



**HAL**  
open science

## **A robust optimization model for affine/quadratic flow thinning -a traffic protection mechanism for networks with variable link capacity**

Ilya Kalesnikau, Michal Pióro, Michael Poss, Dritan Nace, Artur Tomaszewski

### ► **To cite this version:**

Ilya Kalesnikau, Michal Pióro, Michael Poss, Dritan Nace, Artur Tomaszewski. A robust optimization model for affine/quadratic flow thinning -a traffic protection mechanism for networks with variable link capacity. *Networks*, 2020, 75 (4), pp.420-437. 10.1002/net.21929 . hal-02478912

**HAL Id: hal-02478912**

**<https://hal.science/hal-02478912>**

Submitted on 14 Feb 2020

**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.

# A robust optimization model for affine/quadratic flow thinning – a traffic protection mechanism for networks with variable link capacity

Ilya Kalesnikau\*    Michał Pióro†    Michael Poss‡    Dritan Nace§  
Artur Tomaszewski¶

February 14, 2020

## Abstract

Flow thinning (FT) is a traffic protection mechanism for communication networks with variable link capacities, for example wireless networks. With FT, end-to-end traffic demands use dedicated logical tunnels, for example MPLS tunnels, whose nominal capacity is subject to thinning in order to follow fluctuations in link capacities availability. Moreover, instantaneous traffic of each demand is throttled at its originating node accordingly to the current total capacity available on the demands dedicated tunnels so that the network is always capable of carrying the admitted traffic. In this paper we deal with efficient, implementable versions of FT, referred to as AFT (affine FT) and QFT (quadratic FT). By deriving appropriate link availability state and path generation algorithms, we show how real-life network dimensioning problems for AFT/QFT can be efficiently treated using a proper characterization of the network link availability states. Results of a numerical study illustrate tractability of the cost minimization problems, and assess efficiency of AFT/QFT as compared with other protection mechanisms.

**keywords:** robust optimization, uncertainty polytopes, multicommodity flows, traffic protection, variable link capacity, affine and quadratic routing, FSO

## 1 Introduction

*Flow thinning* (FT), a concept introduced in [16], is a traffic protection mechanism designed for communication networks with variable capacity of links. In a typical state of such a network, only a subset of links is fully available while on the remaining links only a fraction of their nominal (i.e., maximum) capacity is usable. Each end-to-end traffic demand is assigned a set of 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 in order to consider variable capacity of the links, fluctuating below the maximum. In effect, the capacity available on the tunnels is also fluctuating below their nominal capacity levels. Thus, since the instantaneous traffic sent between the demand’s end nodes is adapted to the current total capacity available on its dedicated tunnels, the network is always capable of carrying the admitted traffic.

FT is relevant, among other applications, for FSO (free space optics) wireless mesh networks utilizing MPLS tunnels. FSO communications is undoubtedly a promising solution to provide connectivity in areas where deployment of a wired infrastructure is hardly feasible. However, it often faces the problem of time-varying link capacity due to weather disruptions (such as a heavy rainfall) leading to a substantial or even a complete degradation of the capacity available on wireless optical links. Therefore, one of the major issues

---

\*Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland

†Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland. [m.pioro@tele.pw.edu.pl](mailto:m.pioro@tele.pw.edu.pl) *Corresponding author.*

‡UMR CNRS 5506 LIRMM, Université de Montpellier, Montpellier, France

§Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253, Compiègne, France

¶Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland

in the design of FSO architectures is to assure network resilience defined as the ability of the network to guarantee an acceptable level of service in the face of various faults and challenges for normal operation [22].

The current paper presents an optimization model for network dimensioning for two implementable versions of FT: *affine flow thinning* (AFT) [16], and *quadratic flow thinning* (QFT) [20]. With AFT, the capacity of each tunnel is adjusted according to an optimized, tunnel-specific affine flow thinning function, whose arguments are the fractions of currently available link capacities (in relation to the maximum link capacities) on a pre-specified, tunnel-dependent subset of links. With QFT, additional quadratic terms in the flow thinning functions are allowed on top of the affine ones. The main novelty of the presented model is characterization of the link availability states by means of the so called state polytope (an object called the uncertainty polytope in robust optimization [7]) instead of a limited list of preselected states. This feature makes it possible to dimension AFT/QFT networks for representative sets of states that can be met during network operation.

In general, dimensioning of FT (and AFT and QFT) networks is not an easy task because of non-compactness of the related linear programming formulations. Roughly speaking, in such formulations there is a subset of columns (i.e., variables) that corresponds to the flows of the tunnels in different link availability states, and a subset of rows (i.e., constraints) that corresponds to link availability states. Both sets grow exponentially with the size of the network graph; in effect, the formulations must be treated by both column generation (called path generation in our case, see [1, 14]) and row generation (called state generation in our case, see [6]). So far, the issue of non-compactness in question has been solved only in the path generation aspect (see [16] for FT, [18] for AFT, and [20] for QFT). Introducing the state polytope allows for applying, on top of path generation, a systematic algorithm for state generation.

Our previous work, which is extended in this paper, can be summarized as follows:

- The general version of the flow thinning mechanism was introduced in [16], together with a complete presentation of the FT optimization model with a predefined list of (link availability) states, including path generation based on a specific, non-standard pricing problem.
- The concept of affine flow thinning was also introduced in [16], but path generation for the AFT modification of FT was not considered there. This was done, through formulating specific pricing problems for a variety of versions of AFT, in paper [17] and its extended version [18], providing a complete solution algorithm for the basic AFT optimization problem with a predefined list of states.
- In fact, the notion of state polytope was introduced in [16] as well, where a (pretty complicated) optimization linear programming formulation for AFT involving state polytopes for link availability states modeling was derived. However, that formulation, based on dual theory, assumed a predefined list of paths and has never been extended with path generation because of complexity of the related pricing problems.
- Quadratic flow thinning and its various versions were introduced [20] together with an optimization model including path generation.
- The concept of state polytope introduced in [16] was used in an optimization model for the so called global rerouting (a benchmark routing mechanism assuming unrestricted reconfiguration of flows in case of a network failure) presented in paper [8] (and in its extended version [12]). The solution algorithm is iterative and based on Benders' decomposition [5] with dual feasibility tests involving binary variables. Since global rerouting can be modelled by means of link-flows, no path generation was considered there.

The presented paper, together with its conference predecessor [19], introduces an original optimization model for affine and quadratic flow thinning based on path and state generation. Its contribution consists in extending the previous work in the following aspects:

- Although for link availability state description the current model uses state polytopes introduced in [16], and already used for iterative state generation in [12], now the feasibility tests are formulated as linear programs and not as mixed-integer programs as in [12]. This increases efficiency of the current model.

- With respect to the AFT model involving state polytopes described in [16], the current model is enriched with path generation and thanks to that solves the AFT optimization problem taking into account all possible paths.
- The QFT mechanism is added to the optimization model.

With respect to its conference version [19], the current paper contains the following new elements:

- Precise formulation of the feasibility tests for state generation.
- Quadratic flow thinning (on top of AFT considered in [19]).
- An example illustrating the idea of AFT and QFT.
- Enriched discussion on applicability of the proposed approach.
- Numerical study involving a realistic-network example.

The rest of the paper is organized as follows. After introducing basic notation (Section 2), in Section 3 we formulate the basic optimization problem for FT, i.e., the most general version of flow thinning, and its modification for AFT. Then, in Section 4, we describe the notion of state polytope, and then, in Section 5, our optimization procedure for the AFT problem is presented. Next, in Section 6 we generalize the optimization procedure for QFT, and after that, in Section 7, we discuss advantages of AFT/QFT in the practical context. Numerical results illustrating the efficiency of the procedure and cost effectiveness of the optimized networks are presented in Section 8. Finally, after Section 9 that concludes the paper and discusses directions of future work, feasibility tests for state generation are formulated in Appendix A.

## 2 Notation

The (undirected) 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 link represents an undirected pair  $\{v, w\}$  of nodes for some  $v, w \in \mathcal{V}$ ,  $v \neq w$ .

The maximum (called also *nominal*) capacity of link  $e \in \mathcal{E}$  is denoted by  $y_e^0$ , and its unit capacity cost by  $\xi(e)$ ; hence the cost of the network is given by  $C = \sum_{e \in \mathcal{E}} \xi(e)y_e^0$ . The capacity of links is variable, and typically not all links reach their maximum capacity simultaneously. Thus, at any time instant, nominal link capacities  $y_e^0$ ,  $e \in \mathcal{E}$ , are usually available only for a subset of links, and the remaining links have the available capacity reduced. To consider this, we specify a set  $\mathcal{S}$  (called the *state-list*) of (*link availability*) *states*. Each state  $s \in \mathcal{S}$  is characterized by the *link availability ratios*  $\alpha(e, s)$  ( $0 \leq \alpha(e, s) \leq 1$ ),  $e \in \mathcal{E}$ , and, by definition, the capacity of link  $e$  available in state  $s$  is equal to  $y_e^s := \alpha(e, s)y_e^0$ . In the following, we will also use the symmetric notion of *link degradation ratio*  $\beta(e, s) := 1 - \alpha(e, s)$ . 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 (decision) variables that are non-negative and continuous, i.e., non-negative real numbers (in the following the set of real numbers will be denoted by  $\mathbb{R}$ , and the set of non-negative real numbers by  $\mathbb{R}_+$ ).

Traffic demands are represented by set  $\mathcal{D}$ . Each demand  $d \in \mathcal{D}$  is characterized by an undirected pair  $\{o(d), t(d)\}$  composed of the demand's end (origin/destination) nodes, and given traffic volumes  $h(d, s)$ ,  $s \in \mathcal{S}$ , to be realized in the considered set of link availability states. Traffic volumes and link capacities are expressed in the same units. Each demand  $d$  is assigned a set of admissible paths  $\mathcal{P}(d)$  (called the *path-list*) composed of selected elementary paths between  $o(d)$  and  $t(d)$ . Paths in  $\mathcal{P}(d)$ , used to realize the demand (traffic) volumes, are assigned *nominal (path-)flows*  $x_{dp}^0$ ,  $p \in \mathcal{P}(d)$ . The value  $x_{dp}^0$  specifies the (reference) nominal capacity, expressed in the same units as link capacity and demand volume, reserved for demand  $d$  on the tunnel realized on path  $p \in \mathcal{P}(d)$ . The set of all admissible paths is denoted by  $\mathcal{P} := \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$ . The maximum path-lists, i.e., path-lists  $\mathcal{P}(d)$  containing all the elementary paths between  $o(d)$  and  $t(d)$ , will be denoted by  $\hat{\mathcal{P}}(d)$ ,  $d \in \mathcal{D}$ , with  $\hat{\mathcal{P}} := \bigcup_{d \in \mathcal{D}} \hat{\mathcal{P}}(d)$ . Note that since we assume elementary paths, the paths can be identified with the sets of links they traverse – the set of links composing path  $p \in \mathcal{P}(d)$  will be denoted by  $\mathcal{E}(d, p)$ . The sets of admissible paths  $\mathcal{P}(d)$  are parameters in the considered problem formulations, although

we require that all possible elementary paths can be used if this is necessary to achieve the optimum. In the following we will also make use of the sets of paths  $\mathcal{R}(d, e) := \{p \in \mathcal{P}(d) : e \in \mathcal{E}(d, p)\}$ ,  $d \in \mathcal{D}, e \in \mathcal{E}$ . Certainly,  $\mathcal{R}(d, e) \subseteq \mathcal{P}(d)$  and  $\mathcal{R}(d, e)$  is the set of all those paths in  $\mathcal{P}(d)$  that traverse link  $e$ .

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 state  $s \in \mathcal{S}$ , as the available link capacity  $y^s := (y_e^s, e \in \mathcal{E})$  is decreased with respect to the nominal link capacity  $y^0$ . To account for that, the nominal flows must be thinned (decreased) to fit the reduced link capacity. Still, the thinned flows must be sufficient to carry the assumed demand volumes  $h(d, s)$ ,  $d \in \mathcal{D}$ . The thinned nominal path-flows for state  $s \in \mathcal{S}$  are denoted by  $x_{dp}^s$ ,  $d \in \mathcal{D}, p \in \mathcal{P}(d)$ . These flows are reserved on the admissible paths for the duration of the state. As only thinning of nominal flows is allowed, the inequality  $x_{dp}^s \leq x_{dp}^0$  must hold for each path and each state. Clearly, the link loads induced by the nominal flows cannot exceed the nominal link capacity. Moreover, in each state the link loads induced by the flows thinned for that state cannot exceed the currently available link capacity. The path-flows, described by vectors  $x^0 := (x_{dp}^0, d \in \mathcal{D}, p \in \mathcal{P}(d))$  and  $x^s := (x_{dp}^s, d \in \mathcal{D}, p \in \mathcal{P}(d))$ ,  $s \in \mathcal{S}$ , are non-negative continuous optimization variables.

Observe that (following [9]) in our notation (with some obvious exceptions) the objects that are given and fixed (for example problem parameters) are indexed using brackets, like in  $h(d, s), \mathcal{P}(d)$ , etc., while variables are indexed using subscripts and superscripts, like  $y_e^0, x_{dp}^s$ , etc.

To end this section we note that considerations of this paper are applicable to networks with directed links and/or demands as well. Since we are using path-flows rather than link-flows, the direction of links is taken into account when predefining (or generating) paths. The (meaningful) cases are modeled as follows.

1. Undirected links and undirected demands: as discussed above.
2. Directed links (i.e., arcs) and directed demands: paths are directed and each path  $p \in \mathcal{P}(d)$  starts at its originating node  $o(d)$ , then traverses a number of arcs according to their directions, and ends at its terminating node  $t(d)$ .
3. Bi-directed (full-duplex) links and directed demands: as above assuming each bi-directed link is substituted by two oppositely directed arcs with the same capacity.
4. Undirected links and directed demands: undirected paths are used for directed demands.

### 3 Optimization problem for a given list of states

Let path-list  $\mathcal{P} = \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$  and state-list  $\mathcal{S}$  be given. Below we formulate two versions of the basic optimization problem considered in this paper, one for the general version of flow thinning (FT), and one for its affine version (AFT).

#### 3.1 Problem formulation for FT

The basic optimization problem for FT (referred to as FTOP) is represented by the following linear programming (LP) formulation involving non-negative real-valued variables  $y^0, x^0, x^s$  ( $s \in \mathcal{S}$ ).

$$\begin{aligned}
\text{Problem FTOP}(\mathcal{P}, \mathcal{S}): \quad & C(\mathcal{P}, \mathcal{S}) = \min \sum_{e \in \mathcal{E}} \xi(e) y_e^0 & (1a) \\
& \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d, e)} x_{dp}^0 \leq y_e^0, \quad e \in \mathcal{E} & (1b) \\
& \sum_{p \in \mathcal{P}(d)} x_{dp}^s \geq h(d, s), \quad d \in \mathcal{D}, s \in \mathcal{S} & (1c) \\
& \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d, e)} x_{dp}^s \leq \alpha(e, s) y_e^0, \quad e \in \mathcal{E}, s \in \mathcal{S}(e) & (1d) \\
& x_{dp}^s \leq x_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S} & (1e) \\
& y_e^0 \in \mathbb{R}_+, e \in \mathcal{E}; \quad x_{dp}^0 \in \mathbb{R}_+, d \in \mathcal{D}, p \in \mathcal{P}(d); \quad x_{dp}^s \in \mathbb{R}_+, d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S}. & (1f)
\end{aligned}$$

Objective (1a) minimizes the cost of links. The first capacity constraint (1b) does not allow the nominal link loads (i.e., the loads induced by the nominal flows) to exceed the nominal link capacities. Next, the demand

constraint (1c) ensures 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) does not allow the capacity of each link  $e \in \mathcal{E}$  available in each state  $s \in \mathcal{S}(e)$  to be 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 degradations 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).

Note that formulation (1) is non-compact as the number of flow variables  $x^0, x^s$  ( $s \in \mathcal{S}$ ) and the number of constraints (1e) grows exponentially with the size of the network, even if the number of states in  $\mathcal{S}$  is polynomial. In fact, FTOP is  $\mathcal{NP}$ -hard. A path generation algorithm for FTOP based on an appropriate pricing problem, was investigated in depth in [16].

We mention here that FT can be considered as a generalization of the *Demand-Wise Shared Protection* (DWSP) strategy [10, 25] 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 in the SRLG case (where link availability ratios are binary) is automatic: 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 [24]), the problem is  $\mathcal{NP}$ -hard also for FT, and hence, most likely, for the AFT variant of FT considered in the remaining part of this paper.

### 3.2 Problem formulation for AFT

The general version of FT considered in the previous section is hardly implementable in the network, mainly because of potential difficulties in on-line state recognition at the tunnel origins and undefined flow thinning values in the states not included in  $\mathcal{S}$ . These disadvantages of FT are eliminated when an affine version of FT, i.e., AFT, is applied. A disadvantage of AFT as compared with FT are increased link capacities (and hence the network cost  $C$ ) required to protect traffic in the states with decreased link availability. We will discuss these issues in more detail in Section 7.

The basic optimization problem for AFT (referred to as AFTOP) formulated below as an LP problem involves real-valued variables  $y^0, x^0, x^s$  ( $s \in \mathcal{S}$ ) (as FTOP) and, additionally, non-negative real-valued variables  $z$ , where  $z := (z_{dp}^e, d \in \mathcal{D}, p \in \mathcal{P}(d), e \in \mathcal{E})$  that specify the coefficients (called affine coefficients in the following) of the affine flow-thinning formulae.

$$\textbf{Problem AFTOP}(\mathcal{P}, \mathcal{S}): \quad C(\mathcal{P}, \mathcal{S}) = \min \sum_{e \in \mathcal{E}} \xi(e) y_e^0 \quad (2a)$$

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d, e)} x_{dp}^0 \leq y_e^0, \quad e \in \mathcal{E} \quad (2b)$$

$$\sum_{p \in \mathcal{P}(d)} x_{dp}^s \geq h(d, s), \quad d \in \mathcal{D}, \quad s \in \mathcal{S} \quad (2c)$$

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d, e)} x_{dp}^s \leq \alpha(e, s) y_e^0, \quad e \in \mathcal{E}, \quad s \in \mathcal{S}(e) \quad (2d)$$

$$x_{dp}^s = x_{dp}^0 - \sum_{e \in \mathcal{E}(d, p)} \beta(e, s) z_{dp}^e, \quad d \in \mathcal{D}, \quad p \in \mathcal{P}(d), \quad s \in \mathcal{S} \quad (2e)$$

$$y_e^0 \in \mathbb{R}_+, \quad e \in \mathcal{E}; \quad x_{dp}^0 \in \mathbb{R}_+, \quad d \in \mathcal{D}, \quad p \in \mathcal{P}(d); \quad x_{dp}^s \in \mathbb{R}_+, \quad d \in \mathcal{D}, \quad p \in \mathcal{P}(d), \quad s \in \mathcal{S} \quad (2f)$$

$$z_{dp}^e \in \mathbb{R}_+, \quad d \in \mathcal{D}, \quad p \in \mathcal{P}(d), \quad e \in \mathcal{E}. \quad (2g)$$

Interpretation of (2a)-(2d) is the same as interpretation of their counterparts in formulation (1), while equalities (2e) define the state-dependent flows through the affine formula. Note that the form of equations (2e) ensures, due to nonnegativity of the variables  $z$ , that the nominal flows  $x^0$  are actually thinned (or not changed) in any state in  $\mathcal{S}$ . Note that formulation (2) requires path generation as well – this issue will be discussed in Section 5.2.

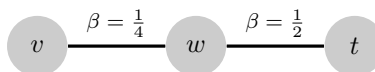


Figure 1: A simple network.

**Example 1:** In order to illustrate the idea of affine flow thinning, consider the simple network shown in Figure 1 with two links (link  $e = 1$  between nodes  $v$  and  $w$ , and link  $e = 2$  between nodes  $w$  and  $t$ , both of nominal capacity equal to 1) and only one demand (demand  $d = 1$  between nodes  $v$  and  $t$  with only one path  $p = 1$  in  $\mathcal{P}(d)$  composed of both links). We assume two states in the state-list  $\mathcal{S}$  ( $\mathcal{S} = \{1, 2\}$ ), each consisting of degradation of a single link: in state  $s = 1$ , link  $e = 1$  is degraded with ratio  $\beta(1, 1) = \frac{1}{2}$  (and  $\beta(2, 1) = 0$ ), while in state  $s = 2$ , link  $e = 2$  is degraded with ratio  $\beta(2, 2) = \frac{1}{4}$  (and  $\beta(1, 2) = 0$ ). Now let us fix the affine coefficients by putting  $z^e = 1$ ,  $e = 1, 2$ , and specify the affine thinning formula resulting from (2e):

$$x^s = x^0 - \beta(1, s) - \beta(2, s), \quad s \in \mathcal{S}. \quad (3)$$

Note that in (3) the subscripts  $d, p$  in  $x^0$  and  $x^s$  are skipped as there is only one demand ( $d = 1$ ) with only one path ( $p = 1$ ). Assuming the maximum admissible nominal flow  $x^0$  which is equal to 1 (as the nominal capacity of both links is 1) we get

$$x^1 = 1 - \beta(1, 1) - \beta(2, 1) = 1 - \frac{1}{2} - 0 = \frac{1}{2} \quad \text{and} \quad x^2 = 1 - \beta(1, 2) - \beta(2, 2) = 1 - 0 - \frac{1}{4} = \frac{3}{4}.$$

For both states these are maximum admissible path-flows and for  $s = 1, 2$  they realize the demand volumes  $h(1) = \frac{1}{2}$  and  $h(2) = \frac{3}{4}$ , respectively (clearly, these flows are achievable with FT as well). Hence, in this particular case, AFT is as efficient as FT.

To end this example, observe that above the nominal flow  $x^0$  serves merely as a reference flow that is thinned in each of the two states. However, if we add the so called *full-availability state*  $s = 0$  with both links fully available ( $\beta(1, 0) = \beta(2, 0) = 0$ ), then the flow for this state calculated according to the above defined AFT formula is feasible (and equal to the nominal flow  $x^0$ ) so that in state  $s = 0$  the demand volume  $h(0) = 1$  is realized. It is important to note here that in general the full-availability state does not have to be considered, and even if it is, the nominal flows do not have to be equal to the their counterparts calculated for this state (but clearly, by definition, cannot be smaller).  $\square$

In essence, formulae (2e) amount to restricting allowable flow decisions  $x_{dp}^s$  by enforcing them to follow an affine decision rule [3], and in this way overcome the aforementioned difficulties of FT. This is because the flow decisions are now entirely determined by fixed (optimized) values of decision variables  $x_{dp}^0$  and  $z_{dp}^e$  and depend only on the values of the degradation ratios  $\beta(e, s)$  in a given state  $s$ . In fact, AFT optimization can be viewed as a two-stage approach. First, we solve formulation (2) to determine optimal values for  $x_{dp}^0$  and  $z_{dp}^e$ . Next, for each path  $p \in \mathcal{P}(d)$ , we use these values to adjust the actual path-flow in real time, depending on the current status of the path, i.e., on the current (state-dependent) values of the degradation ratios of the links along the given path. Importantly, the current values of the degradation ratios along a given path can be efficiently signalled back to its originating node (i.e., with low delay) so that the path-flows are adjusted on-line. Moreover, the affine functions are applicable, through formula (2e), to the link availability states that were not incorporated in the AFTOP formulation.

## 4 State polytope

So far the link availability states have been provided through a predefined explicit list  $\mathcal{S}$ , which means that the number of some constraints and decision variables in both FTOP and AFTOP are proportional to the number of states. Therefore, the list cannot be too large, in particular its size should not grow exponentially with the number of links; otherwise the resulting formulations would contain an excessive number of constraints and variables. This is a well-known drawback of optimization problems under uncertainty where many scenarios are used to represent the possible values taken by the uncertain parameters. In stochastic programming, this difficulty is alleviated by sampling relevant subset of scenarios, obtaining solutions that are good enough with high probability. In this work we have no probability associated with the states, so we rather follow the robust optimization paradigm [4] and replace our explicit list of states by their convex hull, which we call the state polytope in what follows. Considering the convex hull does not further restrict the problem, as one readily verifies that a solution is feasible for a set of states if, and only if, it is feasible for all states in their convex hull (for a formal proof see, e.g., [2]). It turns out that the state polytope enables considering potentially exponentially many different states in a compact manner.

The state polytope is constructed as follows. Consider a set of indices  $\mathcal{K} := \{1, 2, \dots, K\}$  (where  $K \geq 1$ ) and a given sequence of integers  $N := (N(1), N(2), \dots, N(K))$  such that

$$1 \leq N(k) \leq |\mathcal{E}|, k \in \mathcal{K}, \quad \text{and} \quad \sum_{k \in \mathcal{K}} N(k) \geq |\mathcal{E}|. \quad (4)$$

As explained below, each index  $k$  in  $\mathcal{K}$  will define a particular *type of a link* that determines an availability ratio  $a(k)$  and a coefficient  $b(k)$  used for reducing demand traffic volumes.

Now let  $\mathcal{B}(N)$  be the set of all binary vectors  $u := (u_e^k, e \in \mathcal{E}, k \in \mathcal{K})$  fulfilling the conditions:

$$\sum_{k \in \mathcal{K}} u_e^k = 1, e \in \mathcal{E} \quad (5a)$$

$$\sum_{e \in \mathcal{E}} u_e^k \leq N(k), k \in \mathcal{K}. \quad (5b)$$

The so defined set  $\mathcal{B}(N)$  will be called the *set of state patterns*, and its elements *state patterns*. Due to condition (5a), each state pattern  $u \in \mathcal{B}(N)$  determines the unique link types: link  $e$  is of type  $k$  if, and only if,  $u_e^k = 1$ . Note that condition (5a) implies also that  $\sum_{k \in \mathcal{K}} U^k = |\mathcal{E}|$ , where  $U^k := \sum_{e \in \mathcal{E}} u_e^k$  is the number of links of type  $k$  in  $u$ , and that the upper bounds on  $U^k$  imposed by (5b) are valid due to the second condition in (4).

Each type  $k$  in  $\mathcal{K}$  is associated with two numbers,  $a(k)$  and  $b(k)$ , specified by the two following (given) sequences:

$$1 = a(1) > a(2) > \dots > a(K-1) > a(K) \geq 0 \quad (6a)$$

$$0 = b(1) \leq b(2) \leq \dots \leq b(K) < 1. \quad (6b)$$

The quantities introduced in (6a) represent possible values for link availability ratios with  $a(1)$  representing full link availability, and  $a(K)$  the total loss of link's capacity. Coefficients  $b(k)$  introduced in (6b), in turn, are called *reduction coefficients*, and used to define demand reduction ratios. Certainly, in a given state pattern, links of type  $k$  are characterized by the values of  $a(k)$  and  $b(k)$ .

For a given state pattern  $u$  in  $\mathcal{B}(N)$  its *demand (volume) reduction ratio* is defined as follows:

$$B(u) := 1 - \sum_{k \in \mathcal{K}} b(k)U^k. \quad (7)$$

Demand reduction ratios are used to make the demand volumes state-dependent. For that we will assume a given vector  $H := (H(d) \geq 0, d \in \mathcal{D})$  of *reference demand volumes* and treat each value  $H(d)B(u)$  as the (reduced) volume of demand  $d \in \mathcal{D}$  in the state corresponding to pattern  $u \in \mathcal{B}(N)$ . The rationale behind this definition is that for each  $k \in \mathcal{K}$ , the reduction is proportional to coefficients  $b(k)$  ( $b(k)$  are non-decreasing with the link degradation ratio increase, since, due to (6), link degradation ratios  $1 - a(k)$  increase with  $k$ , and  $b(k)$  do not decrease with  $k$ ), and to the number of links  $U^k$  of type  $k$  in pattern  $u$ . Note that in order to fulfil the requirement  $0 \leq B(u) \leq 1$ , the quantities  $b(k)$  must fulfil the technical condition:

$$\sum_{k=k(0)}^K b(k)N(k) + b(k(0) - 1)(|\mathcal{E}| - \sum_{k=k(0)}^K N(k)) \leq 1 \quad (8)$$

where  $k(0)$  is the smallest index for which

$$\sum_{k=k(0)}^K N(k) = |\mathcal{E}| \quad \text{or} \quad (9a)$$

$$\sum_{k=k(0)}^K N(k) < |\mathcal{E}| \quad \text{and} \quad \sum_{k=k(0)-1}^K N(k) > |\mathcal{E}|. \quad (9b)$$

Next, consider the following vectors  $\alpha(u) := (\alpha(e, u), e \in \mathcal{E})$  and  $h(u) := (h(d, u), d \in \mathcal{D})$  for  $u \in \mathcal{B}(N)$ , where:

$$\alpha(e, u) := \sum_{k \in \mathcal{K}} a(k)u_e^k, \quad e \in \mathcal{E} \quad (10a)$$

$$h(d, u) := H(d)B(u), \quad d \in \mathcal{D} \quad (10b)$$

where  $B(u)$  are defined by (7).

For a given setting of parameters

- $N$  – the vector of upper bounds on the number of links of a given type fulfilling (4)



- $a$  – the vector of link availability ratios fulfilling (6a)
- $b$  – the vector of traffic reduction coefficients fulfilling (6b) and (8)-(9)
- $H$  – vector of reference demand volumes

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  $\mathcal{B}(N)$ , where each state  $s(u)$  is characterized by link availability coefficients  $\alpha(u)$  and demand volumes  $h(u)$  specified in (10a) and (10b), respectively. Note again that due to (8)-(9), for each  $d \in \mathcal{D}$  it holds that  $0 \leq h(d, u) \leq H(d)$  for all  $u \in \mathcal{B}(N)$ . In fact, the demand volumes in vector  $H$  are thinned uniformly, according to  $B(u)$  – a measure of the ratio of the amount of unavailable capacity.

As discussed in the Electronic Companion of [16], the so defined set of states  $\widehat{\mathcal{B}}(N, a, b, H) := \{s(u) : u \in \mathcal{B}(N)\}$ , can be used to model a variety of particular state-lists, for example all combinations of simultaneous degradations of at most  $N(1)$  links degraded to availability ratio  $a(1)$ , at most  $N(2)$  links degraded to availability ratio  $a(2)$ , and at most  $N(3)$  links degraded to availability ratio  $a(3)$ .

Now let us consider the *state polytope*  $\mathcal{Q}(N)$  in the  $|\mathcal{E}||\mathcal{K}|$ -dimensional space defined by conditions (5) but this time imposed on real-valued vectors  $u$  with non-negative components, i.e., on  $u \in \mathbb{R}_+^{|\mathcal{E}||\mathcal{K}|}$ , rather than on binary vectors  $u \in \{0, 1\}^{|\mathcal{E}||\mathcal{K}|}$ . Observe that the vertices of polytope  $\mathcal{Q}(N)$  are binary since the coefficient matrix specifying constraints (5) is identical to the one of the assignment problem, and the latter matrix is known to be totally unimodular [13]. Thus,  $\mathcal{Q}(N) = \text{conv}(\mathcal{B}(N))$ , that is  $\mathcal{Q}(N)$  is the convex hull of  $\mathcal{B}(N)$ . Therefore the following property, showing that  $\mathcal{Q}(N)$  becomes crucial when solving of *AFTOP* for a state-list specified by  $\widehat{\mathcal{B}}(N, a, b, H)$ , holds:

**Property 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.

The notion of the state polytope was introduced in [16] under the name *uncertainty polytope*, as it is a variant of the budgeted uncertainty polytope described in [7], a well-known object in robust network design and other linear problems assuming uncertainty.

## 5 Solving AFTOP for a state polytope

Suppose we wish to solve AFTOP for the state-list defined by  $\widehat{\mathcal{B}}(N, a, b, H)$ , i.e., for all states of the form  $s(u)$ ,  $u \in \mathcal{B}(N)$ . Below we will show how this can be achieved through an iterative algorithm based on the state and path generation algorithms.

### 5.1 Feasibility tests and state generation algorithm

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) for  $\mathcal{S}$  (note that flows  $x^s(\mathcal{S})$  are determined by  $x^0(\mathcal{S})$  and  $z(\mathcal{S})$  through equations (2e)). Clearly, in order to see whether this solution is feasible for all states in  $\widehat{\mathcal{B}}(N, a, b, H)$  we need to check if the constraints

$$h(d, u) - \sum_{p \in \mathcal{P}(d)} (x_{dp}^0(\mathcal{S}) - \sum_{e \in \mathcal{E}(d,p)} \beta(e, u) z_{dp}^e(\mathcal{S})) \leq 0, \quad d \in \mathcal{D} \quad (11a)$$

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d,e)} (x_{dp}^0(\mathcal{S}) - \sum_{e \in \mathcal{E}(d,p)} \beta(e, u) z_{dp}^e(\mathcal{S})) - \alpha(e, u) y_e^0(\mathcal{S}) \leq 0, \quad e \in \mathcal{E} \quad (11b)$$

$$-(x_{dp}^0(\mathcal{S}) - \sum_{e \in \mathcal{E}(d,p)} \beta(e, u) z_{dp}^e(\mathcal{S})) \leq 0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (11c)$$

(corresponding to appropriate constraints in (2)) are satisfied for each  $u \in \mathcal{B}(N)$ .

This can be done for each of the constraints (11a)-(11c) separately, by formulating a corresponding binary program of maximizing the left hand of the appropriate inequality in (11) over  $\mathcal{B}(N)$ , where  $\beta(e, u) = 1 - \alpha(e, u)$ , and  $\alpha(e, u)$  and  $h(d, u)$  expressed with  $u$  as follows (cf. (7) and (10)):

$$\alpha(e, u) := \sum_{k \in \mathcal{K}} a(k) u_e^k, \quad e \in \mathcal{E} \quad (12a)$$

$$h(d, u) := H(d)(1 - \sum_{k \in \mathcal{K}} b(k) \sum_{e \in \mathcal{E}} u_e^k), \quad d \in \mathcal{D} \quad (12b)$$

The so described *feasibility tests* can be written in a concise form as follows:

$$T(d) = \max_{u \in \mathcal{B}(N)} \{h(d, u) - \sum_{p \in \mathcal{P}(d)} x_{dp}(u)\}, \quad d \in \mathcal{D} \quad (13a)$$

$$T(e) = \max_{u \in \mathcal{B}(N)} \{ \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d, e)} x_{dp}(u) - \alpha(e, u) y_e^0(\mathcal{S}) \}, \quad e \in \mathcal{E} \quad (13b)$$

$$T(d, p) = \max_{u \in \mathcal{B}(N)} \{ -x_{dp}(u) \}, \quad d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (13c)$$

where  $x_{dp}(u) := x_{dp}^0(\mathcal{S}) - \sum_{e \in \mathcal{E}(d, p)} \beta(e, u) z_{dp}^e(\mathcal{S})$  for  $d \in \mathcal{D}, p \in \mathcal{P}(d), u \in \mathcal{B}(N)$ .

Let  $u^*$  denote an optimal solution for a considered test. Clearly, if for some  $d \in \mathcal{D}$ , the value  $T(d)$  obtained from the first test is positive, then  $\sum_{p \in \mathcal{P}(d)} x_{dp}(u^*) < h(d, u^*)$ , that is demand  $d$  is not satisfied by the considered feasible solution  $y^0(\mathcal{S}), x^0(\mathcal{S}), z(\mathcal{S})$  of AFTOP( $\mathcal{P}, \mathcal{S}$ ). The same holds when  $T(e) > 0$  (then  $\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d, e)} x_{dp}(u^*) > \alpha(e, u^*) y_e^0(\mathcal{S})$ ) or  $T(d, p) > 0$  (then  $x_{dp}(u^*) < 0$ ).

Thus, if the condition

$$\max_{d \in \mathcal{D}} T(d) > 0 \text{ or } \max_{e \in \mathcal{E}} T(e) > 0 \text{ or } \max_{d \in \mathcal{D}, p \in \mathcal{P}(d)} T(d, p) > 0 \quad (14)$$

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*).

It is of outmost importance that, by Property 1 stated by the end of Section 4, the feasibility tests (13) can be made linear programs since the set of feasible solutions, i.e.,  $\mathcal{B}(N)$  can simply be substituted with  $\mathcal{Q}(N)$ . The appropriate LP formulations are given in Appendix A.

The algorithm for solving AFTOP (for fixed path-list  $\mathcal{P}$ ) by state generation based on the introduced feasibility tests is given below. The algorithm requires an initial state-list, for example a reasonable subset of  $\widehat{\mathcal{B}}(N, a, b, H)$ . However, if we wish to consider some states outside  $\widehat{\mathcal{B}}(N, a, b, H)$ , then we simply put them on the initial list.

---

### State generation algorithm – SGA( $\mathcal{P}$ )

**Step 1:** Define an initial state-list  $\mathcal{S}$ .

**Step 2:** Solve AFTOP( $\mathcal{P}, \mathcal{S}$ ); let  $y^0(\mathcal{S}), x^0(\mathcal{S}), z(\mathcal{S})$  be its optimal solution.

**Step 3:** Solve tests (13); put  $u(d) := \operatorname{argmax}_{d \in \mathcal{D}} T(d)$ ;  $u(e) := \operatorname{argmax}_{e \in \mathcal{E}} T(e)$ ;  $u(d, p) := \operatorname{argmax}_{d \in \mathcal{D}, p \in \mathcal{P}(d)} T(d, p)$ .

**Step 4:** For all  $d \in \mathcal{D}, e \in \mathcal{E}, p \in \mathcal{P}(d)$ , define:

$$\begin{aligned} \mathcal{S}'(d) &:= \{s(u(d))\} \text{ if } T(d) > 0 \text{ (and } \mathcal{S}'(d) = \emptyset, \text{ otherwise)} \\ \mathcal{S}'(e) &:= \{s(u(e))\} \text{ if } T(e) > 0 \text{ (and } \mathcal{S}'(e) = \emptyset, \text{ otherwise)} \\ \mathcal{S}'(d, p) &:= \{s(u(d, p))\} \text{ if } T(d, p) > 0 \text{ (and } \mathcal{S}'(d, p) = \emptyset, \text{ otherwise)} \\ \mathcal{S}' &:= \bigcup_{d \in \mathcal{D}} \mathcal{S}'(d) \cup \bigcup_{e \in \mathcal{E}} \mathcal{S}'(e) \cup \bigcup_{d \in \mathcal{D}, p \in \mathcal{P}(d)} \mathcal{S}'(d, p). \end{aligned}$$

If  $\mathcal{S}' = \emptyset$  then stop. Otherwise, put  $\mathcal{S} := \mathcal{S} \cup \mathcal{S}'$  and go to Step 2.

---

Note that, by definition, all elements of  $\mathcal{S}'$  are different and so are all states in  $\mathcal{S}$ . When the algorithm stops, the current vectors  $y^0(\mathcal{S}), x^0(\mathcal{S}), z(\mathcal{S})$  computed in Step 2 form an optimal solution of AFTOP( $\mathcal{P}, \mathcal{S}$ ), where  $\mathcal{S}$  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).

## 5.2 Pricing problems and path generation algorithm

Let the state-list  $\mathcal{S}$  be fixed. Recall that AFTOP( $\mathcal{P}, \mathcal{S}$ ) assumes that the path-lists  $\mathcal{P}(d), d \in \mathcal{D}$ , are predefined, and contain only a subset of the set  $\widehat{\mathcal{P}}(d)$  of all paths that are available for each demand  $d$  in

the network graph  $\mathcal{G}$  (the number of paths in  $\hat{\mathcal{P}}(d)$  grows exponentially with the size of  $\mathcal{G}$ ). Thus, finding a true optimum of AFTOP (for a given list of states  $\mathcal{S}$ ) requires path generation. A path generation (PG) algorithm to solve the full version of AFTOP that takes all paths in  $\hat{\mathcal{P}}$  into consideration was presented and discussed in our previous paper [18]. It was shown there how to derive a pricing problem based on the problem dual to (2) for a fixed state-list  $\mathcal{S}$ :

$$\text{Problem DAFTOP}(\mathcal{P}): \quad W(\mathcal{P}) = \max \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} h(d, s) \lambda_d^s \quad (15a)$$

$$\pi_e^0 + \sum_{s \in \mathcal{S}(e)} \alpha(e, s) \pi_e^s \leq \xi(e), \quad e \in \mathcal{E} \quad (15b)$$

$$\sum_{s \in \mathcal{S}} \varphi_{dp}^s \leq \sum_{e \in \mathcal{E}(d, p)} \pi_e^0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (15c)$$

$$\lambda_d^s \leq \varphi_{dp}^s + \sum_{e \in \mathcal{E}(d, p) \cap \mathcal{E}(s)} \pi_e^s, \quad s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (15d)$$

$$\sum_{s \in \mathcal{S}} \beta(e, s) \varphi_{dp}^s \geq 0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), e \in \mathcal{E}(d, p) \quad (15e)$$

$$\pi_e^0 \in \mathbb{R}_+, e \in \mathcal{E}; \quad \pi_e^s \in \mathbb{R}_+, e \in \mathcal{E}, s \in \mathcal{S}; \quad \lambda_d^s \in \mathbb{R}_+, d \in \mathcal{D}, s \in \mathcal{S} \quad (15f)$$

$$\varphi_{dp}^s \in \mathbb{R}, d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S}. \quad (15g)$$

Note that the dual variables (specified in (15f) and (15g)) correspond to the following primal constraints in AFTOP:  $\pi_e^0$  to (2b),  $\pi_e^s$  to (2d),  $\lambda_d^s$  to (2c), and  $\varphi_{dp}^s$  to (2e).

Let  $\lambda := (\lambda_d^s, d \in \mathcal{D}, s \in \mathcal{S})$ ,  $\pi := (\pi_e^0, e \in \mathcal{E}; \pi_e^s, e \in \mathcal{E}, s \in \mathcal{S})$ ,  $\varphi := (\varphi_{dp}^s, d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S})$ , and let  $D(\mathcal{P})$  denote the dual polyhedron specified by (15b)-(15g) projected onto the  $(\lambda, \pi)$  space.

Now let  $\lambda^*, \pi^*$  be an optimal solution of DAFTOP( $\mathcal{P}$ ). The *pricing problem* for AFTOP( $\mathcal{P}, \mathcal{S}$ ), in the following referred to as PP( $\mathcal{P}, \lambda^*, \pi^*, d$ ), consists in finding, for a given demand  $d \in \mathcal{D}$ , a path  $q(d) \in \hat{\mathcal{P}}(d)$  (if any) such that:

- when constraints (15c)-(15e) for the new path  $q(d)$  are added to the dual problem, then the dual solution  $(\lambda^*, \pi^*)$  is separated from the dual polyhedron  $D(\mathcal{P} \cup \{q(d)\})$
- $q(d)$  maximizes, over all paths in  $\hat{\mathcal{P}}(d)$ , the sum of violations (denoted by  $W(d)$ ) of the new constraints by current  $\lambda^*, \pi^*$  ( $f(d) = 0$  means that no constraints are violated).

A somewhat complicated derivation of the so described pricing problem and the resulting (binary) formulation of PP( $\mathcal{P}, \lambda^*, \pi^*, d$ ) are omitted here as they were already presented in detail in Section 4 of [18] (see also formulation (17) in [20]).

The algorithm for solving AFTOP (for fixed  $\mathcal{S}$ ) by path generation is given below. The algorithm requires an initial path-list, for example a minimal list  $\mathcal{P}$  of simple paths for which AFTOP( $\mathcal{P}, \mathcal{S}$ ) is feasible.

---

### Path generation algorithm – PGA( $\mathcal{S}$ )

**Step 1:** Define initial path-lists  $\mathcal{P}(d), d \in \mathcal{D}$ , and put  $\mathcal{P} := \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$ .

**Step 2:** Solve the dual problem DAFTOP( $\mathcal{P}$ ) given by (15) to obtain optimal dual variables  $\lambda^*, \pi^*$ .

**Step 3:** Solve the pricing problem PP( $\mathcal{P}, \lambda^*, \pi^*, d$ ) for each  $d \in \mathcal{D}$ ; if  $W(d) > 0$  then add the resulting path  $q(d)$  to the path-list  $\mathcal{P}(d)$ . If for no demand a path has been added, then stop: the resulting path-lists are sufficient to solve AFTOP( $\hat{\mathcal{P}}, \mathcal{S}$ ) to optimality. Otherwise, go to Step 2.

---

## 5.3 Full iterative procedure for solving AFTOP

In order to find the true minimum of AFTOP (when all paths are considered) for a given state polytope  $\hat{\mathcal{B}}(N, a, b, H)$ , we need to 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.

---

### SGA+PGA algorithm

**Step 1:** Define initial path-lists  $\mathcal{P}(d), d \in \mathcal{D}$ ,  $\mathcal{P} := \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$ , and an initial state-list  $\mathcal{S}$ .

**Step 2:** Apply PGA( $\mathcal{S}$ ); in consequence, path-lists  $\mathcal{P}(d), d \in \mathcal{D}$ , are in general extended.

**Step 3:** Apply SGA( $\mathcal{P}$ ); in consequence, state-list  $\mathcal{S}$  is in general extended. If the state-list has not been extended, then stop.

**Step 4:** Apply PGA( $\mathcal{S}$ ); If no path-list has been extended, then stop; otherwise, go to Step 3.

---

The above algorithm will always stop 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. Effectiveness of the SGA+PGA algorithm, which is the most important result of the current paper, will be illustrated in Section 8.

## 6 Extension of affine FT to quadratic FT

The simple affine formula (2e) suffers from conservatism since it restricts the flow decisions to very simple functions of the link degradation ratios, namely, affine functions. This is done mainly for tractability reasons as the affine formulae are easy to embed into the optimization algorithms described in the previous section. It is, however, natural to consider more general formulae that offer less conservative solutions (by offering more flexibility) while remaining tractable. One of the options here, called quadratic flow thinning, is discussed below.

With quadratic flow thinning the simple flow defining formula (2e) used in the AFT case is extended with quadratic terms involving link degradation ratios  $\beta$ :

$$x_{dp}^s = x_{dp}^0 - \sum_{e \in \mathcal{E}(d,p)} \beta(e, s) z_{dp}^e + \sum_{\{e, e'\} \in \mathcal{E}^{[2]}(d,p)} \beta(e, s) \beta(e', s) z_{dp}^{ee'}, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), s \in \mathcal{S} \quad (16)$$

where  $\mathcal{E}^{[2]}(d, p)$  denotes the family of all 2-element subsets of  $\mathcal{E}(d, p)$ . As for AFTOP, we assume  $z_{dp}^e \geq 0, z_{dp}^{ee'} \geq 0, d \in \mathcal{D}, p \in \mathcal{P}(d), \{e, e'\} \in \mathcal{E}^{[2]}(d, p)$ . In consequence, the optimization problem for QFT, abbreviated with QFTOP( $\mathcal{P}, \mathcal{S}$ ), is obtained from AFTOP by using equality (16) in (2e), and adding constraint (1e) from FTOP. This additional constraint is necessary because in (16) the quadratic terms with non-negative coefficients  $z_{dp}^{ee'}$  are added to  $x_{dp}^0$ . As AFTOP, this problem requires path generation.

**Example 2:** In order to illustrate the motivation behind introducing the quadratic terms to the affine flow thinning formula let us come back to Example 1 examined in Section 3.2 (and, for that matter, also in [20]). In this case, assuming the affine coefficients  $z^1 = z^2 = 1$  and the nominal flow  $x^0 = 1$  as before, and introducing the quadratic coefficient  $z^{12} = 1$ , we extend the AFT formula (3) to the following QFT formula:

$$x^s = 1 - \beta(1, s) - \beta(2, s) + \beta(1, s)\beta(2, s), \quad s \in \mathcal{S}. \quad (17)$$

Now let us consider a third state,  $s = 3$ , on top of  $s = 1$  and  $s = 2$ . The resulting state list  $\mathcal{S} = \{1, 2, 3\}$  is described in the first three columns of Table 1, where the link degradation ratios for a particular state  $s \in \mathcal{S}$  are specified in column 2 (for link  $e = 1$ ) and in column 3 (for link  $e = 2$ ). Observe that for the first two states the AFT formula (3) and the QFT formula (17) give the same values of the thinned flows  $x^1, x^2$ , since in both cases in the quadratic term one of degradation ratios is equal to 0. However, in state  $s = 3$  (with both links simultaneously degraded) the flow determined with the AFT formula (i.e.,  $x^3 = \frac{1}{4}$ ) is strictly smaller than the flow achieved with the QFT formula (i.e.,  $x^3 = \frac{3}{8}$ ). This shows that using the quadratic term is advantageous from the viewpoint of traffic efficiency. Yet, the maximum flow  $x^3 = \frac{1}{2}$  achieved with FT is still larger.

Finally, observe that the full-availability state  $s = 0$  considered in Example 1 is taken into account by the above QFT formula as well.  $\square$

As illustrated above, adding the quadratic terms makes the use of flow thinning formula more effective in terms of traffic handling (this issue is further addressed in Section 8). However, allowing for quadratic terms

Table 1: Path-flows for AFT and QFT

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

in the thinning formula does not come without a price because it makes both the feasibility tests (13) and the pricing problem for path generation more computationally demanding. As far as the feasibility tests are concerned, the reason is that now the objective functions of the tests contain bi-linearities. More precisely, since the path-flows for a given point  $u \in \mathcal{Q}(N)$  are expressed as

$$x_{dp}(u) = x_{dp}^0(\mathcal{S}) - \sum_{e \in \mathcal{E}(d,p)} \beta(e, u) z_{dp}^e(\mathcal{S}) + \sum_{\{e, e'\} \in \mathcal{E}^{[2]}(d,p)} \beta(e, u) \beta(e', u) z_{dp}^{e e'}(\mathcal{S}),$$

products of the optimization variables composing vector  $u$  (of the form  $u_e^k u_{e'}^{k'}$ ) will appear since

$$\beta(e, u) = 1 - \sum_{k \in \mathcal{K}} a(k) u_e^k \quad \text{and} \quad \beta(e', u) = 1 - \sum_{k' \in \mathcal{K}} a(k') u_{e'}^{k'},$$

and because of that

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

Using the above formulae, the linear (bi-linear) formulations of the feasibility tests (13) adjusted for QFT can be easily obtained from formulations (19)-(21) presented in Appendix A. 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 constraints, and as such can be treated by a quadratic programming solver. It turns out, however, that this would lead to excessive number of SGA iterations. Therefore it is more efficient to use a mixed-integer quadratic programming solver for the modified feasibility tests with the state polytope-defining variables  $u$  assumed to be binary – this version of SGA was used in the calculations reported in Section 8.2.

## 7 Discussion

In fact, the AFT formula (2e) (called *simple* and referred to as SAFT in the following) used so far is only one of many possible. On another extreme is the following *general* AFT formula (GAFT):

$$x_{dp}^s = z_{dp}^0 - \sum_{e \in \mathcal{E}} \beta(e, s) z_{dp}^e \tag{18}$$

where the coefficients  $z$  are unlimited in sign, contrary to SAFT, where all  $z_{dp}^e$  are nonnegative and  $z_{dp}^0 = x_{dp}^0$ . Note that GAFT depends on the degradation ratios  $\beta(\cdot, s)$  of all links in  $\mathcal{E}$ , and not only on the degradation ratios of the links in  $\mathcal{E}(d, p)$  that form the considered path as in SAFT (2e). Clearly, such an extension of the range of the thinning formula (as well as allowing unlimited sign of coefficients  $z$  and introducing  $z_{dp}^0$  instead of  $x_{dp}^0$ ) is advantageous for traffic flows handling (and thus for network cost (2a) minimization) since it makes it possible to better approximate the optimal flows obtained from FTOP formulated in (1).

However, the general AFT formula (18) has a major drawback. Since the concept of flow thinning assumes that the capacity of each (MPLS) tunnel is controlled at its originating node by a packet admission control mechanism based on the on-line knowledge of the currently available link capacity, some kind of the signalling mechanism for interchanging information concerning the current link degradation ratios  $\beta(\cdot, s)$  must be implemented in the network. In the case of the general AFT formula (with the full range  $\mathcal{E}$ ) this requires some kind of a flooding protocol since the tunnels' originating nodes 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 and thus virtually not applicable. Note that this drawback concerns the basic FT mechanism as well, since with FT the optimized capacities of the tunnels are, for all  $s \in \mathcal{S}$ , stored in their originating nodes and activated according to the current values of  $\beta(e, s)$ ,  $e \in \mathcal{E}$ .

Clearly, the above drawback can be eliminated by making GAFT (18) depending, just as SAFT, only on the degradation ratios along the path realizing the given tunnel, i.e., by substituting  $\mathcal{E}$  with  $\mathcal{E}(d, p)$  in the summation on the right-hand side of equation (18). Then, 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. In the following, this version of GAFT will be denoted by GAFT/ $\mathcal{E}(d, p)$ ; symmetrically, formula (18) will be denoted by GAFT/ $\mathcal{E}$ .

In fact, such simplified signalling is also valid for thinning formulae depending on the degradation ratios of all links  $e$  incident to the nodes  $v$  traversed by the considered tunnel. In the following, such defined range will be denoted by  $\mathcal{E}^+(d, p)$ , and formula (18) with range  $\mathcal{E}$  substituted with  $\mathcal{E}^+(d, p)$  will be denoted by GAFT/ $\mathcal{E}^+(d, p)$ . Similar notation, i.e., SAFT/ $\mathcal{E}(d, p)$ , SAFT/ $\mathcal{E}^+(d, p)$ , SAFT/ $\mathcal{E}$  will be used for the three versions of formula (2e) with the summation ranges  $\mathcal{E}(d, p)$ ,  $\mathcal{E}^+(d, p)$ ,  $\mathcal{E}$ , respectively. For a detailed description and classification of feasible AFT formulae the reader is referred to [20]. All these variants of AFT will be considered in Section 8.

Another important issue is the tunnel capacity control in the states unforeseen in optimization, i.e., the states not defined by the state polytope nor on the initial list of states (see Step 1 of SGA+PGA in Section 5.3). Since haphazard setting of tunnel capacity can lead to link overloads and hence to traffic losses, the unconstrained (except for the bound imposed by (1e)) FT mechanism is risky in this aspect, as it does not provide consistent means for this kind of control. On the contrary, the AFT mechanism can simply apply its optimized explicit flow thinning formula also to 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 [16] for the general AFT formula.) Note that with the simple formula (2e) the nominal flows are always thinned whatever the set of degradation ratios  $\beta(\cdot, s)$  is. This makes the tunnel capacity control safer in terms of traffic losses.

To summarize, although the basic FT mechanism with no constraints on thinning (except for (1e)) is the cheapest in terms of the resulting link capacity cost (1a), it is hardly practical due to difficulties with signalling and with control of tunnel capacity outside the state-list considered for dimensioning. Note that the first difficulty makes also the general AFT formula (18) impractical, but its GAFT/ $\mathcal{E}(d, p)$  and GAFT/ $\mathcal{E}^+(d, p)$  versions (as well as SAFT/ $\mathcal{E}(d, p)$  and SAFT/ $\mathcal{E}^+(d, p)$ ) are implementable. Although in terms of cost efficiency, the modified general formula is superior to its simple counterpart (see Section 8.3), the latter is safer to use outside the state space  $\mathcal{S}$  assumed in optimization. We deliberately do not make any firm recommendation for the most suitable version of the AFT formula here, as this requires further studies after extending the optimization model (2) in its state generation part to all feasible AFT formulae. (Path generation algorithms, together with appropriate pricing problems, for the considered AFT formulae are discussed in detail in [20].)

As far as QFT is concerned, all variants of the AFT thinning formulae considered above can be easily transformed to analogous variants for QFT (i.e., to SQFT and GQFT with different ranges) in the same way in which the simple AFT formula (2e) has been transformed to the simple QFT formula (16). Then, the above statements characterizing AFT remain valid for QFT, with one general comment. Adding quadratic terms to the AFT formulae improves the network traffic efficiency and thus leads to less expensive network in terms of link capacity. However, optimization models for QFT become more complicated and time consuming with respect to corresponding model for AFT, both in the state and path generation aspects. This issue is further addressed in Section 8.3. Path generation algorithms for the feasible QFT formulae are presented and discussed in [20].

## 8 Numerical study

In this section we present numerical results that illustrate efficiency of the SGA+PGA iterative optimization procedure for AFT/QFT described in Sections 5.3 and 6, from the computational efficiency and the optimized network cost viewpoints, for different variants of the flow thinning formula. Observe that solving AFTOP for the general case of AFT (and, for that matter, also for the general case of QFT) requires not only an appropriate adjustment of formula (2e) in order to take the extended range into account, but also (for GAFT and QFTP) adding a new constraint, namely constraint (1e) from FTOP, since in the case of GAFT and GQFT, thinning of nominal flows is no longer automatically guaranteed. Moreover, in this case an additional feasibility test to find out whether  $x_{dp}(u) = z_{dp}^0(\mathcal{S}) - \sum_{e \in \mathcal{E}(d,p)} \beta(e, u) z_{dp}^e(\mathcal{S}) \leq x_{dp}^0(\mathcal{S})$  is required (see Section 5.1). In all cases, appropriate pricing problems for path generation (to be found in [20]) were applied.

### 8.1 Network instance

The study was performed for an FSO network instance designed for Paris metropolitan area using realistic data, like population data (to calculate the traffic matrix) and weather records (to calculate typical FSO link degradation ratios). The considered instance (referred to as PMAN) was introduced in [12]. The network (depicted in Figure 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 2. The link capacities are expressed in Gbps as well, while the unit link capacity cost is assumed to be equal to 1 per 1 Gbps for all links (i.e.,  $\xi(e) = 1, e \in \mathcal{E}$ ). It is

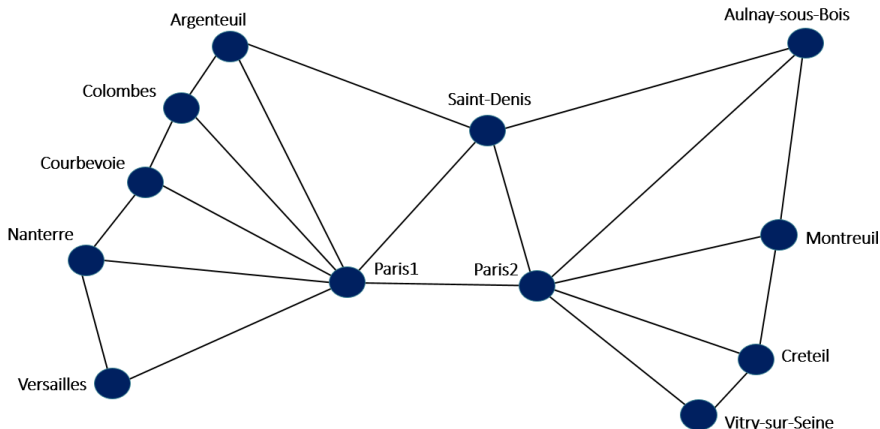


Figure 2: Paris Metropolitan Area network

important to note that although in reality the traffic demands (modelling the Internet traffic) are directed and the FSO links are composed of full-duplex FSO systems (i.e., an FSO link consists of two oppositely directed arcs of the same capacity), the use of undirected links and undirected demands in our optimization model is correct provided the traffic matrix is symmetric (as in PMAN). In such a 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 2) and dimension the network using the undirected network model. Then, in the post processing phase, we can bring back the directions of the demands for the upper-part demands and direct the optimal 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  $v \rightarrow w$  direction, and a set of flows in the  $w \rightarrow v$  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, we consider path-flows for the directed demands from the lower part symmetric (below the diagonal) of the traffic matrix that are symmetrical (but oppositely directed) to their counterparts from the upper-part of the matrix. 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

realise 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 both directions.

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.

Table 2: 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 | –   | –     | –    | –    | –    | –    | –    | –    | –    | –    | –    | –    |

The results of the numerical study are presented in Tables 3-8. 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 12.8.0.0.

## 8.2 Efficiency of SGA+PGA

In Table 3 we report the results of solving AFTOP (2) – the dimensioning problem assuming SAFT/ $\mathcal{E}(d, p)$  – the simple affine flow thinning formula (2e). We examine six sets of states  $\hat{\mathcal{B}}(N, a, b, H)$  (rows 1-6 in Table 3). Recall that, considering for example row no. 3, the parameters (21, 1, 2), (1, 0.75, 0.75), and (0, 0, 0.05) imply that the corresponding state set includes all combinations of simultaneous degradation of  $U^2 = 0, 1$  links that are degraded to availability ratio 0.75, and  $U^3 = 0, 1, 2$  links degraded also to availability ratio 0.75 (then  $U^1 := 21 - U^2 - U^3$  links are fully available). When state with  $(U^1, U^2, U^3)$  is observed, the traffic of each demand  $d \in \mathcal{D}$  is equal to  $H(d)(1 - 0.05U^3)$  (i.e., the reference demand volumes are reduced only when  $U^3 > 0$ ). It should be noted here that the reference demand values  $H(d)$ ,  $d \in \mathcal{D}$  are taken from Table 2 assuming the lexicographical order of the node pairs (1, 2), (1, 3), ..., (11, 12).

Analogous results for GAFT/ $\mathcal{E}(d, p)$ , i.e., the results of solving AFTOP with (2e) substituted with formula (18) with range  $\mathcal{E}(d, p)$  are presented in Table 4.

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 (i.e., to the number of hops, as  $\xi(e) \equiv 1$ ). 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 ratio  $\alpha(e, s) = 0.75$  (22 states in total). For all the states in the initial list of states 100% traffic protection is assumed, i.e.,  $h(d, s) = H(d)$ ,  $d \in \mathcal{D}$ . 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  $F^* = 246.38$ .



Tables 3 and 4 show that the SGA+PGA is computational efficient. It requires up to 10 – 12 iterations (column “iter”) and generates a reasonable number of paths (column “ $\Delta\mathcal{P}$ ”) and states (column “ $\Delta\mathcal{S}$ ”). 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 Step 2. We notice that although the total execution times are typically (column “total [s]”) of the order of hours, they are acceptable considering the size of the problems and the fact that they are not supposed to be solved online but rather once in the network management cycle.

Table 3: Results of SGA+PGA for SAFT/ $\mathcal{E}(d, p)$

| $N(k)$        | $a(k)$            | $b(k)$          | $F^*$  | $\Delta\mathcal{P}$ | $\Delta\mathcal{S}$ | iter | total [s] | t/PG [s] | t/SG [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     |

**Explanation:**  $N(k)$ ,  $a(k)$ ,  $b(k)$ , state polytope parameters;  $F^*$ , cost of the optimal solution;  $\Delta\mathcal{P}$ , number of generated paths;  $\Delta\mathcal{S}$ , number of generated states; iter, number of iterations of SGA+PGA; total, total computation time; t/PG, computation time spent in PGA( $\mathcal{S}$ ); t/SG, computation time spent in SGA( $\mathcal{P}$ ).

Table 4: Results of SGA+PGA for GAFT/ $\mathcal{E}(d, p)$

| $N(k)$        | $a(k)$            | $b(k)$          | $F^*$  | $\Delta\mathcal{P}$ | $\Delta\mathcal{S}$ | iter | total [s] | t/PG [s] | t/SG [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    |

In Table 5 we compare performance of the simple AFT formula (SAFT) and the general AFT formula (GAFT) for two ranges:  $\mathcal{E}(d, p)$  and  $\mathcal{E}^+(d, p)$ . In the table, the asterisks in the last column denote that the result was obtained through applying AFTOP (only once) to the final path-lists and state-lists obtained for the range  $\mathcal{E}(d, p)$ . Finally, the percentages in column 8 give the cost increase of SAFT/ $\mathcal{E}^+(d, p)$  as compared to GAFT/ $\mathcal{E}^+(d, p)$ . Note that the network cost for both SAFT variants considered in Table 5 is the same.

A general conclusion is that optimization of GAFT is more time consuming than SAFT, especially the  $\mathcal{E}^+(d, p)$  version, but its cost can be noticeably lower than for SAFT.

Table 5: Comparison of network cost and computation time of SGA+PGA for different AFT formulae

| $N(k)$        | $a(k)$            | $b(k)$          | SAFT/ $\mathcal{E}(d, p)$ |           | GAFT/ $\mathcal{E}(d, p)$ |           | SAFT/ $\mathcal{E}^+(d, p)$ |           | GAFT/ $\mathcal{E}^+(d, p)$ |           |
|---------------|-------------------|-----------------|---------------------------|-----------|---------------------------|-----------|-----------------------------|-----------|-----------------------------|-----------|
|               |                   |                 | $F^*$                     | total [s] | $F^*$                     | total [s] | $F^*$                       | total [s] | $F^*$                       | total [s] |
| (21, 1)       | (1, .75)          | (0, 0)          | 312.36                    | 161       | 312.36                    | 408       | 312.36 (0%)                 | 1*        | 312.36                      | 1*        |
| (21, 1, 1)    | (1, .75, .75)     | (0, 0, .05)     | <i>325.72</i>             | 1477      | <i>325.72</i>             | 4544      | <i>325.72</i> (3%)          | 23*       | <i>317.59</i>               | 57*       |
| (21, 1, 2)    | (1, .75, .75)     | (0, 0, .05)     | 326.83                    | 4288      | 326.83                    | 25891     | 326.83 (3%)                 | 31*       | 317.91                      | 307*      |
| (21, 1, 2, 1) | (1, .75, .75, .5) | (0, 0, .05, .1) | 432.30                    | 5853      | 432.30                    | 18668     | 432.30 (6%)                 | 171*      | 409.38                      | 1553*     |
| (21, 1, 2, 2) | (1, .75, .75, .5) | (0, 0, .05, .1) | 448.58                    | 12296     | 448.58                    | 31459     | 448.58 (8%)                 | 150*      | 415.27                      | 1757*     |
| (21, 1, 2, 3) | (1, .75, .75, .5) | (0, 0, .05, .1) | 449.98                    | 7534      | 449.98                    | 24725     | 449.98 (8%)                 | 80*       | 416.99                      | 896*      |

Table 6 presents results of SGA extended to quadratic FT formula (16). The calculations were performed for fixed path-lists obtained from the respective final path-lists in the solutions presented in Table 5. As expected, the quadratic formulae lead to less expensive networks in terms of the link capacity cost (by about 1-4%, see the  $F^*$  values in italics in the second row in Tables 5 and 6). However, due to excessive computation

time, it was possible to solve the optimization problem only for the first two state polytopes. Note that the costs in the first rows of Tables 5 and 6 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 SQFT/ $\mathcal{E}^+(d, p)$  case is smaller than the cost ( $F^* = 316.44$ ) for the GQFT/ $\mathcal{E}^+(d, p)$  case is that path-lists for QFT were not optimized but just taken from the corresponding AFT solutions.

Table 6: Results of SGA for different QFT formulae

| $N(k)$     | $a(k)$        | $b(k)$      | SQFT/ $\mathcal{E}(d, p)$ |           | GQFT/ $\mathcal{E}(d, p)$ |           | SQFT/ $\mathcal{E}^+(d, p)$ |           | GQFT/ $\mathcal{E}^+(d, p)$ |           |
|------------|---------------|-------------|---------------------------|-----------|---------------------------|-----------|-----------------------------|-----------|-----------------------------|-----------|
|            |               |             | $F^*$                     | total [s] | $F^*$                     | total [s] | $F^*$                       | total [s] | $F^*$                       | total [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) | <i>317.53</i>             | 54        | <i>316.52</i>             | 145       | <i>313.99</i>               | 184       | <i>316.44</i>               | 483       |

To end this section we mention that link availability ratios considered in the above tables, i.e., 1.0, 0.75 and 0.5, correspond to the modulation and coding schemes applicable to FSO links; see the discussion in [12] based on [23, 11].

### 8.3 Network cost efficiency

Below, in Table 7, we briefly summarize the cost efficiency of the FT/AFT mechanisms, in particular in comparison with Global Rerouting (GR). Recall that GR is a well known protection mechanism which allows for unrestricted flow rearrangement whenever a network state is changed (GR is called unrestricted reconfiguration in [21]). This feature makes GR impractical, but at the same time makes it a benchmark mechanism, as it provides the lower bound on the network cost achievable with any other mechanism. The comparison encompasses GR, FT, GAFT and SAFT, and is based on explicit (predefined) lists of states (SL – single link degradations, DL – double link degradations, TL – triple link degradations) fully described in Section 6.1 of [20]. (The state polytope model is not used here since formulation (2) and the SGA+PGA algorithm are not applicable to GR and FT.) The percentages in the FT column express the increase of the network cost for FT in comparison with the cost for GR, while the percentages in the GAFT/ $\mathcal{E}$  column show the increase of the cost for GAFT/ $\mathcal{E}$  in comparison with the cost for FT. In all other columns, the percentages compare the reported cost to the cost for GAFT/ $\mathcal{E}$ .

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

Recall that the decrease in network cost observed when the AFT formula is substituted by its QFT counterpart for the cases reported in Tables 5 and 6 was merely between 1 and 4 percents. However, such a decrease can be more prominent, as illustrated in Table 8. This table shows the results obtained for another 12-node network called *polska* (its data is available in the library SNDlib, sndlib.zib.de, see [15]) that was studied in our previous papers, for example in [16] and recently in [20]. In Table 8 all six variants of the QFT 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 AFT formula is applied. This time the gain from QFT can be quite substantial, ranging from 4 to 17 percents.

Table 7: Comparison of the network cost for different AFT formulae

|    | GR     | FT          | GAFT/ $\mathcal{E}$ | SAFT/ $\mathcal{E}$ | GAFT/ $\mathcal{E}^+(d, p)$ | SAFT/ $\mathcal{E}^+(d, p)$ | GAFT/ $\mathcal{E}(d, p)$ | SAFT/ $\mathcal{E}(d, p)$ |
|----|--------|-------------|---------------------|---------------------|-----------------------------|-----------------------------|---------------------------|---------------------------|
| 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%)               |

Table 8: Comparison of the network cost for AFT and QFT formulae

|    | GQFT/ $\mathcal{E}$ | SQFT/ $\mathcal{E}$ | GQFT/ $\mathcal{E}^+(d, p)$ | SQFT/ $\mathcal{E}^+(d, p)$ | GQFT/ $\mathcal{E}(d, p)$ | SQFT/ $\mathcal{E}(d, p)$ |
|----|---------------------|---------------------|-----------------------------|-----------------------------|---------------------------|---------------------------|
| 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%)               |

## 9 Final remarks

The optimization model for AFT network dimensioning introduced in this paper can be helpful in practical design of FT networks robust to link capacity fluctuations. First, the use of AFT formulae for tunnel thinning is relatively simple to implement using existing network protocols. Second, the state description (using the state polytope) applied in the optimization model makes it capable of taking into account a great deal of those link availability states that can occur during network operation (i.e., against which the network should be made robust). Finally, as discussed in [16], the opportunity provided by the use of affine thinning formulae, i.e., automatic adjustment of the tunnel capacity at the originating nodes to current link availability ratios (once they become known to the tunnel originating node), enables AFT to work properly also in the states outside the state polytope. This is because for such states usually acceptable traffic losses are experienced even though the AFT formulae are not optimized for those states. This is implied by the very nature of the AFT that throttles the traffic at the originating nodes according to a consistent formula dependent on the link degradation ratios.

Besides affine FT, we considered also its quadratic extension which leads to more efficient traffic handling and thus requires less link capacity.

Numerical results show that the optimization procedure introduced in Section 5.3 is computationally efficient, despite the binary pricing problems required in the path generation step. In fact, an analogous optimization model can be found in [8] and in its journal version [12]. However, that model uses a simplified, less practical state polytope description, and assumes an impractical Global Rerouting mechanism (characterized in Section 8.3) which is much simpler to deal with than AFT/QFT, as it does not require path generation and poses a compact linear programming formulation (and hence does not require path generation) and a simpler state generation algorithm.

As for the future work, the introduced model will be extended to modular link capacities. This feature will transform AFTOP to a mixed-integer programming problem with integer variables  $y^0$ , and will most likely require additional valid inequalities to keep the solution approach efficient. Also, a more comprehensive numerical study involving extended versions of the state polytope, not considered in this paper, is planned. Finally, as already observed in the numerical section, the simple AFT formula with the range limited to the path’s links turns out to be as efficient as its versions with the extended range. Most likely this is not a coincidence but a general property, at least for the kind of state polytopes considered. This question should definitely be clarified.

## Acknowledgements

The Polish authors were supported by the National Science Centre, Poland, grant number 2015/17/B/ST7/03910 “Logical tunnel capacity control - a traffic routing and protection strategy for communication networks with variable link capacity”. The work of D. Nace was carried out in the Labex MS2T framework funded by the French Government through the program “Investments for the future” managed by the National Agency for Research [Ref. ANR-11-IDEX-0004-02].

## References

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, NJ, 1993.
- [2] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

- [3] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.
- [4] A. Ben-Tal and A. Nemirovski. Robust Convex Optimization. *Math. Oper. Res.*, 23(4):769–805, 1998.
- [5] J. Benders. Partitioning procedures for solving mixed variable programming problems. *Numerische Mathematik*, 4:238–252, 1962.
- [6] J. F. Benders. Partitioning procedures for solving mixed variable programming problems. *Numerische Math.*, 4:238–252, 1962.
- [7] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.
- [8] 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.
- [9] B. Korte and J. Vygen. *Combinatorial Optimization Theory and Algorithms*. Springer, 2012.
- [10] 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.
- [11] J. Malhotra, M. Kumar, and A. 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.
- [12] 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.
- [13] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.
- [14] S. Orłowski and M. Pióro. Complexity of column generation in network design with path-based survivability mechanisms. *Networks*, 59(1):132–147, 2012.
- [15] S. Orłowski, M. Pióro, A. Tomaszewski, and R. Wessälly. SNDlib 1.0 – Survivable Network Design Library. *etworks*, 55(3):276–285, 2010.
- [16] M. Pióro, Y. Fouquet, D. Nace, and M. Poss. Optimizing flow thinning protection in multicommodity networks with variable link capacity. *Operations Research*, 64(2):273–289, 2016. Best paper award in INFORMS Telecom and Network Analytics 2016-2018.
- [17] 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.
- [18] M. Pióro, I. Kalesnikau, and M. Poss. Path Generation for Affine Flow Thinning. *Electronic Notes in Discrete Mathematics*, 64(0):355–364, 2018.
- [19] 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)*, 2018.
- [20] 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.
- [21] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann, 2016.
- [22] J. Rak. *Resilient Routing in Communication Networks*. Springer International Publishing, 2015.
- [23] A. Svensson. An Introduction to Adaptive QAM Modulation Schemes for Known and Predicted Channels. *Proceedings of the IEEE*, 95(12):2322–2336, 2007.

- [24] A. Tomaszewski, M. Pióro, and M. Żotkiewicz. On the complexity of resilient network design. *Networks*, 55:109–118, 2010.
- [25] R. Wessäly, S. Orlowski, A. Zymolka, A. Koster, and C. 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.

## A Formulations of the feasibility tests

The linear programming formulations of feasibility tests (13) applied in state generation are as follows.

$$T(d) = \max \{ h_d - \sum_{p \in \mathcal{P}(d)} x_{dp} \} \quad (19a)$$

$$\sum_{k \in \mathcal{K}} u_e^k = 1, \quad e \in \mathcal{E} \quad (19b)$$

$$\sum_{e \in \mathcal{E}} u_e^k \leq N^k, \quad k \in \mathcal{K} \quad (19c)$$

$$U^k = \sum_{e \in \mathcal{E}} u_e^k, \quad k \in \mathcal{K} \quad (19d)$$

$$\beta_e = 1 - \sum_{k \in \mathcal{K}} a(k) u_e^k, \quad e \in \mathcal{E} \quad (19e)$$

$$h_d = H(d)(1 - \sum_{k \in \mathcal{K}} b(k) U^k) \quad (19f)$$

$$x_{dp} = x_{dp}^0(\mathcal{S}) - \sum_{e \in \mathcal{E}(d,p)} \beta_e z_{dp}^e(\mathcal{S}), \quad p \in \mathcal{P}(d) \quad (19g)$$

$$u, U, \beta, h \text{ non-negative continuous; } x \text{ continuous.} \quad (19h)$$

$$T(e) = \max \{ \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{R}(d,e)} x_{dp} - \alpha_e y_e^0(\mathcal{S}) \} \quad (20a)$$

$$\sum_{k \in \mathcal{K}} u_e^k = 1, \quad e \in \mathcal{E} \quad (20b)$$

$$\sum_{e \in \mathcal{E}} u_e^k \leq N^k, \quad k \in \mathcal{K} \quad (20c)$$

$$\alpha_e = \sum_{k \in \mathcal{K}} a(k) u_e^k \quad (20d)$$

$$\beta_e = 1 - \sum_{k \in \mathcal{K}} a(k) u_e^k, \quad e \in \mathcal{E} \quad (20e)$$

$$x_{dp} = x_{dp}^0(\mathcal{S}) - \sum_{e \in \mathcal{E}(d,p)} \beta_e z_{dp}^e(\mathcal{S}), \quad d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (20f)$$

$$u, U, \alpha, \beta, h \text{ non-negative continuous; } x \text{ continuous.} \quad (20g)$$

$$T(d, p) = \max \{ -x_{dp} \} \quad (21a)$$

$$\sum_{k \in \mathcal{K}} u_e^k = 1, \quad e \in \mathcal{E} \quad (21b)$$

$$\sum_{e \in \mathcal{E}} u_e^k \leq N^k, \quad k \in \mathcal{K} \quad (21c)$$

$$\beta_e = 1 - \sum_{k \in \mathcal{K}} a(k) u_e^k, \quad e \in \mathcal{E} \quad (21d)$$

$$x_{dp} = x_{dp}^0(\mathcal{S}) - \sum_{e \in \mathcal{E}(d,p)} \beta_e z_{dp}^e(\mathcal{S}), \quad d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (21e)$$

$$u, U, \alpha, \beta \text{ non-negative continuous; } x \text{ continuous.} \quad (21f)$$

Recall that above  $y^0(\mathcal{S}), x^0(\mathcal{S}), z(\mathcal{S})$  are vectors of parameters, while  $u, U, \alpha, \beta, h, x$  are vectors of variables.