R}, s \in \mathcal{S}. \quad (13e)$$

In order to solve PP, we first notice that we can as well maximize the value of $P(q)$ over all paths $q \in \widehat{\mathcal{P}}(d)$ since for all $p \in \mathcal{P}(d)$ the value $P(p)$ is equal to zero, and a new path q will be added to the current list $\mathcal{P}(d)$ only when $P(q)$ is greater than zero. Hence, PP turns out to be the max-min problem of the form

$$\max_{q \in \widehat{\mathcal{P}}(d)} \min_{X, Y, Z, \varphi} \left\{ Z + \sum_{s \in \mathcal{S}} Y_s + \sum_{e \in \widetilde{\mathcal{E}}(d,q)} X_e \right\} \quad (14a)$$

$$\text{subject to constraints (13b)-(13e)} \quad (14b)$$

(where Y and X are appropriate vectors of variables). As such, problem (14) cannot be solved directly. A way to resolve this issue is to consider the problem dual to (13), formulated using the dual variables specified in the square brackets in constraints (13b)–(13d):

$$P(q) = \max \left\{ -g|q|^0 + \sum_{s \in \mathcal{S}} (g - \sum_{e \in \widetilde{\mathcal{E}}(d,q)} \beta(e, s) b_e) (\lambda_d^{s*} - |q|^s) \right\} \quad (15a)$$

$$g \leq 1; b_e \leq 1, e \in \widetilde{\mathcal{E}}(d, q); \sum_{e \in \widetilde{\mathcal{E}}(d,q)} \beta(e, s) b_e \leq g, s \in \mathcal{S} \quad (15b)$$

$$g \in \mathbb{R}_+; b_e \in \mathbb{R}_+, e \in \widetilde{\mathcal{E}}(d, q). \quad (15c)$$

Note that in (15), dual variables a_s , $s \in \mathcal{S}$, have been eliminated. This is possible because variables x_{dp}^s , for all d, p, s , are auxiliary, and can be expressed through x_{dp}^0 and z_{dp}^e , $e \in \widetilde{\mathcal{E}}(d, p)$, on the condition $x_{dp}^0 \geq \sum_{e \in \widetilde{\mathcal{E}}(d,p)} \beta(e, s) z_{dp}^e$.

Now we introduce binary quantities u_e , $e \in \mathcal{E}$, that represent the path q : $\mathcal{E}(d, q) = \{e \in \mathcal{E} : u_e = 1\}$. Also, from now on we will skip superscript $*$ in π^* and λ^* in the derivations,

i.e., we will denote the optimal dual solution simply by π, λ . Hence, $|q|^0 = \sum_{e \in \mathcal{E}} \pi_e^0 u_e$ and $|q|^s = \sum_{e \in \mathcal{E}(s)} \pi_e^s u_e$, $s \in \mathcal{S}$, and, in consequence, we can rewrite (15):

$$P(u) = \max \left\{ -g(\sum_{e \in \mathcal{E}} \pi_e^0 u_e) + \sum_{s \in \mathcal{S}} \left(g - \sum_{e' \in \tilde{\mathcal{E}}(d,q)} \beta(e', s) b_{e'} \right) (\lambda_d^s - \sum_{e \in \mathcal{E}(s)} \pi_e^s u_e) \right\} \quad (16a)$$

$$g \leq 1; b_{e'} \leq 1, e' \in \tilde{\mathcal{E}}(d, q); \sum_{e' \in \tilde{\mathcal{E}}(d,q)} \beta(e', s) b_{e'} \leq g, s \in \mathcal{S} \quad (16b)$$

$$g \in \mathbb{R}_+; b_{e'} \in \mathbb{R}_+, e' \in \tilde{\mathcal{E}}(d, q). \quad (16c)$$

Finally, by transforming objective (16a), treating u_e , $e \in \mathcal{E}$, as variables, and adding appropriate path-defining constraints (for undirected links) we arrive at the following formulation of the pricing problem for problem (10).

$$P = \max \left\{ -\sum_{e \in \mathcal{E}} \pi_e^0 g u_e + (\sum_{s \in \mathcal{S}} \lambda_d^s) g - \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}(s)} \pi_e^s g u_e + \sum_{s \in \mathcal{S}} \sum_{e' \in \tilde{\mathcal{E}}(d,q)} \sum_{e \in \mathcal{E}(s)} \beta(e', s) \pi_e^s b_{e'} u_e - \sum_{s \in \mathcal{S}} \sum_{e' \in \tilde{\mathcal{E}}(d,q)} \lambda_d^s \beta(e', s) b_{e'} \right\} \quad (17a)$$

$$\sum_{e \in \delta(v)} u_e = 1, v \in \{s(d), t(d)\}; \sum_{e \in \delta(v)} u_e = 2z_v, v \in \mathcal{V} \setminus \{s(d), t(d)\} \quad (17b)$$

$$g \leq 1; b_{e'} \leq 1, e' \in \tilde{\mathcal{E}}(d, q); \sum_{e' \in \tilde{\mathcal{E}}(d,q)} \beta(e', s) b_{e'} \leq g, s \in \mathcal{S} \quad (17c)$$

$$g \in \mathbb{R}_+; b_{e'} \in \mathbb{R}_+, e' \in \tilde{\mathcal{E}}(d, q); u_e \in \mathbb{B}, e \in \mathcal{E}; z_v \in \mathbb{B}, v \in \mathcal{V}, \quad (17d)$$

where $\delta(v)$ denotes the set of all links incident to node $v \in \mathcal{V}$. In the formulation, (17b) are path-defining constraints so that an optimal solution u^* of (17) defines the path q with $\mathcal{E}(d, q) := \{e \in \mathcal{E} : u_e^* = 1\}$ (and, for that matter, with $\mathcal{V}(d, q) := \{v \in \mathcal{V} : z_v^* = 1\}$) that should be added to (10), provided $P(q) > 0$. Note that any feasible set $\mathcal{U} := \{e \in \mathcal{E} : u_e = 1\}$ will necessarily contain an elementary path between $s(d)$ and $t(d)$ and, possibly, a set of disjoint isolated loops. The path is elementary because variables z_v , $v \in \mathcal{V}$, are binary (note that assuming these variables to be integer-valued would allow the path to contain loops as well). In fact, such loops may appear in optimal PP solutions because of a tradeoff related to the number of elements in the set \mathcal{U} : the more elements the more flexible formulae (10e), and the less elements the lower link capacities.

Certainly, formulation (17) can be easily modified for directed networks (with links in \mathcal{E} directed) using the node-arc formulation with load conservation equations instead of (17b):

$$\sum_{e \in \delta^-(t(d))} u_e - \sum_{v \in \mathcal{V} \setminus \{s(d), t(d)\}} u_e = 1 \quad (18a)$$

$$\sum_{e \in \delta^+(v)} u_e - \sum_{a \in \delta^-(v)} u_e = 0, \quad e \in \mathcal{E}. \quad (18b)$$

In fact, formulation (17) is incomplete as it makes use of the set $\tilde{\mathcal{E}}(d, q)$ (specifying the range of the thinning formula) which in general is path-dependent and not known in advance. It also contains multiplication of variables. Thus, in order to obtain a proper MIP formulation, an appropriate representation of $\tilde{\mathcal{E}}(d, q)$ is required (by means of optimization variables), along with linearization of variables' multiplication. The particular forms of such a MIP depend on the assumed thinning formula range, as shown below.

5.1.2. Case $\tilde{\mathcal{E}}(d, p) = \mathcal{E}(d, p)$

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

$$P = \max \left\{ - \sum_{e \in \mathcal{E}} \pi_e^0 G_e + \left(\sum_{s \in \mathcal{S}} \lambda_d^s \right) g - \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}(s)} \pi_e^s G_e + \right. \\ \left. + \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}(s)} \sum_{e' \in \mathcal{E}} \pi_e^s \beta(e', s) T_{e'e} - \sum_{s \in \mathcal{S}} \sum_{e' \in \mathcal{E}} \lambda_d^s \beta(e', s) B_{e'} \right\} \quad (19a)$$

$$\sum_{e \in \delta^-(v)} u_e = 1, v \in \{s(d), t(d)\}; \quad \sum_{e \in \delta^-(v)} u_e = 2z_v, v \in \mathcal{V} \setminus \{s(d), t(d)\} \quad (19b)$$

$$B_e \leq u_e, B_e \leq b_e, B_e \geq b_e + u_e - 1, \quad e \in \mathcal{E} \quad (19c)$$

$$g \leq 1; \quad b_e \leq 1, e \in \mathcal{E}; \quad \sum_{e \in \mathcal{E}} \beta(e, s) B_e \leq g, \quad s \in \mathcal{S} \quad (19d)$$

$$G_e \leq g, G_e \leq u_e, G_e \geq g + u_e - 1, \quad e \in \mathcal{E} \quad (19e)$$

$$T_{e'e} \leq u_{e'}, T_{e'e} \leq u_e, T_{e'e} \leq b_{e'}, T_{e'e} \geq b_{e'} + u_{e'} + u_e - 2, \quad e', e \in \mathcal{E} \quad (19f)$$

$$g \in \mathbb{R}_+; \quad b_e, B_e, G_e \in \mathbb{R}_+, \quad e \in \mathcal{E}; \quad T_{ee'} \in \mathbb{R}_+, \quad e, e' \in \mathcal{E} \quad (19g)$$

$$u_e \in \{0, 1\}, \quad e \in \mathcal{E}; \quad z_v \in \{0, 1\}, \quad v \in \mathcal{V} \setminus \{s(d), t(d)\}. \quad (19h)$$

The extra variables in the above formulation are used to eliminate multiplication of variables and have the following meaning: $B_e = b_e \cdot u_e$, $G_e = g \cdot b_e$, $T_{e'e} = b_{e'} \cdot u_{e'} \cdot u_e$.

In order to eliminate the (isolated) loops in $\{e \in \mathcal{E} : u_e = 1\}$ we introduce the bi-directed version $\mathcal{G}' = (\mathcal{V}, \mathcal{A})$ of the original undirected network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of (directed) arcs is defined as $\mathcal{A} := \{(v, w), (w, v) : \{v, w\} \in \mathcal{E}\}$, i.e., each undirected link $e \in \mathcal{E}$ is substituted by two oppositely directed arcs $a'(e)$ and $a''(e)$. Below, $\delta^-(v)$ denotes the set of all arcs incoming to node v , and $\delta^+(v)$ – the set of all arcs outgoing from node v ($v \in \mathcal{V}$). Adding aggregated arc-flow variables $f_a \geq 0$, $a \in \mathcal{A}$, and the following constraints to (19) will eliminate the loops in question.

$$\sum_{a \in \delta^-(t(d))} f_a = \sum_{v \in \mathcal{V} \setminus \{s(d), t(d)\}} z_v \quad (20a)$$

$$\sum_{a \in \delta^+(s(d))} f_a = \sum_{a \in \delta^-(s(d))} f_a \quad (20b)$$

$$\sum_{a \in \delta^+(v)} f_a = \sum_{a \in \delta^-(v)} f_a + z_v, \quad v \in \mathcal{V} \setminus \{s(d), t(d)\} \quad (20c)$$

$$f_{a'(e)} + f_{a''(e)} \leq (|\mathcal{V}| - 2)u_e, \quad e \in \mathcal{E}. \quad (20d)$$

We note that this case was considered in [5].

5.1.3. Case $\tilde{\mathcal{E}}(d, p) = \mathcal{E}^+(d, p)$

The version of PP for $\tilde{\mathcal{E}}(d, p) = \mathcal{E}^+(d, p)$ (more complicated than (19)) is given in formulation (A.1) presented in Section Appendix A.1.1 of Appendix A. We note that formulation (A.1) requires loop-elimination constraints (20). It is also computationally less efficient than its counterparts for $\tilde{\mathcal{E}}(d, p) = \mathcal{E}(d, p)$ and $\tilde{\mathcal{E}}(d, p) = \mathcal{E}$. For this reason, the cases for range $\mathcal{E}^+(d, p)$ reported in Section 6 were optimized using the sets of paths generated for the corresponding cases assuming range \mathcal{E} .

5.1.4. Case $\tilde{\mathcal{E}}(d, p) = \mathcal{E}$

The PP formulation for the considered full range case is given by (A.2) in Section Appendix A.1.2. In the considered full range case, the PP formulation becomes simpler than (A.2) 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 $u_{e'} b_{e'}$ rather than $u_{e'} u_e b_{e'}$. Moreover, the loops in the optimal path will not appear due to the maximal range of the thinning formula. We observe that formulation (A.2) is simpler and computationally more efficient than its counterpart (19) for $\tilde{\mathcal{E}}(d, p) = \mathcal{E}(d, p)$.

5.2. PP for general affine formula A/G

Now we consider problem formulation (10) with equation (10e) substituted by its general affine version

$$x_{dp}^s = z_{dp}^0 - \sum_{e \in \tilde{\mathcal{E}}(d,p)} \beta(e, s) z_{dp}^e \quad (21)$$

where the affine coefficients z_{dp}^0, z_{dp}^e , $d \in \mathcal{D}$, $p \in \mathcal{P}(d)$, $e \in \tilde{\mathcal{E}}(d,p)$, are continuous and unrestricted in sign. Not surprisingly, it turns out that pricing for such obtained formulations (for different thinning formula ranges) is more complicated than for their counterparts described in Section 5.1 and Appendix A.1.

In Appendix A.2.1 we give a general derivation of PP in the considered A/G case, following the derivation for A/S presented in Section 5.1.1. Then, in Section Appendix A.2.2, we present the proper MIP formulation (A.7) for the range $\tilde{\mathcal{E}}(d,p) = \mathcal{E}(d,p)$. As for the simple affine formula case, PP formulations for the two remaining cases of thinning formula range are analogous to (A.7). As before, the PP formulation for $\tilde{\mathcal{E}}(d,p) = \mathcal{E}^+(d,p)$ is more complicated (and less time consuming) while the PP formulation for $\tilde{\mathcal{E}}(d,p) = \mathcal{E}$ is less complicated (and less time consuming) than (A.7). Besides, formulations for $\tilde{\mathcal{E}}(d,p) = \mathcal{E}(d,p)$ and $\tilde{\mathcal{E}}(d,p) = \mathcal{E}^+(d,p)$ require loop-elimination.

5.3. PP for quadratic formulae

The formulations of pricing problems for the thinning formulae containing the quadratic terms as in the general formula Q/G (8e) or its simplified version Q/S can be derived analogously as for the affine cases. Yet, the resulting PP formulations require much more variables and become computationally inefficient. For example, products of the form $u_e \cdot u_{e'} \cdot u_{e''} \cdot b_e$ appearing in the non-linear versions will have to be eliminated, and this will involve $|\mathcal{E}|^3$ auxiliary variables. To resolve this issue, we simply use the sets of paths generated for the affine counterparts of the quadratic problem in question (see Section 6.2.3).

6. Numerical study

In this section we will present optimization results obtained for medium size network instance. The study illustrates efficiency of the considered versions of the flow thinning formulae (in terms of the network cost), as well as of the path generation algorithm (in terms of the computation time).

6.1. Network instance

The study was performed for a network described in the communication network instances library SNDlib (sndlib.zib.de, see [13]) under the name *polska*. In the study we used the data from the first instance (i.e., *polska-D-B-M-N-C-A-N-N*). The network is composed of $|V| = 12$ nodes, $|E| = 18$ (undirected) links, and $|D| = 66$ (undirected) demands. The link unit capacity cost $\xi(e)$ is equal to the link's `<module_cost>` divided by `<module_capacity>` for the first pair of these values. Thus, for the first link (between Gdansk and Warsaw) $\xi(1) = 156/155 = 1.0645$, and the entire vector ξ of the unit 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 note that the same network instance was used in the case study for A/S presented in [4].

We consider three link availability state scenarios:

- (SL) Single link failure scenario: SL contains the nominal state $s(0)$ (all links fully available), and all states with exactly one failing 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 all single link failure states s . Link failure coefficients of the affected links are assumed to be $\beta(e, s) = 0.5$.
- (DL) Double link failure scenario: DL contains SL and all states with exactly two failing links (hence DL contains $|\mathcal{E}|(|\mathcal{E}| - 1)/2 + |\mathcal{E}| + 1$ states). Link failure coefficients in the double link failure states are assumed to be $\beta(e, s) = 0.4$. For the double link failure states the demand volumes are reduced to 95% of $h(d, s(0))$ for all $d \in \mathcal{D}$.
- (TL) Triple link failure scenario: TL contains SL and DL, and all states with exactly three failing links (TL contains $|\mathcal{E}|(|\mathcal{E}| - 1)(|\mathcal{E}| - 2)/6 + |\mathcal{E}|(|\mathcal{E}| - 1)/2 + |\mathcal{E}| + 1$ states). Link failure coefficients of the failing links in the triple link failure states are assumed to be $\beta(e, s) = 0.3$ and the reduction of $h(d, s(0))$ is equal to 90%.

6.2. Network cost and algorithm efficiency

The results of the numerical study for *polska* are presented in Tables 2, 3, and 4. All the reported calculations 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 12.4.0.0. In the calculations, the initial path-lists for the PG algorithm contain only one path per demand – the shortest path with respect to the link unit costs $\xi(e)$, $e \in \mathcal{E}$ (66 paths in total). Note that assigning the entire nominal demand volume $h(d, s(0))$ for each demand d to its shortest path will result in the optimally dimensioned network when protection is not considered. For the examined network example this cost turns out to be $F^0 = 30275$ (as indicated in Table 2).

6.2.1. Results for GR and FT

Table 2 compares results obtained by means of the PG algorithm for GR (Global Rerouting, see [6, 7]), and FT (Flow Thinning). The results for FT are obtained by solving problem (1). The analogous problem for GR is obtained from (1) simply by deleting constraints (1b) and (1e). Note that because of that GR, as the least constrained, and thus least costly, protection mechanism is used as a benchmark since it provides the lower bound for the network cost achievable with any other mechanism.

For each failure scenario (SL, DL, TL), the rows in Table 2 contain the results for GR 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 PG algorithm
- 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 GR ($\Delta F^* = \frac{F^*(FT) - F^*(GR)}{F^*(GR)} \times 100\%$)
- $|\mathcal{P}^*|$: final number of generated paths
- $|\mathcal{P}^u|$: number of paths used in the final solution (i.e., paths with positive optimal x_{dp}^0)
- number of iterations (iter), total computation time (total), time per iteration (t/iter), computation time spent in problem P(\mathcal{P}) (called the master problem, MP, in this context) per iteration (t/MP), and pricing problem computation time per iteration (t/PP).

		F^0	F^*	$\Delta F^0[\%]$	$\Delta F^*[\%]$	$ \mathcal{P}^* $	$ \mathcal{P}^u $	iter	total	t/iter	t/MP	t/PP
SL	GR	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	GR	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	GR	30275	36630	21	–	249	249	4	4m13s	1m3s	45s	18s
	FT	30275	37200	23	2	294	156	8	24m3s	3m	2m28s	32s

Table 2: Results for GR and FT.

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

6.2.2. Results for AFT

Table 3 gives the results for AFT, i.e., for flow thinning with affine thinning formulae. For each failure scenario, all six combinations of the range and the structure considered for the thinning formulae of form A (see Section 4.2) are examined (and specified in columns “range” and “structure”). Now, column “ ΔF^* ” expresses the increase in the network cost for a given variant of AFT with respect to the corresponding solution F^* for FT given in Table 2. The meaning of the remaining columns is the same as before.

Regarding the network cost F^* achievable by the considered AFT formulae, we first of all observe that in the SL case (single link failures), the minimal cost (i.e., the cost of FT) is achieved already with the simplest form of the thinning formula, that is with $A/S/\mathcal{E}(d, p)$.

	range	structure	F^*	$\Delta F^*[\%]$	$ \mathcal{P}^* $	$ \mathcal{P}^u $	iter	total	t/iter	t/MP	t/PP
SL	$\mathcal{E}(d, p)$	G	40236	0	275	125	8	7m34s	57s	0.1s	56.9s
		S	40236	0	276	123	9	2m12s	15s	0.06s	14.94s
	$\mathcal{E}^+(d, p)$	G	40236	0	283	128	–	0.3s	–	–	–
		S	40236	0	274	124	–	0.3s	–	–	–
	\mathcal{E}	G	40236	0	283	125	9	6m55s	46s	0.1s	45.9s
		S	40236	0	274	124	9	2m5s	14s	0.1s	13.9s
DL	$\mathcal{E}(d, p)$	G	47174	18	285	161	9	11m59s	1m19s	3s	1m16s
		S	47174	18	283	161	9	4m34s	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	13m51s	1m43s	25s	1m18s
		S	47174	18	187	161	7	2m21s	20s	1.6s	18.4s
TL	$\mathcal{E}(d, p)$	G	41863	13	232	161	10	27m21s	2m44s	29s	2m15s
		S	41863	13	223	162	9	12m52s	1m25s	19s	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	33m15s	4m45s	2m6s	2m39s
		S	41863	13	165	160	6	5m51s	58s	19s	39s

Table 3: Results for affine thinning formulae (A).

The reason is as follows. Suppose x_{dp}^0 and x_{dp}^s are optimal for FT. Then the thinning formula

$$x_{dp}^s = x_{dp}^0 - \sum_{e \in \mathcal{E}(d,p)} \beta(e, s) z_{dp}^e, \quad \text{where } z_{dp}^e := \frac{x_{dp}^0 - x_{dp}^{s(e)}}{\beta(e, s)}, \quad e \in \mathcal{E}(d, p)$$

will give the optimal values of x_{dp}^s , 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 failure coefficient $\beta(e, s(e)) > 0$ (in the remaining states, i.e., for $s \in \mathcal{S} \setminus \{s(e)\}$, $\beta(e, s) = 0$). (Recall that in the considered example $\beta(e, s(e)) = 0.5$.)

For DL and for TL this is no longer the case. Assuming the range $\mathcal{E}(d, p)$, the increase in F^* given by ΔF^* is quite high: for DL it is equal to 18% for both structures G (general) and S (simple) while for TL it is equal to 13% for both structures. 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 is roughly 2 times smaller than for S, while for \mathcal{E} it is 3.25 (TL) to 3.6 (DL) times smaller. Finally, we note that the cost increase ΔF^* for the case A/G/ \mathcal{E} is only 4%, while for A/G/ $\mathcal{E}^+(d, p)$ it rises to 7%.

Note that for $\mathcal{E}^+(d, p)$ Table 3 reports one computation time (t/MP). This is because this particular range case was not treated by the path generation algorithm because of excessive computation time (of the order of hours) required for the pricing problem (see the discussion in the last paragraph of Section 5.1.3). Instead, we just solved the appropriate variant of the master problem (10), assuming the path-sets \mathcal{P}^* generated for \mathcal{E} . For the same reason the number of iterations is not reported (– in column “iter”). Although this makes the optimization approach heuristic, the obtained solutions are clearly near-optimal.

It is also interesting to note that the case 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 A/S/ $\mathcal{E}(d, p)$ and A/G/ $\mathcal{E}(d, p)$ happen to be equal. Yet, this is not always true. We have found some (randomly generated) state scenarios (with the states containing 4-9 links affected with the failure coefficient equal to 0.4) for the *polska* network where the costs for these two cases are different. Still, the difference is small, of the order of 1.5%.

The numbers of paths generated by the PG algorithm (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 in pricing is also reasonable although this requires solving MIP formulations. The time spent in the 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.

6.2.3. Results for QFT

	range	structure	F^*	$\Delta F^*[\%]$	$ \mathcal{P}^* $	$ \mathcal{P}^u $	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

Table 4: Results for quadratic thinning formulae (Q).

Table 4 shows the results for QFT, i.e., for flow thinning with quadratic thinning formulae. As for the AFT case, all the six combinations of range/structure (see Section 4.2) are considered for each failure scenario. As already mentioned in Section 5.3, all the QFT cases were directly optimized through solving an appropriate version of the master problem (8) for

the sets of paths obtained for the corresponding AFT cases – this is due to excessive pricing time for QFT. (Therefore, Table 4 has a reduced set of columns as compared with Table 3.) This near-optimal procedure has already been applied for the $\mathcal{E}^+(d, p)$ range of the AFT.

Clearly, the QFT cost values given in Table 4 for SL are the same as for AFT (and, for that matter, for FT), since AFT is a special case of QFT. Note that corresponding total solution times are much shorter since no pricing is involved.

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

6.3. Implementation issues and suggested formula

The flow thinning mechanisms assume that the capacity of each (MPLS) tunnel is controlled at its source node by a packet admission control mechanism based on the on-line knowledge of the currently available link capacity. Therefore, some signalling mechanism for interchanging information concerning the current state of link capacities must be applied. In the case of FT, and of QFT and AFT 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 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, QFT/ \mathcal{E} and AFT/ \mathcal{E} is problematic.

On the other hand, in the case of QFT and AFT with the $\mathcal{E}(d, p)$ range, the signalling in question is simple and fast: when availability of a link is changed, an appropriate message is propagated backwards to the source nodes of all the tunnels 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 in the set \mathcal{S} . For such a purpose the most cost effective flow thinning mechanism, i.e., FT, is not satisfactory, as it does not provide consistent means for this kind of 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 QFT and AFT mechanisms can simply apply the flow thinning formula also for the unforeseen states, modifying the tunnel capacity to 0 or x_{dp}^0 when necessary, i.e., when the value obtained from the flow thinning formula is below 0 or above x_{dp}^0 , respectively. (Traffic efficiency of such extended thinning is addressed in Section 8.2 of [3] for AFT/G.) We may also expect that the larger the range of the affine formula, the better approximation of the proper tunnel capacity in the states not considered in optimization – this, however, needs to be verified.

Here, the simple structure (S) of the flow thinning formula seems more safe than the 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 benchmark GR mechanism, which provides the lower bound on the link capacity cost for *any* protection/restoration mechanism (see [7]). Tables 2, 3 and 4 show that the network cost achieved with QFT/G with range \mathcal{E} is almost the same as the cost of FT, while the cost for QFT/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 QFT/G with the smallest range $\mathcal{E}(d, p)$ is up to 7.4% higher than the cost for FT. As far as AFT is concerned, the network cost increase is significantly higher than for QFT.

Having in mind that the optimization of flow thinning is performed off-line, the total computation times are acceptable in all the considered cases, besides QFT with the ranges \mathcal{E} and $\mathcal{E}^+(d, p)$. This, 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.

	range	structure	implementation	states not in \mathcal{S}	capacity cost	optimization time
GR	\mathcal{E}	–	infeasible	not covered	very low	very short
FT	\mathcal{E}	–	problematic	not covered	low	medium
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+
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

Table 5: Summary of the results.

The above observations (summarized in Table 5) suggest that QFT/G/ $\mathcal{E}^+(d, p)$ (the mechanism with the general quadratic thinning formula encompassing all links adjacent to the nodes of the tunnel) is a reasonable traffic protection mechanism to be considered for implementation in FSO networks.

7. Concluding remarks and future work

In the paper we have presented an original optimization model for the quadratic flow thinning (QFT) traffic protection mechanism. The solution algorithm (based on path generation) developed for the model was tested in a numerical study and its time efficiency turned

out to be satisfactory. In the numerical study we have examined a medium-size network in order to compare the cost of various variants of QFT, including its simpler affine version (AFT). Clearly, in comparison with the original FT mechanism, the thinning formula used in QFT introduces an additional constraint on thinning (recall that with FT, (1e) is the only constraint on flow thinning). In effect, QFT requires more link capacity than FT. Still this extra cost can be made acceptable (of the order of few percent) when the proper range (the set of all links, or the set of all links incident to the nodes of the path) of the quadratic formula is adopted. Taking into account all the observations gathered in Table 5, we conclude that the variant $Q/G/\mathcal{E}^+(d, p)$ delivers the best compromise protection solution out of all the considered thinning options.

As far the future research directly related to the presented considerations is concerned, it is important to examine the following issues:

- Add modularity of link capacity (parallel FSO links); this will involve integer variables y^0 in problem FTOP (1) and its QFT counterpart (8).
- Test effectiveness of applying path-sets generated for AFT to QFT (this is done in the paper) instead of using the exact approach with the QFT pricing problems.
- Improve efficiency of the path-generation algorithm for QFT with the $\mathcal{E}^+(d, p)$ range.
- Repeat the numerical study for other network instances to verify the conclusions of this paper.

Acknowledgement: M. Pióro and I. Kalesnikau were supported by the National Science Center (Poland) [Grant 2015/17/B/ST7/03910 “Logical tunnel capacity control – a traffic routing and protection strategy for communication networks with variable link capacity”]. Research of M. Poss was supported by the ANR project ROBUST [ANR-16-CE40-0018].

References

- [1] V. Chan, Free-space optical communications, *Journal of Lightwave Technology* 24 (12) (2006) 4750–4762.
- [2] Y. Li, N. Pappas, V. Angelakis, M. Pióro, D. Yuan, Optimization of free space optical wireless network for cellular backhauling, *IEEE JSAC* 33 (9) (2015) 1841–1854.
- [3] M. Pióro, Y. Fouquet, D. Nace, M. Poss, Optimizing flow thinning protection in multicommodity networks with variable links capacity, *Operations Research* 64 (2) (2016) 273–289.
- [4] M. Pióro, I. Kalesnikau, M. Poss, Path generation for affine flow thinning, in: *Proc. INOC 2017, Lisbon, Portugal, 2017*.
- [5] M. Pióro, I. Kalesnikau, 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), Alghero, Italy, 2017*.

- [6] S. Orlowski, M. Pióro, Complexity of column generation in network design with path-based survivability mechanisms, *Networks* 59 (1) (2012) 132–147.
- [7] M. Pióro, D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*, Morgan-Kaufmann, 2004.
- [8] A. Koster, A. Zymolka, M. Jäger, R. Hülselman, Demand-wise shared protection for meshed optical networks, *Journal of Network and System Management* 13 (1) (2005) 35–55.
- [9] R. Wessäly, S. Orlowski, A. Zymolka, A. Koster, C. Gruber, Demand-wise shared protection revisited: A new model for survivable network design, in: *Proc. 2nd International Network Optimization Conference (INOC 2005)*, 2005, pp. 100–105.
- [10] A. Tomaszewski, M. Pióro, M. Żotkiewicz, On the complexity of resilient network design, *Networks* 55 (2010) 109–118.
- [11] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.
- [12] M. Minoux, *Mathematical Programming: Theory and Algorithms*, John Wiley & Sons, 1986.
- [13] S. Orlowski, R. Wessäly, M. Pióro, A. Tomaszewski, SNDlib 1.0 – survivable network design library, *Networks* 55 (3) (2010) 276–286.

Appendix A. Derivations of pricing problems

Appendix A.1. Simple affine formula

Appendix A.1.1. Case $\tilde{\mathcal{E}}(d, p) = \mathcal{E}^+(d, p)$

In this case the PP formulation is obtained using the following equality which is valid for an arbitrary vector of link-dependent quantities ($A(e)$, $e \in \mathcal{E}$).

$$\sum_{e' \in \mathcal{E}^+(d, q)} A(e) = \sum_{v \in \mathcal{V}} \sum_{e' \in \delta(v)} A(e') z_v - \sum_{e' \in \mathcal{E}} A(e') u_{e'}, \quad s \in \mathcal{S},$$

where $\mathcal{E}(d, q) = \{e \in \mathcal{E} : u_e = 1\}$ and $\mathcal{V}(d, q) = \{v \in \mathcal{V} : z_v = 1\}$. The formulation is more complicated than (19).

$$P = \max \left\{ - \sum_{e \in \mathcal{E}} \pi_e^0 G_e + \left(\sum_{s \in \mathcal{S}} \lambda_d^s g - \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}(s)} \pi_e^s G_e + \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}(s)} \left(\sum_{v \in \mathcal{V}} \sum_{e' \in \delta(v)} \pi_e^s \beta(e', s) R_{e'ev} - \sum_{e' \in \mathcal{E}} \pi_e^s \beta(e', s) T_{e'e} \right) + \sum_{s \in \mathcal{S}} \left(\sum_{v \in \mathcal{V}} \sum_{e' \in \delta(v)} \lambda_d^s \beta(e', s) H_{e'v} - \sum_{e' \in \mathcal{E}} \lambda_d^s \beta(e', s) B_{e'} \right) \right\} \quad (\text{A.1a})$$

$$\sum_{e \in \delta(v)} u_e = 1, \quad v \in \{s(d), t(d)\}; \quad \sum_{e \in \delta(v)} u_e = 2z_v, \quad v \in \mathcal{V} \setminus \{s(d), t(d)\} \quad (\text{A.1b})$$

$$g \leq 1; \quad b_e \leq 1, \quad e \in \mathcal{E}; \quad \sum_{v \in \mathcal{V}} \sum_{e \in \delta(v)} \beta(e, s) H_{ev} - \sum_{e \in \mathcal{E}} \beta(e, s) B_e \leq g, \quad s \in \mathcal{S} \quad (\text{A.1c})$$

$$B_e \leq u_e, B_e \leq b_e, B_e \geq b_e + u_{e'} - 1, \quad e \in \mathcal{E} \quad (\text{A.1d})$$

$$H_{ev} \leq b_e, H_{ev} \leq z_v, H_{ev} \geq b_e + z_v - 1, \quad v \in \mathcal{V}, e \in \delta(v) \quad (\text{A.1e})$$

$$G_e \leq g, G_e \leq u_e, G_e \geq g + u_e - 1, \quad e \in \mathcal{E} \quad (\text{A.1f})$$

$$T_{e'e} \leq u_{e'}, T_{e'e} \leq u_e, T_{e'e} \leq b_{e'}, T_{e'e} \geq b_{e'} + u_{e'} + u_e - 2, \quad e', e \in \mathcal{E} \quad (\text{A.1g})$$

$$R_{e'ev} \leq b_{e'}, R_{e'ev} \leq u_e, H_{e'e} \leq z_v, H_{e'ev} \geq b_{e'} + u_e + z_v - 2, \quad v \in \mathcal{V}, e' \in \delta(v), e \in \mathcal{E}, \quad (\text{A.1h})$$

$$g \in \mathbb{R}_+; b_e \in \mathbb{R}_+, e \in \mathcal{E}; u_e \in \{0, 1\}, e \in \mathcal{E}; z_v \in \{0, 1\}, v \in \mathcal{V} \setminus \{s(d), t(d)\} \quad (\text{A.1i})$$

$$B_e \in \mathbb{R}_+, e \in \mathcal{E}; G_e \in \mathbb{R}_+, e \in \mathcal{E}; T_{ee'} \in \mathbb{R}_+, e, e' \in \mathcal{E}; \quad (\text{A.1j})$$

$$H_{ev} \in \mathbb{R}_+, e \in \mathcal{E}, v \in \mathcal{V}; R_{e'ev}, e', e \in \mathcal{E}, v \in \mathcal{V}. \quad (\text{A.1k})$$

Appendix A.1.2. Case $\tilde{\mathcal{E}}(d, p) = \mathcal{E}$

In the considered case the PP formulation becomes simpler than (19) since now variables $B_{e'}$ are equal to $b_{e'}$ rather than to $b_{e'} \cdot u_{e'}$, and variables $T_{e'e}$ express the product $u_{e'} \cdot b_{e'}$ rather than $u_{e'} \cdot u_e \cdot b_{e'}$. Moreover, the loops in the optimal path will not appear due to the maximal range of the thinning formula.

$$P = \max \left\{ - \sum_{e \in \mathcal{E}} \pi_e^0 G_e + \left(\sum_{s \in \mathcal{S}} \lambda_d^s \right) g - \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}(s)} \pi_e^s G_e + \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}(s)} \sum_{e' \in \mathcal{E}} \pi_e^s \beta(e', s) T_{e'e} - \sum_{s \in \mathcal{S}} \sum_{e' \in \mathcal{E}} \lambda_d^s \beta(e', s) b_{e'} \right\} \quad (\text{A.2a})$$

$$\sum_{e \in \delta(v)} u_e = 1, v \in \{s(d), t(d)\}; \sum_{e \in \delta(v)} u_e = 2z_v, v \in \mathcal{V} \setminus \{s(d), t(d)\} \quad (\text{A.2b})$$

$$g \leq 1; b_e \leq 1, e \in \mathcal{E}; \sum_{e \in \mathcal{E}} \beta(e, s) b_e \leq g, s \in \mathcal{S} \quad (\text{A.2c})$$

$$G_e \leq g, G_e \leq u_e, G_e \geq g + u_e - 1, \quad e \in \mathcal{E} \quad (\text{A.2d})$$

$$T_{e'e} \leq u_e, T_{e'e} \leq b_{e'}, T_{e'e} \geq b_{e'} + u_e - 1, \quad e', e \in \mathcal{E} \quad (\text{A.2e})$$

$$g \in \mathbb{R}_+; b_e \in \mathbb{R}_+, e \in \mathcal{E}; G_e \in \mathbb{R}_+, e \in \mathcal{E}; T_{ee'} \in \mathbb{R}_+, e, e' \in \mathcal{E} \quad (\text{A.2f})$$

$$u_e \in \{0, 1\}, e \in \mathcal{E}; z_v \in \{0, 1\}, v \in \mathcal{V} \setminus \{s(d), t(d)\}. \quad (\text{A.2g})$$

The above formulation is computationally more efficient than its counterpart (19) for $\tilde{\mathcal{E}}(d, p) = \mathcal{E}(d, p)$.

Appendix A.2. General affine formula

Appendix A.2.1. A/G - general PP derivation (for arbitrary range $\tilde{\mathcal{E}}(d, p)$)

Problem D(\mathcal{P}):

$$W(\mathcal{P}) = \max \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} h(d, s) \lambda_d^s \quad (\text{A.3a})$$

$$[y_e^0 \geq 0] \quad \pi_e^0 + \sum_{s \in \mathcal{S}(e)} \alpha(e, s) \pi_e^s \leq \xi(e), \quad e \in \mathcal{E} \quad (\text{A.3b})$$

$$[x_{dp}^0 \geq 0] \quad \sum_{s \in \mathcal{S}} \sigma_{dp}^s \leq \sum_{e \in \mathcal{E}(d, p)} \pi_e^0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (\text{A.3c})$$

$$[x_{dp}^s \geq 0] \quad \lambda_d^s - \varphi_{dp}^s \leq \sigma_{dp}^s + \sum_{e \in \mathcal{E}(d, p) \cap \mathcal{E}(s)} \pi_e^s, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S} \quad (\text{A.3d})$$

$$[z_{dp}^e] \quad \sum_{s \in \mathcal{S}} \beta(e, s) \varphi_{dp}^s = 0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), e \in \tilde{\mathcal{E}}(d, p) \quad (\text{A.3e})$$

$$[z_{dp}^0] \quad \sum_{s \in \mathcal{S}} \varphi_{dp}^s = 0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (\text{A.3f})$$

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

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

As before, we consider a given demand $d \in \mathcal{D}$ and a given path $q \in \widehat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$ and an optimal solution π, λ of the dual (A.3). The minimum of the sum of such violations (taken

over all feasible local dual variables $\sigma = (\sigma^s, s \in \mathcal{S})$ and $\varphi = (\varphi^s, s \in \mathcal{S})$ is expressed through the quantity

$$P(q) = \min_{\sigma \geq 0, \varphi} \left\{ \max \left\{ \sum_{s \in \mathcal{S}} \sigma^s - |q|^0, 0 \right\} + \sum_{s \in \mathcal{S}} \max \left\{ \lambda_d^s - \varphi^s - \sigma^s - |q|^s, 0 \right\} + \sum_{e \in \tilde{\mathcal{E}}(d, q)} \left| \sum_{s \in \mathcal{S}} \beta(e, s) \varphi^s \right| + \left| \sum_{s \in \mathcal{S}} \varphi^s \right| \right\},$$

where $|q|^0 := \sum_{e \in \mathcal{E}(d, q)} \pi_e^0$ (nominal dual length of q), and $|q|^s := \sum_{e \in \mathcal{E}(d, q) \cap \mathcal{E}(s)} \pi_e^s$ (state-dependent dual length of q), $s \in \mathcal{S}$. The LP formulation for computing $P(q)$ is as follows:

$$P(q) = \min \left\{ Z + \sum_{s \in \mathcal{S}} Y_s + \sum_{e \in \tilde{\mathcal{E}}(d, q)} (X'_e + X''_e) + W + W' \right\} \quad (\text{A.5a})$$

$$[g \geq 0] \quad Z \geq \sum_{s \in \mathcal{S}} \sigma^s - |q|^0 \quad (\text{A.5b})$$

$$[a_s \geq 0] \quad Y_s \geq \lambda_d^s - \varphi^s - \sigma^s - |q|^s, \quad s \in \mathcal{S} \quad (\text{A.5c})$$

$$[b'_e \geq 0] \quad X'_e \geq \sum_{s \in \mathcal{S}} \beta(e, s) \varphi^s, \quad e \in \tilde{\mathcal{E}}(d, q) \quad (\text{A.5d})$$

$$[b''_e \geq 0] \quad X''_e \geq -\sum_{s \in \mathcal{S}} \beta(e, s) \varphi^s, \quad e \in \tilde{\mathcal{E}}(d, q) \quad (\text{A.5e})$$

$$[c' \geq 0] \quad W' \geq \sum_{s \in \mathcal{S}} \varphi^s \quad (\text{A.5f})$$

$$[c'' \geq 0] \quad W'' \geq -\sum_{s \in \mathcal{S}} \varphi^s \quad (\text{A.5g})$$

$$\sigma^s \in \mathbb{R}_+, s \in \mathcal{S}; \quad \varphi^s \in \mathbb{R}, s \in \mathcal{S} \quad (\text{A.5h})$$

$$Z, W', W'' \in \mathbb{R}_+; \quad Y_s \in \mathbb{R}_+, s \in \mathcal{S}; \quad X'_e, X''_e \in \mathbb{R}_+, e \in \tilde{\mathcal{E}}(d, q). \quad (\text{A.5i})$$

Next we formulate the problem dual to (A.5), using the dual variables specified in the square brackets in constraints (A.5b)-(A.5g).

$$P(q) = \max \left\{ -g|q|^0 + \sum_{s \in \mathcal{S}} a_s (\lambda^s - |q|^s) \right\} \quad (\text{A.6a})$$

$$0 \leq g \leq 1, \quad 0 \leq c' \leq 1, \quad 0 \leq c'' \leq 1 \quad (\text{A.6b})$$

$$0 \leq a_s \leq g, \quad s \in \mathcal{S} \quad (\text{A.6c})$$

$$0 \leq b'_e \leq 1, \quad 0 \leq b''_e \leq 1, \quad e \in \tilde{\mathcal{E}}(d, q) \quad (\text{A.6d})$$

$$a_s = c' - c'' + \sum_{e \in \tilde{\mathcal{E}}(d, q)} \beta(e, s) (b'_e - b''_e), \quad s \in \mathcal{S}. \quad (\text{A.6e})$$

Below we formulate the pricing problem for the considered general affine formula case only for $\tilde{\mathcal{E}}(d, p) = \mathcal{E}(d, p)$ (and undirected networks).

Appendix A.2.2. Case $\tilde{\mathcal{E}}(d, p) = \mathcal{E}$

Proceeding as in Section 5.1 (and eliminating variables $a_s, s \in \mathcal{S}$, in (A.6)) we obtain the following formulation.

$$P = \max \left\{ -\sum_{e \in \mathcal{E}} \pi_e^0 \overbrace{g u_e}^{G_e} + \sum_{s \in \mathcal{S}} \lambda^s c' - \sum_{s \in \mathcal{S}} \lambda^s c'' + \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}} \beta(e, s) \lambda^s \overbrace{b'_e u_e}^{B'_e} + \right. \\ \left. - \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}} \beta(e, s) \lambda^s \overbrace{b''_e u_e}^{B''_e} - \sum_{s \in \mathcal{S}} \sum_{e' \in \mathcal{E}(s)} \sum_{e \in \mathcal{E}} \beta(e, s) \pi_{e'}^s \overbrace{b'_e u_e u_{e'}}^{T'_{e'e}} \right\} \quad (\text{A.7a})$$

$$\begin{aligned}
& + \sum_{s \in \mathcal{S}} \sum_{e' \in \mathcal{E}(s)} \sum_{e \in \mathcal{E}} \beta(e, s) \pi_{e'}^s \overbrace{b_e'' u_e u_{e'}}^{T_{e'e}''} - \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}(s)} \pi_e^s \overbrace{c' u_e}^{K_e'} + \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}(s)} \pi_e^s \overbrace{c'' u_e}^{K_e''} \} \\
& \sum_{e \in \delta(v)} u_e = 1, v \in \{s(d), t(d)\}; \quad \sum_{e \in \delta(v)} u_e = 2z_v, v \in \mathcal{V} \setminus \{s(d), t(d)\} \tag{A.7b} \\
& 0 \leq g \leq 1, 0 \leq c' \leq 1, 0 \leq c'' \leq 1; 0 \leq b_e \leq 1, 0 \leq b_e'' \leq 1, e \in \mathcal{E} \tag{A.7c} \\
& 0 \leq c' - c'' + \sum_{e \in \mathcal{E}} \beta(e, s) (B_e' - B_e'') \leq g, s \in \mathcal{S} \tag{A.7d} \\
& G_e \leq g, G_e \leq u_e, G_e \geq g + u_e - 1, e \in \mathcal{E} \tag{A.7e} \\
& B_e' \leq u_e, B_e' \leq b_e', B_e' \geq b_e' + u_e - 1, e \in \mathcal{E} \tag{A.7f} \\
& B_e'' \leq u_e, B_e'' \leq b_e'', B_e'' \geq b_e'' + u_e - 1, e \in \mathcal{E} \tag{A.7g} \\
& T_{e'e}' \leq u_{e'}, T_{e'e}' \leq u_e, T_{e'e}' \leq b_e', T_{e'e}' \geq b_e' + u_{e'} + u_e - 2, e', e \in \mathcal{E} \tag{A.7h} \\
& T_{e'e}'' \leq u_{e'}, T_{e'e}'' \leq u_e, T_{e'e}'' \leq b_e'', T_{e'e}'' \geq b_e'' + u_{e'} + u_e - 2, e', e \in \mathcal{E} \\
& K_e' \leq c', K_e' \leq u_e, K_e' \geq c' + u_e - 1, e \in \mathcal{E} \tag{A.7i} \\
& K_e'' \leq c'', K_e'' \leq u_e, K_e'' \geq c'' + u_e - 1, e \in \mathcal{E} \tag{A.7j} \\
& B_e', B_e'', G_e, K_e', K_e'' \in \mathbb{R}_+, e \in \mathcal{E} \tag{A.7k} \\
& T_{e'e}', T_{e'e}'' \in \mathbb{R}_+, e', e \in \mathcal{E} \tag{A.7l} \\
& u_e \in \{0, 1\}, e \in \mathcal{E}; z_v \in \{0, 1\}, v \in \mathcal{V} \setminus \{o(d), t(d)\}. \tag{A.7m}
\end{aligned}$$

Above, variables $G_e', B_e', T_{e'e}', K_e'$ and $G_e'', B_e'', T_{e'e}'', K_e''$ are used to eliminate the products (bilinearities and tri-linearities) of the original variables – this is indicated in (A.7b). We note that this particular case was considered in [4].