



HAL
open science

Optimizing Flow Thinning Protection in Multicommodity Networks with Variable Link Capacity

Michal Pioro, Yoann Fouquet, Dritan Nace, Michael Poss

► **To cite this version:**

Michal Pioro, Yoann Fouquet, Dritan Nace, Michael Poss. Optimizing Flow Thinning Protection in Multicommodity Networks with Variable Link Capacity. *Operations Research*, 2016, 64 (2), pp.273-289. 10.1287/opre.2016.1486 . hal-01298600

HAL Id: hal-01298600

<https://hal.science/hal-01298600>

Submitted on 30 Oct 2019

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.

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

Optimizing Flow Thinning Protection in Multicommodity Networks with Variable Link Capacity

Michał Pióro

Institute of Telecommunications, Warsaw University of Technology, Poland, mpp@tele.pw.edu.pl
Department of Electrical and Information Technology, Lund University, Sweden, michal.pioro@eit.lth.se

Yoann Fouquet, Dritan Nace

UMR CNRS 7253 Heudiasyc, Université de Technologie de Compiègne,
Centre de Recherches de Royallieu, 60200 Compiègne, France, yoann.fouquet,dritan.nace@hds.utc.fr

Michael Poss

UMR CNRS 5506 LIRMM, Université de Montpellier,
161 rue Ada, 34392 Montpellier Cedex 5, France, michael.poss@lirmm.fr

Flow thinning (FT) is a concept of a traffic routing and protection strategy applicable to communication networks with variable capacity of links. In such networks the links do not attain their nominal (maximum) capacity simultaneously, so that in a typical network state only some links are fully available while on each of the remaining links only a fraction of its maximum capacity is usable. Every 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 to account for variable capacity of the links fluctuating below the maximum. Accordingly, the capacity available on the tunnels is also fluctuating below their nominal levels and hence the instantaneous traffic sent between the demand's end nodes must accommodate to the current total capacity available on its dedicated tunnels. The related multi-commodity flow optimization problem is \mathcal{NP} -hard and its non-compact linear programming formulation requires path generation. For that, we formulate an integer programming pricing problem, at the same time showing the cases when the pricing is polynomial. We also consider an important variant of FT, affine thinning, that may lead to practical FT implementations. We present a numerical study illustrating traffic efficiency of FT and computational efficiency of its optimization models. Our considerations are relevant, among others, for wireless mesh networks utilizing MPLS tunnels.

Key words: Survivable network design; Multiple partial link failures; Mixed-integer programming;

Multicommodity flows; Path generation; Robust optimization

Area of review: Games, Information, and Networks

1. Introduction

The paper is devoted to optimization models for *flow thinning* (FT)—a concept of a generic traffic routing and protection strategy applicable to communication networks with variable link capacity. The concept of FT, preliminarily presented in Pióro et al. (2013), is as follows.

FT (abbreviations are listed in Section EC.1 of the electronic companion) assumes that at any instant of time the network is in some (link availability) state characterized, for each link, by the fraction of its currently available nominal (maximal) capacity. Typically, nominal capacity is not simultaneously available on all links, as the usable capacity of each link fluctuates between zero and its nominal value. The state with all links fully available is called the nominal state.

FT makes use of logical tunnels. Each traffic demand is assigned a fixed set of tunnels, each realized along a selected path connecting the end nodes of the demand, whose joint capacity is dedicated to carry the demand (packet) traffic. The maximum (logical) capacity reserved on a tunnel is called its nominal capacity. The actual capacity of every tunnel is state-dependent and controlled on-line at its originating node in response to fluctuations in the capacity currently available on the network links. For that, FT requires a mechanism (protocol) for broadcasting link-status messages. When a change in a link availability occurs, its originating node sends an appropriate message to the network nodes that require this information tunnel capacity adjustment. The adjustment of the tunnel capacity consists in thinning, to the degree depending on the current link availability state, its nominal capacity. Clearly, capacity of the tunnels should be controlled in a consistent way, so that in each state the link loads (link load is the sum of the capacity of all the tunnels using the link) will not exceed their current capacity when the capacity currently assigned to the tunnels is fully utilized by the packet traffic.

For each demand, the instantaneous traffic admitted to each of its dedicated tunnels at the originating node cannot exceed the current tunnel capacity. Hence, FT operation requires a traffic admission mechanism that enables the demand originating node to locally control the amount of the demand traffic admitted to enter each of its dedicated tunnels.

The instantaneous traffic realized for a given demand is limited by the current overall capacity of its dedicated tunnels. In order to satisfy the demands, the tunnels, and consequently the links, should be properly dimensioned so that the state-dependent overall capacity of the dedicated tunnels is always sufficient to carry a fraction of the nominal traffic volume acceptable for every demand. Certainly, if the nominal traffic is to be always carried regardless of the state, the nominal capacity of the tunnels (and hence the nominal capacity of links) has to be made redundant so that after any thinning the tunnel capacity will still be sufficient to carry the nominal traffic. In fact, to avoid substantial network over-dimensioning, and hence excessive network cost, the demands

should in general accept a reasonable decrease of their nominal traffic volumes, at least in the states with significant capacity reduction, and the state-dependent reductions of demands' nominal traffic volumes should reflect the fractions of the nominal link capacity available in a given state. Such reductions, acceptable for the users, should be determined by the operator using the optimization models like those developed in the paper assuming realistic link availability state scenarios.

We defer the discussion of the FT application, implementation and practicality issues to Section 9, where we in particular discuss an important FT implementation applicable to Wireless Mesh Networks (WMN) (Mogre et al. 2007)), based on Label Switched Paths (LSP), i.e., logical tunnels realized in the Multiprotocol Label Switching (MPLS) technology (Minei and Lucek 2011), with link-status information exchange realized by means of an adjusted version of the Resource Reservation Protocol – Traffic Engineering (RSVP-TE) (Minei and Lucek 2011)) and a packet traffic admission control mechanism based on low-level QoS packet flow shaping and policing mechanism such as for example leaky bucket (Harhira and Pierre 2007)).

The main body of our paper deals with multi-state optimization of resilient multicommodity flow networks. The advances in this area are remarkable, see for example Wessály (2000), Grover (2003), Pióro and Medhi (2004), and the references therein. However, most of the reported research is not well fit to our purposes since in our optimization models we look at resilience from a non-standard angle: we treat partial availability of multiple links as typical and deal with a large set of link availability states (associated with appropriate reduction of traffic demand) that can occur in network operation. This viewpoint is valid, and even necessary, for dealing with wireless networks in general, and with WMNs in particular (see Section 9.2). The reason is two-fold. First, the notion of the state has typically been limited to total failures (when a link fails then its entire capacity is lost) of single links (all the links can fail but only one at a time). The multiple link failure case has also been studied but again assuming totally failing links (*shared-risk link group*—SRLG (Strand et al. 2001)), for a discussion see for example Orłowski and Pióro (2012)). In fact, the only problem relevant to ours that we have found in the literature is the problem studied in Classen et al. (2011) in the context of microwave links. It uses chance constrained programming with independent random variables characterizing links' availability (our approach assumes explicit description of the state space). The problem assumes static routing and realization (with a given probability) of nominal traffic in all the states, thus considerably simplifying the problem.

The second reason of limited practicability of the research reported in the literature is that it has been focussed on traffic protection strategies, mostly on traffic flow rerouting strategies (Dahl and Stoer 1998, Wessály 2000, Pióro and Medhi 2004) that are not applicable to multiple partial failures. To the best of our knowledge, only the so called global rerouting (GR) has been studied in the partial multiple failure context, see Pióro and Medhi (2004) and the references therein. GR

restores the demand traffic by establishing path-flows (path-flow corresponds to capacity of a FT tunnel) from scratch in the surviving link capacity, i.e., through a mechanism quite impractical in communication networks due to excessive end-to-end flow rerouting (see Section 9.4).

FT can be considered as an extension of *demand-wise shared protection* (DWSP) developed in Koster et al. (2005) and Wessály et al. (2005) (DWSP was also studied in Tomaszewski et al. (2010), Orłowski and Pióro (2012) under the name *path diversity protection*) to multiple partial link failure scenarios. DWSP assumes multiple total link failures. In a failure state the path-flows using the failed links are simply disconnected and the surviving path-flows must be sufficient to realize the demand traffic volumes, possibly decreased with respect to the nominal volumes. FT is also related to *elastic rerouting* (ER) (Shinko et al. 2013). As DWSP, ER assumes multiple total link failures but allows for decreasing path-flows of the demands unaffected by a failure, as well as for increasing (to a limited extent) path-flows of the affected demands.

The main purpose of this paper is to study in detail original, theoretically interesting optimization problems for FT—a concept of novel generic traffic routing and protection strategy. We consider two versions of the problem. The first version (FTOP) corresponds to the basic version of FT which does not impose any restrictions on the tunnel capacity thinning. The second version (AFTOP) is related to affine flow thinning (AFT), a modification of FT that imposes (affine) restrictions on flow thinning and thanks to that leads to practical implementations of FT.

The paper is organized as follows. In Section 2 we study the basic optimization problem (FTOP) related to FT, consisting in selection of the sets of tunnels (and their nominal and state-dependent capacities) and link dimensioning, minimizing the cost of links for a given list of link availability states and the traffic volumes to be carried at each state. As FTOP is \mathcal{NP} -hard, we approach it using a non-compact linear programming formulation accompanied with an appropriate (unavoidably \mathcal{NP} -hard) pricing problem (Section 3) used for path generation (Section 4). In Section 5, we exploit specific properties of FTOP to obtain polynomial solution algorithms for certain special cases and discuss complexity computational complexity of the pricing problem. In Section 6 we present a compact linear lower bound formulation for FTOP. In Section 7, we study AFTOP using new kind of uncertainty sets, derived from the budgeted uncertainty set notion of Bertsimas and Sim (2004), that allow to partially relax some of the problem constraints when the uncertainty (in regard to link capacity in our case) is profound. In this way AFTOP effectively covers more general sets of link availability states. These theoretical investigations are illustrated in Section 8 with the results of an extensive numerical study on traffic and cost efficiency of FT and AFT in comparison with GR (global rerouting—a benchmark protection strategy), as well as computational efficiency of the developed optimization approaches to FTOP and AFTOP. Application, implementation and practicality issues of FT are discussed in Section 9, followed by concluding remarks given in Section 10. The electronic companion complements and extends the considerations of the paper.

2. FTOP—flow thinning optimization problem

Below, after introducing the notation, we formulate the basic problem studied in this paper. In the problem, we minimize the cost of nominal link capacity that supports a set of nominal tunnel capacities (i.e., path-flows) that are subject to thinning in each of the considered network states.

2.1 Notation

The network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (undirected or directed) is composed of the set of nodes \mathcal{V} and the set of links \mathcal{E} . We will by default assume directed graphs where each link $e \in \mathcal{E}$ represents a directed pair (v, w) of nodes $v, w \in \mathcal{V}$. The set of links incident to node $v \in \mathcal{V}$ is denoted by $\delta(v)$ and $\delta(v) := \delta^-(v) \cup \delta^+(v)$ where $\delta^-(v)$ and $\delta^+(v)$ are the sets of all links incoming to and outgoing, respectively, from node v . Each link e is assigned a unit capacity cost ξ_e (a parameter) and the nominal capacity y_e^0 (an optimization variable). Traffic demands are represented by the set \mathcal{D} . Each demand $d \in \mathcal{D}$ is characterized by a directed pair $(o(d), t(d))$, composed of the originating and the terminating node, respectively, and a given value h_d^0 (a parameter) of the nominal traffic volume. 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 from $o(d)$ to $t(d)$. (Recall that an elementary path does not traverse any node more than once.) Paths in \mathcal{P}_d , used to realize the demand (traffic) volumes, are assigned nominal flows x_{dp}^0 , $p \in \mathcal{P}_d$, which are optimization variables. Each x_{dp}^0 specifies the nominal capacity (expressed in the same units as link capacity and demand volume) reserved on the tunnel realized by 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 from $o(d)$ to $t(d)$, will be denoted by $\hat{\mathcal{P}}_d, d \in \mathcal{D}$, with $\hat{\mathcal{P}} := \bigcup_{d \in \mathcal{D}} \hat{\mathcal{P}}_d$.

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

The capacity of links is variable, and typically not all the links reach their maximum (nominal) capacity simultaneously. At any time instant, nominal link capacities defined by the vector $y^0 = (y_e^0, e \in \mathcal{E})$ are typically available only for a subset of links, and the remaining links have the available capacity reduced. We thus specify a set of link availability states \mathcal{S} , referred to as the *state scenario*. Each state $s \in \mathcal{S}$ is characterized by the two fixed vectors with the components

between 0 and 1: the link availability coefficient vector $\alpha^s := (\alpha_e^s, e \in \mathcal{E})$, and the demand reduction coefficient vector $\beta^s := (\beta_d^s, d \in \mathcal{D})$. By definition, the capacity of link e available in state s is equal to $y_e^s := \alpha_e^s y_e^0$ while the traffic volume of demand d to be carried in s is given by $h_d^s := \beta_d^s h_d^0$.

Any state s with all link coefficients α_e^s equal to 1 is called the maximum (availability) state, and the subset of all such states in the given scenario \mathcal{S} is denoted by \mathcal{S}_0 . Note that the states in \mathcal{S}_0 differ by the vectors β^s only, so that the case $\mathcal{S} = \mathcal{S}_0$ can be used to model a network without link capacity reduction but taking into account different traffic demand vectors. In general, however, in a given state $s \in \mathcal{S}$, one or more links can have their availability coefficients smaller than 1 so our model includes multiple partial failures of the network links under varying state-dependent traffic.

The set of links that are not fully available in state $s \in \mathcal{S}$ will be denoted by $\mathcal{E}_s := \{e \in \mathcal{E} : \alpha_e^s < 1\}$, and, symmetrically, the set of states in which link $e \in \mathcal{E}$ is not fully available, by $\mathcal{S}_e := \{s \in \mathcal{S} : \alpha_e^s < 1\}$. Note that $\mathcal{E}_s = \emptyset$ for each $s \in \mathcal{S}_0$. The particular maximum state s_0 with $\beta_d^{s_0} = 1, d \in \mathcal{D}$, is called the nominal state. It is the state with all links fully available that realizes the nominal traffic demand vector $h^0 := (h_d^0, d \in \mathcal{D})$. We do not necessarily require that $s_0 \in \mathcal{S}_0$ since the nominal state may never occur during network operation. In fact, we do not even require that $\mathcal{S}^0 \neq \emptyset$.

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 $s \in \mathcal{S} \setminus \mathcal{S}_0$, 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 (reduced) demand vector $h^s := (h_d^s, d \in \mathcal{D})$. 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 and are optimization variables. As only thinned nominal flows are allowed, $x_{dp}^s \leq x_{dp}^0, s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}^d$.

An illustration of the above considerations is given in the electronic companion (Example EC.1).

2.2 Basic problem formulation

Below we will introduce a formulation of FTOP (*flow thinning optimization problem*)— an optimization problem for FT. For given admissible path-lists $\mathcal{P}_d, d \in \mathcal{D}$, the considered problem is denoted by $P(\mathcal{P})$ (where $\mathcal{P} = \bigcup_{d \in \mathcal{D}} \mathcal{P}_d$) and represented by the following link-path linear programming (LP) formulation involving variables $x_{dp}^0 (d \in \mathcal{D}, p \in \mathcal{P}_d)$, $x_{dp}^s (d \in \mathcal{D}, p \in \mathcal{P}_d, s \in \mathcal{S})$ and $y_e^0 (e \in \mathcal{E})$:

$$\mathbf{Problem } P(\mathcal{P}): \quad C(\mathcal{P}) = \min \sum_{e \in \mathcal{E}} \xi_e y_e^0 \quad (1a)$$

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

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

$$[\pi_e^s \geq 0] \quad \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} x_{dp}^s \leq \alpha_e^s y_e^0, \quad e \in \mathcal{E}, \quad s \in \mathcal{S}_e \quad (1d)$$

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

$$x^0, x^s, y^0 \geq 0 \text{ and continuous} \quad (1f)$$

(where the quantities in the square brackets on the left-hand side are dual variables to be used later). Objective (1a) minimizes the overall cost of links. The first capacity constraint (1b) makes sure that the nominal link loads do not exceed the nominal link capacities. Next, the demand constraint (1c) assures that in each state $s \in \mathcal{S}$, the thinned nominal flows are sufficient to realize the volume of each demand $d \in \mathcal{D}$ assumed for this state. Then, the second capacity constraint (1d) assures that in each state $s \in \mathcal{S}$, the available capacity of each link $e \in \mathcal{E}_s$ is not exceeded. Finally, constraint (1e) assures thinning. 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, $\alpha_e^s = 1$) such a constraint would be redundant—it is implied by (1b), as the link load $\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} x_{dp}^s$ does not exceed its nominal load $\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} x_{dp}^0$ due to constraint (1e). The optimal value of the cost (1a) of $P(\mathcal{P})$ is denoted by $C(\mathcal{P})$.

Observe that our assumption of admitting only elementary paths on the path-lists made in Section 2.1 is not limiting. If we admitted non-elementary paths in the overall path-list \mathcal{P} , then eliminating loops in any feasible solution of $P(\mathcal{P})$ would never increase $C(\mathcal{P})$. This is true because if path p' is an elementary version of path $p \in \mathcal{P}_d$, then adding p' to \mathcal{P}_d , putting $x_{dp'}^0 := x_{dp}^0$ and $x_{dp'}^s := x_{dp}^s, s \in \mathcal{S}$, and finally setting x_{dp}^0 and $x_{dp}^s, s \in \mathcal{S}$, to 0 would lead to a feasible solution of (1) which does not require the capacity reserved on the links in $p \setminus p'$. The last statement also implies that non-elementary paths will not appear in optimal solutions of (1) when $\xi_e > 0, e \in \mathcal{E}$.

An illustration of the above considerations is given in the electronic companion (Example EC.2).

2.3 Complexity of FTOP, global rerouting, and flow thickening

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

polynomial with the size of the network, then problem $P(\mathcal{P})$ is polynomial as its LP formulation $\text{FTOP}^<$ becomes compact (provided the number of states is polynomial).

Difficulty of FTOP stems from the thinning constraint (1e). In fact, when constraints (1e), together with the associated nominal path-flow variables x^0 and constraints (1b), are deleted from formulation (1), then the resulting optimization problem corresponds to the so called *global rerouting* (GR) flow restoration strategy, also referred to as unrestricted flow reconfiguration or dynamic routing, see Pióro and Medhi (2004), Orlowski and Pióro (2012). In GR, the flows x^s realizing the demands in different states are independent of each other and are only coupled by the capacity constraints (1d). This can be interpreted as if for each state $s \in \mathcal{S}$ the flows x^s were established from scratch in the currently available capacity $\alpha_e^s y_e, e \in \mathcal{E}$. We note that the resulting GR optimization problem (1a), (1c), (1d) is polynomial also for the full set of admissible paths $\hat{\mathcal{P}}$, as, because of the independence of flow variables $x^s, s \in \mathcal{S}$, it can be easily formulated as a compact linear program using the node-link notation. GR is the most flexible protection mechanism we can think of and therefore it provides a lower bound for the network cost (1a). Yet, for the network considered in Example EC.1 shown in the electronic companion, the optimal FTOP solution happens to be optimal for GR as well.

A natural extension of FTOP that allows for limited tunnel thickening is discussed in Section EC.4 of the electronic companion.

3. Dual problem and dual separation

In this section we will formulate the problem dual to the primal problem $P(\mathcal{P})$ defined by (1) and discuss the separation problem corresponding to the dual polyhedron.

3.1 Dual problem formulation

Problem $D(\mathcal{P})$ dual to $P(\mathcal{P})$ for a given set of admissible paths \mathcal{P} is as follows (for derivation see Pióro and Medhi (2004)):

$$\mathbf{Problem } D(\mathcal{P}): \quad W(\mathcal{P}) = \max \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} h_d^s \lambda_d^s \quad (2a)$$

$$\pi_e^0 + \sum_{s \in \mathcal{S}_e} \alpha_e^s \pi_e^s \leq \xi_e, \quad e \in \mathcal{E} \quad (2b)$$

$$\sum_{s \in \mathcal{S}} \sigma_{dp}^s \leq \sum_{e \in \mathcal{E}} \delta_{edp} \pi_e^0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d \quad (2c)$$

$$\lambda_d^s \leq \sigma_{dp}^s + \sum_{e \in \mathcal{E}_s} \delta_{edp} \pi_e^s, \quad s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}_d \quad (2d)$$

$$\pi, \lambda, \sigma \geq 0 \text{ and continuous.} \quad (2e)$$

Let $\Pi(\mathcal{P})$ denote the dual polyhedron of all feasible solutions (π, λ, σ) of $D(\mathcal{P})$, where $\pi = (\pi_e^0, \pi_e^s, e \in \mathcal{E}, s \in \mathcal{S}_e)$, $\lambda = (\lambda_d^s, d \in \mathcal{D}, s \in \mathcal{S})$, $\sigma = (\sigma_{dp}^s, d \in \mathcal{D}, p \in \mathcal{P}_d, s \in \mathcal{S})$. We treat the dual solutions (π, λ, σ) as vectors in the \mathbb{R}^N space of dimension $N = |\mathcal{E}| + \sum_{e \in \mathcal{E}} |\mathcal{S}_e| + |\mathcal{D}| \cdot |\mathcal{S}| + (\sum_{d \in \mathcal{D}} |\mathcal{P}_d|) \cdot |\mathcal{S}|$, and observe that $\Pi(\mathcal{P})$ is a fully dimensional subset of \mathbb{R}^N . The projection of polyhedron $\Pi(\mathcal{P})$ onto the (π, λ) space \mathbb{R}^n , where $n = |\mathcal{E}| + \sum_{e \in \mathcal{E}} |\mathcal{S}_e| + |\mathcal{D}| \cdot |\mathcal{S}|$, is defined as $\Omega(\mathcal{P}) := \{(\pi, \lambda) \in \mathbb{R}^n : \exists \sigma \in \mathbb{R}^{N-n}, (\pi, \lambda, \sigma) \in \Pi(\mathcal{P})\}$. Polyhedron $\Omega(\mathcal{P})$ is also fully dimensional.

Assume that the linear program $P(\mathcal{P})$, where $\mathcal{P} \subseteq \hat{\mathcal{P}}$, formulated in (1) is feasible. Then, due to the strong duality property (Lasdon 1970, Minoux 1986), the optimal objective $W(\mathcal{P})$ of the corresponding dual problem $D(\mathcal{P})$ formulated in (2) is equal to the optimal primal objective $C(\mathcal{P})$ of $P(\mathcal{P})$, i.e., $W(\mathcal{P}) = C(\mathcal{P})$. Moreover, $\mathcal{P} \subseteq \hat{\mathcal{P}}$ implies that polyhedron $\Pi(\mathcal{P})$ contains polyhedron $\Pi(\hat{\mathcal{P}})$, $\Pi(\mathcal{P}) \supseteq \Pi(\hat{\mathcal{P}})$, so that $W(\mathcal{P}) \geq W(\hat{\mathcal{P}})$ and hence $W(\mathcal{P})$ is actually an upper bound on the optimal objective $C(\hat{\mathcal{P}})$ of $P(\hat{\mathcal{P}})$, i.e., of FTOP.

Certainly, problem $D(\hat{\mathcal{P}})$ represented by formulation (2) is \mathcal{NP} -hard since it is an LP dual of an \mathcal{NP} -hard LP problem $P(\hat{\mathcal{P}})$ represented by formulation (1).

3.2 Dual separation

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

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

Let us first notice that if $(\pi, \lambda) \notin \mathbb{R}_+^n$ then it is separated from Ω trivially, by one of the non-negativity inequalities in (2e). Hence, we can assume that $(\pi, \lambda) \in \mathbb{R}_+^n$. Then, it is easy to check whether π fulfills all constraints in (2b). If it does not, then (π, λ) is separated from Ω by one of the unfulfilled inequalities in (2b), i.e., by one of the hyperplanes $\pi_e^0 + \sum_{s \in \mathcal{S}_e} \alpha_e^s \pi_e^s = \xi_e$.

So now we can assume that $(\pi, \lambda) \in \mathbb{R}_+^n$ and π is feasible with respect to (2b). Under this assumption $(\pi, \lambda) \notin \Omega$ if, and only if, there exists a demand $d \in \mathcal{D}$ and a path $q \in \hat{\mathcal{P}}_d$ such that for any $\sigma_{dq}^s \geq 0, s \in \mathcal{S}$, (π, λ) does not fulfill constraints (2c) or (2d) for path q . Before formulating a condition for such a path q (in Proposition 1), we introduce the following useful definitions.

For a given path $q \in \hat{\mathcal{P}}_d$, the quantity $|q|^0 := \sum_{e \in \mathcal{E}} \delta_{edq} \pi_e^0$ is called the nominal dual length of q , while the quantities $|q|^s := \sum_{e \in \mathcal{E}_s} \delta_{edq} \pi_e^s, s \in \mathcal{S}$, are called the state-dependent dual lengths of path q , where as before, $\delta_{edq}, e \in \mathcal{E}$, are the link-path incidence coefficients characterizing path q . Note that, by definition, $\mathcal{E}_s \cap q = \emptyset$ implies $|q|^s = 0$, and in particular $|q|^s = 0$ for all $q \in \hat{\mathcal{P}}_d$ and $s \in \mathcal{S}_0$.

With these two notions we can rewrite constraints (2c) and (2d) to get the following concise system of inequalities in variables $\sigma_{dq}^s, s \in \mathcal{S}$:

$$\sum_{s \in \mathcal{S}} \sigma_{dq}^s \leq |q|^0 \quad (3a)$$

$$\lambda_d^s \leq \sigma_{dq}^s + |q|^s, \quad s \in \mathcal{S} \quad (3b)$$

$$\sigma_{dq}^s \geq 0, \quad s \in \mathcal{S}. \quad (3c)$$

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

$$\langle q \rangle := |q|^0 + \sum_{s \in \mathcal{S}_q(\pi, \lambda)} (|q|^s - \lambda_d^s), \quad (4)$$

where

$$\mathcal{S}_q(\pi, \lambda) := \{s \in \mathcal{S} : |q|^s < \lambda_d^s\}. \quad (5)$$

Proposition 1 *Consider a given vector $(\pi, \lambda) \in \mathbb{R}_+^n$ with π fulfilling constraints (2b). Then $(\pi, \lambda) \notin \Omega$ if, and only if, there exists a demand $d \in \mathcal{D}$ and a path $q \in \hat{\mathcal{P}}_d$ such that*

$$\langle q \rangle < 0. \quad (6)$$

If this is the case, the inequality reverse to (6), that is,

$$\sum_{e \in \mathcal{E}} \delta_{edq} \pi_e^0 + \sum_{s \in \mathcal{S}_q(\pi, \lambda)} \left(\sum_{e \in \mathcal{E}_s} \delta_{edq} \pi_e^s - \lambda_d^s \right) \geq 0, \quad (7)$$

is not fulfilled by (π, λ) and thus the hyperplane $\sum_{e \in \mathcal{E}} \delta_{edq} \pi_e^0 + \sum_{s \in \mathcal{S}_q(\pi, \lambda)} (\sum_{e \in \mathcal{E}_s} \delta_{edq} \pi_e^s - \lambda_d^s) = 0$ separates (π, λ) from Ω . (The proof is given in Section EC.3.2 of the electronic companion.)

Proposition 1 states that $(\pi, \lambda) \in \Omega$ if, and only if, (i) $(\pi, \lambda) \in \mathbb{R}_+^n$, (ii) π fulfills (2b), and (iii) for each $d \in \mathcal{D}, q \in \hat{\mathcal{P}}_d$, the total dual length of q with respect to (π, λ) is nonnegative, i.e., $\langle q \rangle \geq 0$. The last condition, however, cannot be used as an inequality characterizing polyhedron Ω since its form (7) depends on a particular vector (π, λ) through $\mathcal{S}_q(\pi, \lambda)$. Still, it directly implies such inequalities—the explicit characterization of the projected polyhedron Ω is given in Section EC.3.1 of the electronic companion.

Finally, we observe that by equivalence of optimization and separation in linear programming, see Grotschel et al. (1988), Nemhauser and Wolsey (1988), DSP is \mathcal{NP} -hard since problem $D(\hat{\mathcal{P}})$ specified in (2) is \mathcal{NP} -hard as a problem dual to FTOP which is \mathcal{NP} -hard (already for polynomially bounded number of states $|\mathcal{S}|$, see Section 2.3). This fact is reflected by the difficulty of finding a path q violating inequality (7). We will come back to the complexity issues related to FTOP and DSP in Section 5.

4. Path generation

The link-path LP formulation (1) of FTOP is noncompact because of the presence of exponentially many path-flow variables x (and, as a matter of fact, of exponentially many constraints (1e)) corresponding to all possible elementary paths included in the maximum admissible path-list $\hat{\mathcal{P}}$. The formulation is potentially noncompact also because of possibly exponential number of states in the set \mathcal{S} (since the number of all link availability coefficient vectors is in general exponential, equal to $m^{|\mathcal{E}|}$ when each link has m levels of availability). To avoid such an additional intrinsic hardness of the problem, we assume that $|\mathcal{S}|$ is polynomially bounded by the number of links.

Thus, in practice, formulation (1) of FTOP cannot be solved directly, as we are virtually not able to include all elementary routing paths in the path-lists. In order to consider all paths in $\hat{\mathcal{P}}$, we need to apply path generation (PG), see Ahuja et al. (1993)—a classical technique in multicommodity flow networks related to column generation in linear programming, cf. Lasdon (1970), Minoux (1986). With PG, starting from some initial path-lists \mathcal{P}_d , $d \in \mathcal{D}$, we iteratively generate new paths, one per demand, and add those paths that may improve the solution to the path-lists—in effect, we are adding the corresponding path-flow variables to the problem formulation. As discussed below, this is done by solving an appropriate pricing problem using, as parameters, optimal dual variables, i.e., an optimal solution (λ^*, π^*) of the current problem $D(\mathcal{P})$ dual to $P(\mathcal{P})$.

The application of PG to FTOP is explained below.

4.1 Pricing problem

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

- the point to be separated from polyhedron Ω is not just any vector $(\pi, \lambda) \in \mathbb{R}_+^n$ with π fulfilling (2b), but one of the optimal solutions of the dual problem $D(\mathcal{P})$ for some $\mathcal{P} \subseteq \hat{\mathcal{P}}$
- in PP we need to find, for each $d \in \mathcal{D}$, not just a path (if any) in $\hat{\mathcal{P}}_d$ which separates the current optimal dual solution (π, λ) of $D(\mathcal{P})$, but rather a path $q \in \hat{\mathcal{P}}_d \setminus \mathcal{P}_d$ for which the constraints of Ω corresponding to the given d and q are *most violated* by (π, λ) . In the sequel, the problem of finding such a path, i.e., PP for this setting, will be denoted by $PP(\mathcal{P}, \pi, \lambda, d)$.

In fact, the second property does not necessarily have to be obeyed in PG since any path $q \in \hat{\mathcal{P}}_d$ that violates at least one of the constraints (7) provides a cut that decreases, in a valid way, the current polyhedron $\Omega(\mathcal{P})$. Still, the use of the most violated cuts implied by the second property can substantially speed up the path generation process, especially when finding the most violated cut is not substantially more time consuming than finding an arbitrary cut.

Consider a given dual optimal solution (π, λ) of $D(\mathcal{P})$ and a given path $q \in \hat{\mathcal{P}}_d$ for some $d \in \mathcal{D}$. Let $D(q)$ denote the degree of violation of constraints (7) (for the considered d and q) by vector (π, λ) .

Proposition 2 $D(q) = \max\{-\langle q \rangle, 0\}$. (The proof is given in Section EC.3.3 of the electronic companion.)

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

$$\text{minimize } \langle q \rangle \text{ over } q \in \hat{\mathcal{P}}_d, \quad (8)$$

i.e., to finding the most violated separating inequality (7). Clearly, when the solution $\langle q \rangle$ of (8) is negative, then for a given $d \in \mathcal{D}$ $q \notin \mathcal{P}_d$ and $D(q) = -\langle q \rangle$ is the largest degree of violation of constraints (7) by (π, λ) corresponding to a given $d \in \mathcal{D}$.

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

4.2 Binary formulation of PP

The pricing problem formulated in (8) is difficult because of the particular form of the total dual length $\langle q \rangle$. Nevertheless, the problem can be stated as a binary programming problem by means of formulation (9) given below. In the formulation, binary variables $u_e, e \in \mathcal{E}$, specify the path $q \in \hat{\mathcal{P}}_d$ we are looking for: $q = \{e \in \mathcal{E} : u_e = 1\}$. Binary variables $z^s, s \in \mathcal{S}$, in turn, identify the set $\mathcal{S}_q(\pi, \lambda)$ corresponding to the so defined path q : $\mathcal{S}_q(\pi, \lambda) = \{s \in \mathcal{S} : z^s = 1\}$. Besides, $\delta^+(v)$ and $\delta^-(v)$ denote the sets of all links outgoing from, and all links incoming to node $v \in \mathcal{V}$, respectively. Also, $o(d)$ is the originating node of the considered demand $d \in \mathcal{D}$ and $t(d)$ is its terminating node.

$$\min \quad L = \sum_{e \in \mathcal{E}} \pi_e^0 u_e + \sum_{s \in \mathcal{S}} \left(\sum_{e \in \mathcal{E}} \pi_e^s u_e - \lambda_d^s \right) z^s \quad (9a)$$

$$\sum_{e \in \delta^+(o(d))} u_e - \sum_{e \in \delta^-(o(d))} u_e = 1 \quad (9b)$$

$$\sum_{e \in \delta^+(v)} u_e - \sum_{e \in \delta^-(v)} u_e = 0, \quad v \in \mathcal{V} \setminus \{o(d), t(d)\} \quad (9c)$$

$$u_e \in \{0, 1\}, \quad e \in \mathcal{E}; \quad z^s \in \{0, 1\}, \quad s \in \mathcal{S}. \quad (9d)$$

Constraints (9b) and (9c) define, through variables u_e equal to 1, a routing path q from $o(d)$ to $t(d)$. Note that in any optimal solution $z^s = 0$, when the length, with respect to π^s , of the constructed path q (i.e., $|q^s|$) is greater than λ_d^s , and $z^s = 1$ when $|q^s| < \lambda_d^s$. This is easily seen from the form of the objective function (9a), which in order to be minimized should have all the terms $\sum_{e \in \mathcal{E}} \pi_e^s u_e - \lambda_d^s$ cancelled (through $z_s = 0$) when positive, and maintained (through $z_s = 1$) when negative. Hence, an optimal solution u, z, L of (9) defines an optimal path $q := \{e \in \mathcal{E} : u_e = 1\}$, with $L = |q|^0 + \sum_{s \in \mathcal{S}_q(\pi, \lambda)} (|q^s| - \lambda_d^s)$ equal to its total dual length $\langle q \rangle$.

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

$$L = \sum_{e \in \mathcal{E}} \pi_e^0 u_e + \sum_{e \in \mathcal{E}} \sum_{s \in \mathcal{S}} \pi_e^s Z_e^s - \sum_{s \in \mathcal{S}} \lambda_d^s z^s, \quad (10)$$

and add the constraints

$$Z_e^s \leq u_e, Z_e^s \leq z^s, Z_e^s \geq 0, Z_e^s \geq u_e + z^s - 1, e \in \mathcal{E}, s \in \mathcal{S}. \quad (11)$$

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

4.3 PG algorithm

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

Step 1: Define initial admissible path-lists $\mathcal{P}_d, d \in \mathcal{D}$, and $\mathcal{P} := \bigcup_{d \in \mathcal{D}} \mathcal{P}_d$.

Step 2: Solve the dual problem $D(\mathcal{P})$ given by (2) to obtain optimal dual variables π, λ .

Step 3: For each $d \in \mathcal{D}$ solve $PP(\mathcal{P}, \pi, \lambda, d)$ formulated in (9). If optimal L is negative then add the resulting path q 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 FTOP to optimality. Otherwise, go to Step 2.

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

Assume that the set \mathcal{S}_0 of maximum availability states is not empty. If there exists a path $q \in \hat{\mathcal{P}}_d$ with $|q|^0 < \sum_{s \in \mathcal{S}_0} \lambda_d^s$, then $\langle q \rangle \leq |q|^0 - \sum_{s \in \mathcal{S}_0} \lambda_d^s < 0$ (because $|q|^s = 0$ for all $s \in \mathcal{S}_0$). Hence, for any such path q the degree of violation $D(q)$ is greater than or equal to $\sum_{s \in \mathcal{S}_0} \lambda_d^s - |q|^0 > 0$. Thus, while the pricing problem (finding a path q with minimum $\langle q \rangle$) is in general \mathcal{NP} -hard (see Section 5), finding a path q with minimum $|q|^0$ is obviously polynomial. Hence, we may speed up the PG process: in Step 2 for each $d \in \mathcal{D}$ we find a path $q \in \hat{\mathcal{P}}_d$ with minimum $|q|^0$, and if $|q|^0 < \sum_{s \in \mathcal{S}_0} \lambda_d^s$, then we add path q to the problem and skip solving $PP(\mathcal{P}, \pi, \lambda, d)$ for the current d . Certainly, the overall time efficiency of this modification is in general hard to assess since there is a tradeoff between the value of $D(q)$ (the larger the better) and the time spent in Step 2.

An illustration of the above considerations is given in the electronic companion (Example EC.3).

5. Computational complexity of PP, DSP, and special cases

Although not theoretically guaranteed, the iterative PG process described in Section 4.3 will typically terminate in a polynomial number of steps, as it is closely related to the revised simplex method (Lasdon 1970, Minoux 1986), an exponential algorithm which exhibits this desirable property in practical applications. Indeed, consider a path $q \in \hat{\mathcal{P}}_d \setminus \mathcal{P}_d$ that solves $\text{PP}(\mathcal{P}, \pi, \lambda, d)$ for $d \in \mathcal{D}$. The variables x_{dq}^0 and $x_{dq}^s, s \in \mathcal{S}$, are by assumption non-basic as they are not seen in formulation (1) of $\text{P}(\mathcal{P})$, and the maximum degree of constraint violation $D(q)$ is equal to the sum of the reduced cost of variable x_{dq}^0 and of the reduced costs of those variables among $x_{dq}^s, s \in \mathcal{S}$, that have the negative reduced cost. Thus, the computational complexity of the PG algorithm is in practice determined by the complexity of the pricing problem PP.

Let us consider the special case of state scenarios containing only partial failures of single links. Hence, $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1$, where \mathcal{S}_0 is a set of maximum states (recall that in a maximum state, all links maintain full availability), and $\mathcal{S}_1 := \bigcup_{e \in \mathcal{E}} \mathcal{S}(e)$ is a set of single-link partial availability states. Each set $\mathcal{S}(e)$ represents a set of states in which only link e has the reduced capacity, i.e., $s \in \mathcal{S}(e)$ implies $\alpha_e^s < 1$ and $\alpha_{e'}^s = 1, e' \in \mathcal{E} \setminus \{e\}$. Note that if link e is assumed to be perfectly reliable then the set $\mathcal{S}(e)$ is empty. It turns out that in the considered case the pricing problem is polynomial.

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

$$\langle q \rangle = \sum_{e \in q} \omega(e) - \sum_{s \in \mathcal{S}} \lambda_d^s, \quad (12)$$

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

$$\omega(e) := \pi_e^0 + \sum_{s \in \mathcal{S}(e)} \min\{\pi_e^s, \lambda_d^s\}. \quad (13)$$

(The proof is given in Section EC.5.1 of the electronic companion.)

Since the term $\sum_{s \in \mathcal{S}} \lambda_d^s$ in (12) is constant, solving PP is equivalent to finding a shortest path with respect to the link weights $\omega(e), e \in \mathcal{E}$, e.g., by the Dijkstra algorithm. Hence, PP is polynomial, and, by equivalence of optimization and separation in linear programming (Grotschel et al. 1988, Nemhauser and Wolsey 1988), FTOP is in this case polynomial as well (an appropriate compact LP formulation is given in Section EC.5.2 of the electronic companion).

The same argument shows that DSP (the dual separation problem), and hence the dual to FTOP, are polynomial as well. Moreover, the dual can be solved by the ellipsoidal method of Khachian (1979) (a polynomial algorithm for LP) using the cuts implied by Proposition 1, i.e., by

finding successive paths that separate the current dual solution. The paths found in this way will constitute a polynomial set of paths required to solve FTOP to optimality.

The above weight construction applies to undirected networks as well. Also, the compact FTOP formulation (EC.26) can be easily modified for the undirected links case. Besides, as shown in Section EC.6 of the electronic companion, the single-node failure scenarios can be treated by polynomial algorithms analogously as in the single-link case.

As already mentioned at the end of Section 3.2, the dual separation problem is \mathcal{NP} -hard for the general case of a state scenario with polynomially bounded size. Moreover, DSP is \mathcal{NP} -hard already for partial double-link failures, i.e., when any two links can fail simultaneously. This was demonstrated by Coudert et al. (2007) (see also Orłowski and Pióro (2012)) for a DWSP problem formulation analogous to (1). Thus the pricing problem (8) is most likely \mathcal{NP} -hard in those cases as well, as it is very similar to DSP. Yet, we cannot be sure of that as \mathcal{NP} -hardness of DSP (and, for that matter, \mathcal{NP} -hardness of FTOP) does not necessarily imply \mathcal{NP} -hardness of PP, since formally PP is simpler than DSP (see the discussion at the beginning of Section 4.1).

6. A lower bound formulation

Since FTOP is \mathcal{NP} -hard, it is worthwhile finding a polynomial problem that provides a lower bound for its cost $C(\hat{\mathcal{P}})$. A compact linear program, referred to as LBP, whose solutions can be used as the lower bound of FTOP without the necessity of generating paths is as follows.

$$\min C' = \sum_{e \in \mathcal{E}} \xi_e y_e^0 \quad (14a)$$

$$\sum_{e \in \delta^+(v)} x_{ed}^0 - \sum_{e \in \delta^-(v)} x_{ed}^0 = 0, \quad d \in \mathcal{D}, v \in \mathcal{V} \setminus \{o(d), t(d)\} \quad (14b)$$

$$\sum_{d \in \mathcal{D}} x_{ed}^0 \leq y_e^0, \quad e \in \mathcal{E} \quad (14c)$$

$$\sum_{e \in \delta^+(o(d))} x_{ed}^s - \sum_{e \in \delta^-(o(d))} x_{ed}^s \geq h_d^s, \quad d \in \mathcal{D}, s \in \mathcal{S} \quad (14d)$$

$$\sum_{e \in \delta^+(v)} x_{ed}^s - \sum_{e \in \delta^-(v)} x_{ed}^s = 0, \quad d \in \mathcal{D}, v \in \mathcal{V} \setminus \{o(d), t(d)\}, s \in \mathcal{S} \quad (14e)$$

$$x_{ed}^s \leq x_{ed}^0, \quad e \in \mathcal{E}, d \in \mathcal{D}, s \in \mathcal{S} \quad (14f)$$

$$\sum_{d \in \mathcal{D}} x_{ed}^s \leq \alpha_e^s y_e^0, \quad e \in \mathcal{E}, s \in \mathcal{S}_e \quad (14g)$$

$$\text{all variables } x, y^0 \text{ continuous and nonnegative.} \quad (14h)$$

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

As shown in Section EC.7 of the electronic companion, the solution of formulation (14) is a lower bound for FTOP because all constraints of (14) are obviously fulfilled by the link-flows induced by every feasible solution of (1). It is also shown there that in general the cost of LBP is strictly lower than the cost of FTOP so that a natural question arises: **why not an LBP-based protection strategy (i.e., a kind of link-flow thinning) instead of FT is considered in this paper?** The reason is (see Section EC.7) that (i) the link-flow solution delivered by LBP cannot in general be transformed to a feasible path-flow solution and therefore the simple traffic control mechanism of FT cannot be applied; (ii) as illustrated in Tables EC.2 and EC.3 in the electronic companion, the cost advantage of LBP over FTOP is only marginal; (iii) in general the optimal nominal link-flow x^0 delivered by LBP must contain loops and this might cause problems in traffic routing.

7. Affine versions of FT

As discussed in more detail in Section 9, the practical usefulness of the FT strategy in its form presented so far is limited for three reasons. First, broadcasting of the current link capacity information in real time could be difficult. Next, it would be difficult to solve FTOP (and to store the resulting solution) for an excessive (exponential) number $|\mathcal{S}|$ of states. Finally, even if we managed to achieve an optimal solution (and could store it) for a given (reasonably large) set of states \mathcal{S} , for the valid states outside \mathcal{S} the thinning would be undefined.

Because of that, in this section we will consider a version of FT called *affine flow thinning* (AFT) that addresses the above issues. In particular, we will study a variant of FTOP (called AFTOP) corresponding to AFT. AFTOP relies on using the decision rules largely used in multi-stage robust optimization, and more particularly, affine decision rules (Ben-Tal et al. 2004) (application of affine decision rules to traffic routing were introduced by Ouorou and Vial (2007) and further investigated by Poss and Raack (2013)). In essence, AFTOP turns out to be nothing else but a particular example of an adjustable robust linear program, where the uncertainty set is \mathcal{S} and the adjustable variables are x_{dp}^s . Hence, rather than letting the adjustable variables depend on each specific state s , following Ben-Tal et al. (2004) in AFTOP we let these variables depend *affinely* on the parameters that describe the state: $\alpha_e^s, e \in \mathcal{E}$. We adapt these ideas by considering two special types of FT described in Section 7.1 and Section 7.2 below.

7.1 Affine flow thinning and the related version of FTOP

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

$$x_{dp}^s = A_{dp}(\alpha^s) = z_{dp}^0 + \sum_{e \in \mathcal{E}} z_{dp}^e \alpha_e^s, \quad d \in \mathcal{D}, p \in \mathcal{P}_d. \quad (15)$$

Note that parameters $z_{dp}^0, z_{dp}^e, e \in \mathcal{E}$, of the affine function A_{dp} specified for the given path $p \in \mathcal{P}_d$ do not depend on the state s . The AFT version of FTOP (formulated for FT in (1)) is referred to as AFTOP. In AFTOP, the parameters z become optimization variables and variables x^s play only an auxiliary role. Such a formulation is obtained by adding equations (15) to (1).

Observe also that with appropriately limited cardinality of \mathcal{S} , traffic efficiency of AFT can be very close to traffic efficiency of FT. In fact, as shown in Poss and Raack (2013) for a similar problem, AFT becomes equivalent to FT if, and only if, \mathcal{S} contains at most $|\mathcal{E}| + 1$ states. In Section 8.2 we will compare traffic efficiency of AFT and FT for state scenarios with $|\mathcal{S}| > |\mathcal{E}| + 1$.

In the implementation aspect, AFT is more practical than FT. Although as FT, AFT must be aware of the current link states α^s , it will use the (optimized) affine functions $A_{dp}, d \in \mathcal{D}, p \in \mathcal{P}_d$, for thinning the nominal flows. Such thinning, contrary to FT, is applicable to all possible states, i.e., also to the states not in the set \mathcal{S} assumed for optimization. Traffic efficiency of such extended thinning is addressed in Section 8.2.

In spite of the above advantages, AFT suffers from two problems. First, introduction of affine coefficients (variables) $z_{dp}^0, z_{dp}^e, e \in \mathcal{E}$, for each path $p \in \mathcal{P}_d, d \in \mathcal{D}$, and each link $e \in \mathcal{E}$, leads to large linear programs for AFTOP, both when included in formulation (1) or when using the dualized formulation presented in Section 7.4. Second, AFT can still be hard to implement because the affine function of each path p involves, as arguments, the link availability coefficients of all links. Thus, disseminating the link-status information is as difficult as for FT.

7.2 Restricted affine flow thinning

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

$$x_{dp}^s = A_{dp}(\alpha^s(p)) = z_{dp}^0 + \sum_{e \in p} z_{dp}^e \alpha_e^s, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, \quad (16)$$

where $\alpha^s(p) = (\alpha_e^s, e \in p)$. Clearly, RAFT involves less variables z than AFT (an advantage), but it is not necessarily as efficient as FT, in particular for sets \mathcal{S} with $|\mathcal{S}| \leq |\mathcal{E}| + 1$ (a disadvantage). Traffic efficiency of RAFT, AFT and FT is compared in Section 8.2.

The most important advantage of RAFT over AFT (and for that matter, over FT as well) is that with RAFT disseminating link capacity information is, as discussed in Section 9.2, not a major issue, as a path-originating node thins the corresponding path-flow only on the basis of the current capacities of the links along the path. In fact, we could easily use even more information on the link availability states, namely the thinning function of a path-flow could be made dependent on

coefficients α_e^s of all the links e belonging to the sets $\delta(v)$ for all nodes v traversed by the considered path, i.e.,

$$x_{dp}^s = A_{dp}(\underline{\alpha}^s(p)) = z_{dp}^0 + \sum_{e \in \mathcal{E}'(p)} z_{dp}^e \alpha_e^s, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, \quad (17)$$

where $\underline{\alpha}^s(p) = (\alpha_e^s, e \in \mathcal{E}'(p))$, $\mathcal{E}'(p) := \bigcup_{v \in \mathcal{V}(p)} \delta(v)$, and $\mathcal{V}(p)$ is the set of all nodes visited by path $p \in \mathcal{P}_d$. However, we will not further study this opportunity, leaving it for future investigations.

7.3 Uncertainty polytope

From the computational efficiency viewpoint, the main interest in AFT and RAFT stems from their capability of handling large sets \mathcal{S} (referred to as *uncertainty sets* in this section) of link availability states, provided \mathcal{S} is given as the set of vertices (extreme points) of a polytope described by a polynomial number of variables and constraints.

The objective of this section is to describe a polytope, referred to as *FT uncertainty polytope*, that is a natural generalization of uncertainty sets \mathcal{S} containing single and double-link partial availability states considered previously. We point out that the polytope must describe the values taken by two sets of parameters playing, in a sense, opposite roles in the optimization problems: α_e^s and h_d^s . While the values of α_e^s describe the impact of disadvantageous events on the network, a positive difference between h_d^0 and h_d^s can be interpreted as relaxation of the demand constraints.

The presented considerations are in the line of the framework of *globalized robust counterparts* studied in Ben-Tal et al. (2006). Our approach differs in the way the set of all *physically possible* scenarios is defined. While Ben-Tal et al. (2006) express the latter set as the sum of a bounded set and a cone, we introduce in the following an uncertainty polytope that can translate the respective roles of α_e^s and h_d^s . As explained in the Section EC.8.1, the polytope is an interesting contribution in itself, as it can provide an alternative to Ben-Tal et al. (2006) for optimization problems admitting relaxation of some of the problem constraints when the uncertainty is profound.

The FT uncertainty polytope, denoted by $S(\mathcal{N})$, is a variant of the budgeted uncertainty polytope introduced in Bertsimas and Sim (2004), a well-known object in robust network design and other linear problems assuming uncertainty. The polytope is a subset of the $\mathbb{R}_+^{|\mathcal{E}|+|\mathcal{D}|}$ space, and its points (vectors) are denoted with (α, h) where $\alpha = (\alpha_e, e \in \mathcal{E})$ and $h = (h_d, d \in \mathcal{D})$. The definition of $S(\mathcal{N})$ assumes a given finite set $\mathcal{A} := \{a^1, a^2, \dots, a^K\}$ of link availability levels, and a given set $\mathcal{N} := \{N^1, N^2, \dots, N^K\}$ of positive integers with the following properties:

- $0 \leq a^1 < a^2 < \dots < a^K = 1$
- $1 \leq N^k \leq |\mathcal{E}|$, $k \in \mathcal{K}$ (where $\mathcal{K} := \{1, 2, \dots, K\}$), $\sum_{k=1}^K N^k \geq |\mathcal{E}|$.

Also, in the polytope the demand volumes and link availability coefficients are dependent, and are related through a given sequence of fractional numbers b^k , $k \in \mathcal{K}$, fulfilling the conditions:

- $1 > b^1 \geq b^2 \dots \geq b^K = 0$
- $\sum_{k=1}^{k_0} b^k N^k + b^{k_0+1} (|\mathcal{E}| - \sum_{k=1}^{k_0} N^k) \leq 1$, where k_0 is the smallest index in \mathcal{K} for which $\sum_{k=1}^{k_0} N^k < |\mathcal{E}|$ and $\sum_{k=1}^{k_0+1} N^k > |\mathcal{E}|$, or $\sum_{k=1}^{k_0} N^k = |\mathcal{E}|$.

To simplify the description of the notion of uncertainty polytope, we first introduce an auxiliary polytope $S'(\mathcal{N})$ —the polytope in $\mathbb{R}^{|\mathcal{E}||\mathcal{K}|}$ defined as the set of all vectors $u = (u_e^k, e \in \mathcal{E}, k \in \mathcal{K})$ fulfilling the conditions

$$\sum_{k \in \mathcal{K}} u_e^k = 1, \quad e \in \mathcal{E} \quad (18a)$$

$$\sum_{e \in \mathcal{E}} u_e^k \leq N^k, \quad k \in \mathcal{K} \quad (18b)$$

$$u \geq 0. \quad (18c)$$

Below, the quantity $\sum_{e \in \mathcal{E}} u_e^k$ will be denoted by U^k . Note that (18a) implies that $\sum_{k \in \mathcal{K}} U^k = |\mathcal{E}|$.

The uncertainty polytope $\mathcal{S}(\mathcal{N})$ is the polytope in the $\mathbb{R}^{|\mathcal{E}|+|\mathcal{D}|}$ space induced by $S'(\mathcal{N})$

$$\mathcal{S}(\mathcal{N}) = \{(\alpha(u), h(u)) : u \in S'(\mathcal{N})\}, \quad (19)$$

through the following (linear) mappings:

$$\alpha(u)_e = \sum_{k \in \mathcal{K}} a^k u_e^k, \quad e \in \mathcal{E} \quad (20a)$$

$$h(u)_d = h_d^0 (1 - \sum_{k \in \mathcal{K}} b^k U^k), \quad d \in \mathcal{D}. \quad (20b)$$

Note that for a given link $e \in \mathcal{E}$, if $u_e^k = 1$ then $\alpha(u)_e = a^k$, and for a binary vector $u \in S'(\mathcal{N})$, each quantity U^k ($k \in \mathcal{K}$) denotes the number of links in $\alpha(u)$ at failure level a^k . Also, equality (20b) implies that for each point $(\alpha, h) \in \mathcal{S}(\mathcal{N})$, the demand volumes $h_d^0, d \in \mathcal{D}$, are thinned uniformly, according to $(1 - \sum_{k \in \mathcal{K}} b^k U^k)$ —a measure of the ratio of the amount of unavailable capacity.

In Section EC.8.2 of the electronic companion it is discussed how the uncertainty sets describing n -link failures can be obtained as special cases of $\mathcal{S}(\mathcal{N})$.

7.4 A compact reformulation of AFTOP with respect to \mathcal{S}

In this section we reformulate the problem statement given in (1) (modified for AFTOP in the way described in Section 7.1) using FT uncertainty polytopes $\mathcal{S}(\mathcal{N})$ instead of finite uncertainty sets \mathcal{S} . The reformulation is valid for AFT and RAFT, and is given in Proposition 4. (To get a uniform formulation for both cases, below we use the notation $\mathcal{E}_{dp} := \mathcal{E}$ for AFT, and $\mathcal{E}_{dp} := \mathcal{P}$ for RAFT ($d \in \mathcal{D}, p \in \mathcal{P}_d$).)

Proposition 4 Consider the problem formulation (1) under the affine decision rules

$$x_{dp}^s = z_{dp}^0 + \sum_{\bar{e} \in \mathcal{E}_{dp}} z_{dp}^{\bar{e}} \alpha_{\bar{e}}^s, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, \quad (21)$$

and the uncertainty set described by polytope $\mathcal{S}(\mathcal{N})$. Suppose that $z_{dp}^e = 0$ for each $e \in \mathcal{E} \setminus \mathcal{E}_{dp}$. Then, (1) can be reformulated as follows:

$$\textbf{Problem } A(\mathcal{P}): \quad C(\mathcal{P}) = \min \sum_{e \in \mathcal{E}} \xi_e y_e^0 \quad (22a)$$

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} x_{dp}^0 \leq y_e^0, \quad e \in \mathcal{E} \quad (22b)$$

$$\sum_{\bar{e} \in \mathcal{E}} \tau_{\bar{e}d}^1 + \sum_{k \in \mathcal{K}} N^k \theta_{kd}^1 + h_d^0 \leq \sum_{p \in \mathcal{P}_d} z_{dp}^0, \quad d \in \mathcal{D} \quad (22c)$$

$$\tau_{\bar{e}d}^1 + \theta_{kd}^1 \geq -a^k \sum_{p \in \mathcal{P}_d} z_{dp}^{\bar{e}} - h_d^0 b^k, \quad \bar{e} \in \mathcal{E}, k \in \mathcal{K}, d \in \mathcal{D} \quad (22d)$$

$$\sum_{\bar{e} \in \mathcal{E}} \tau_{\bar{e}e}^2 + \sum_{k \in \mathcal{K}} N^k \theta_{ke}^2 \leq - \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} z_{dp}^0, \quad e \in \mathcal{E} \quad (22e)$$

$$\tau_{\bar{e}e}^2 + \theta_{ke}^2 \geq a^k \left(\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} z_{dp}^{\bar{e}} - \Delta(\bar{e}, e) y_e^0 \right), \quad \bar{e} \in \mathcal{E}, k \in \mathcal{K}, e \in \mathcal{E} \quad (22f)$$

$$\sum_{\bar{e} \in \mathcal{E}} \tau_{\bar{e}dp}^3 + \sum_{k \in \mathcal{K}} N^k \theta_{kdp}^3 \leq x_{dp}^0 - z_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d \quad (22g)$$

$$\tau_{\bar{e}dp}^3 + \theta_{kdp}^3 \geq a^k z_{dp}^{\bar{e}}, \quad \bar{e} \in \mathcal{E}, k \in \mathcal{K}, d \in \mathcal{D}, p \in \mathcal{P}_d \quad (22h)$$

$$\sum_{\bar{e} \in \mathcal{E}} \tau_{\bar{e}dp}^4 + \sum_{k \in \mathcal{K}} N^k \theta_{kdp}^4 \leq z_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d \quad (22i)$$

$$\tau_{\bar{e}dp}^4 + \theta_{kdp}^4 \geq -a^k z_{dp}^{\bar{e}}, \quad \bar{e} \in \mathcal{E}, k \in \mathcal{K}, d \in \mathcal{D}, p \in \mathcal{P}_d \quad (22j)$$

$$x^0, y^0, \theta^1, \theta^2, \theta^3, \theta^4 \text{ cont. and nonnegative, } z, \tau^1, \tau^2, \tau^3, \tau^4 \text{ cont.} \quad (22k)$$

Above, $\Delta(\bar{e}, e) = 1$ if $e = \bar{e}$, and 0 otherwise.

The proof, based on duality, is given in Section EC.8.3 of the electronic companion. It applies the classical dual approach of robust optimization (see for example Ben-Tal et al. (2009)), using uncertainty polytopes $\mathcal{S}(\mathcal{N})$ instead of finite uncertainty sets \mathcal{S} to reformulate (1) for AFT and RAFT.

The above formulation of AFTOP can be extended to account for an additional finite set of states \mathcal{S} on top of $\mathcal{S}(\mathcal{N})$ —see Section EC.8.3 of the electronic companion.

8. Efficiency of flow thinning

In this section we present the results of our numerical study carried out for three realistic undirected network examples taken from the library of network instances SNDlib (Orlowski et al. 2010): *pdh*, *polska*, and *nobel-us*. The networks have 11/12/14 nodes, 34/18/21 undirected links, and 24/66/91

demands, respectively. The demand matrices for those networks specified in SNDlib are taken as the nominal demand volumes in our models.

The purpose of our computational experiments is twofold. First, we assess the computational difficulty of FTOP. Hence, in Section 8.1 we present the results of two sets of experiments which evaluate the computational efficiency of the path generation algorithm, and the computational efficiency of solving AFTOP—the affine version of FTOP, respectively. The second objective of the experiments is to demonstrate the traffic efficiency of flow thinning and its affine versions. Our results from Section 8.2 illustrate the efficiency in question by comparing the corresponding minimum capacity allocation costs.

In the sequel we will consider undirected networks, and thus the optimization problems are treated in their versions for undirected network graphs. The study was carried out on a computer equipped with an Intel(R) Xeon(R) CPU E5-2670 @ 2.60GHz processor, and 132 GB of RAM, using CPLEX 12.5 (CPLEX 2013).

8.1 Computational efficiency of flow thinning optimization

8.1.1 Path generation Recall that PG is the path generation algorithm for FTOP (i.e., problem (1) assuming the full set of admissible paths), based on the binary pricing problem (PP) formulation (9) (see Section 4.3), and that $\text{FTOP}^<$ denotes instances of (1) with limited sets of admissible paths.

Below we report results illustrating the computational efficiency of solving FTOP through PG, and of solving $\text{FTOP}^<$ through a direct use of CPLEX for a predefined set of admissible paths. The PG algorithm was initialized by putting only one shortest (with respect to the unit capacity costs $\xi_e, e \in \mathcal{E}$) path on the path-list of each demand. (Note that such minimal path-lists determine an optimal solution for FTOP involving only the nominal state, i.e., for FTOP with $\mathcal{S} = \{s_0\}$.) For $\text{FTOP}^<$, the predefined sets of admissible paths contain all elementary paths having up to 5 links.

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

Tables 1 and 2 present results for the three networks under the single-link failure scenario (SL) and the double-link failure scenario (DL). SL contains the nominal state plus one failure state for each link (with the assumed α for the failing link). DL contains SL plus one failure state for each pair of links (with the assumed α for both failing links). Each type of scenario is considered for the four pairs (α, β) specified by all four combinations of $\alpha \in \{0.25, 0.5\}$ and $\beta \in \{0.75, 1\}$.

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

- $t_{FTOP<}$ – solving FTOP for the predefined set of admissible paths
- t_{GP} – generating the predefined set of admissible paths ($t_{FTOP<}$ includes t_{GP})
- t_{FTOP} – solving FTOP through PG
- t_{PP} – solving the pricing problems (9) within the PG algorithm (t_{FTOP} includes t_{PP}).

In addition, the quantity in brackets in column “network” gives the number of predefined paths (generated in time t_{GP}), column “#iter.” gives the number of iterations of the PG algorithm, and column “#gen.paths” gives the number of paths generated by PG. We set the running time limit to 3 hours; the cases when the timeout had to be applied (this happened only for FTOP<) are denoted by *.

Table 1 Effectiveness of the path generation algorithm for SL.

network	α	β	$t_{FTOP<}$ (seconds)	t_{GP} (seconds)	t_{FTOP} (seconds)	t_{PP} (seconds)	#iter.	#gen.paths
<i>pdh</i> (6 639 paths)	0.5	0.75	467	308	28	22	8	298
	0.5	1	460	299	46	35	11	335
	0.25	0.75	478	311	43	34	11	198
	0.25	1	293	473	52	43	12	316
<i>polska</i> (491 paths)	0.5	0.75	533	5	16	13	5	412
	0.5	1	560	6	28	22	8	466
	0.25	0.75	383	4	23	19	7	471
	0.25	1	2451	2	27	20	5	506
<i>nobel-us</i> (609 paths)	0.5	0.75	4	2	29	23	6	530
	0.5	1	4	2	43	35	8	591
	0.25	0.75	4	2	30	24	6	560
	0.25	1	4	2	40	33	7	621

Table 2 Effectiveness of the path generation algorithm for DL.

network	α	β	$t_{FTOP<}$ (seconds)	t_{GP} (seconds)	t_{FTOP} (seconds)	t_{PP} (seconds)	#iter.	#gen.paths
<i>pdh</i> (6 639 paths)	0.5	0.75	*	*	2914	2506	13	354
	0.5	1	*	*	3612	3034	14	391
	0.25	0.75	*	*	1882	1524	10	354
	0.25	1	*	*	2653	2202	10	354
<i>polska</i> (491 paths)	0.5	0.75	539	54	345	286	8	466
	0.5	1	667	53	387	317	8	486
	0.25	0.75	425	55	858	746	8	511
	0.25	1	303	61	1193	1038	10	530
<i>nobel-us</i> (609 paths)	0.5	0.75	1798	162	755	649	8	542
	0.5	1	2804	140	2187	1925	10	572
	0.25	0.75	1999	160	1719	1444	9	615
	0.25	1	2741	137	2494	2020	11	646

Tables 1 and 2 show that the relative efficiency of FTOP< and of FTOP is instance-dependent. Still, FTOP (i.e., the PG algorithm), despite the necessity of solving the binary PP subproblems,

is faster than FTOP[<] (predefining the admissible set of paths plus application of the linear solver) for 18 out of 24 instances tested, including 4 instances (*pdh* for DL) for which FTOP could not solve the problem within the time limit. The results also show that a significant fraction of the PG algorithm execution time is spent in the pricing problem.

An important aspect of the PG algorithm, illustrated in Figure EC.5 for *pdh* and *polska* (see Section EC.9 in the electronic companion) is the number of added paths and the network cost for the consecutive iterations. Practically, for all the considered cases the final optimum is obtained after only several (5-7) PG iterations.

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

- FT (flow thinning): problem (1), i.e., FTOP[<]
- AFT (affine flow thinning): problem (1) plus equation (15)
- RAFT (restricted affine flow thinning): problem (1) plus eq. (16)
- GR (global rerouting): problem (1) minus constraints (1b) and (1e) (see Section 2.3)
- AGR (affine GR): problem (1) minus constraint (1b) and (1e) plus eq. (15)
- RAGR (restricted affine GR): problem (1) minus constraint (1b) and (1e) plus eq. (16).

The results for affine flow thinning (AFT, RAFT) and affine flow thinning/thickening (AGR, RAGR) presented in this section (running times) and in Section 8.2 (network costs) are obtained by solving the compact reformulation (22) stated in Proposition 4.

We consider three kinds of state scenarios: single-link (SL), double-link (DL), and triple-link (TL). Scenarios SL and DL are described in Section 8.1.1 (recall that SL consists of the nominal state and failures of single links, and DL contains SL plus failures of pairs of links) while scenario TL consist of DL and failures of triples of links. In the particular scenarios considered below, single failures consist of one occurrence of each link with $\alpha = 0.5$. Double failures encompass 2 occurrences of each pair of links, corresponding to 2 different assignments of $\alpha = 0.5$ or $\alpha = 0.75$ (availability coefficients assigned to the links of a pair are different). Finally, triple failures contain all triples of links, and each triple occurs 6 times, which corresponds to 6 possible assignments of 3 different availability coefficients 0.5, 0.75, 0.9 to the links in a triple. For each of these settings, we consider two cases. In the first case the traffic volume carried in each state must be equal to the nominal volume, i.e., $\beta = 1$ for all demands in all states. Note that this means that for the

Table 3 Solution times in seconds for different routing variants.

scenario	network	GR	FT	AGR	AFT	RAGR	RAFT
SL	<i>pdh</i>	5	100	440	2972	386	594
	<i>polska</i>	0	1	4	6	1	1
	<i>nobel-us</i>	1	2	7	9	4	5
DL	<i>pdh</i>	2262	*	1519	9802	704	1279
	<i>polska</i>	6	269	9	13	1	2
	<i>nobel-us</i>	12	2367	17	25	7	11
TL	<i>pdh</i>	*	*	3664	11942	693	1774
	<i>polska</i>	389	*	16	22	5	6
	<i>nobel-us</i>	281	*	37	67	10	17

uncertainty polytope we assume $b^k = 0, k \in \mathcal{K}$, see (20b). In the second case all demand coefficients are equal to 0.9 for SL, to 0.8 for DL, and to 0.7 for TL. This means that $b^k = 0.1, k \in \mathcal{K}$.

Table 3 reports the running times in seconds, taking the averages of the times corresponding to the two cases of β coefficients described above. We set the time limit to 10 hours—the cases when the timeout occurred are denoted by *. Table 3 reveals that the running times for the general flow thinning procedure (FT) are highly impacted by the number of the simultaneously failing links. In particular, all instances assuming TL could not be solved to optimality within the time limit. On the contrary, affine flow thinning (AFT), and even to a larger extent restricted affine flow thinning (RAFT), can easily cope with the increased number of simultaneous link failures. Also, increase in the running time when passing from SL to DL, and to TL, is moderate.

8.2 Cost efficiency of flow thinning

Tables 4 and 5 compare the optimal cost $C = \sum_{e \in \mathcal{E}} \xi_e y_e^0$ for *polska* and *nobel-us* for the six protection variants and the six state scenarios described in Section 8.1.2. Let the quantity C_{GR} denote the optimal cost for GR (recall that GR implies the cheapest networks and is thus taken as the reference value), while the cost increase with respect to GR (in percents) for each of the remaining variants is given by $f_M := \frac{C_M - C_{GR}}{C_{GR}} 100$, where C_M is the cost for all remaining protection strategies, i.e., M represents any of $FT, AGR, AFT, RAGR, RAFT$. The cases (for FT) that experienced timeout are denoted by *. The two cases of b^k correspond to two values 0.0 and 0.1 in column b .

The results show that assuming FT instead of GR increases the cost by up to 14 %. Imposing the affine version of FT further increases the cost, up to 24 % for AFT, and up to 35 % for RAFT. These are quite reasonable values. Note that for SL, $f_{AGR} = f_{GR}$ (by definition $f_{GR} = 0$), and $f_{AFT} = f_{FT}$, confirming numerically the proven optimality of the affine protection when the uncertainty set has $|\mathcal{E}| + 1$ extreme points (Poss and Raack 2013). Note also that some instances could not be solved to optimality due to the time limit, but only for FT. It happened that in all cases reported in Tables 4 and 5 the value of f_{RAGR} is always equal to f_{RAFT} . However, we constructed examples (not reported here) showing that this is not the case in general.

Table 4 Optimal network costs for *polska*.

scen.	b	f_{FT}	f_{AGR}	f_{AFT}	f_{RAGR}	f_{RAFT}
SL	0.0	12	0	12	12	12
	0.1	14	0	14	14	14
DL	0.0	6	0	16	25	25
	0.1	14	1	22	26	26
TL	0.0	*	3	20	35	35
	0.1	*	1	24	30	30

Table 5 Optimal network costs for *nobel-us*.

scen.	b	f_{FT}	f_{AGR}	f_{AFT}	f_{RAGR}	f_{RAFT}
SL	0.0	14	0	14	14	14
	0.1	16	0	16	16	16
DL	0.0	9	0	17	26	26
	0.1	15	1	22	27	27
TL	0.0	*	3	20	35	35
	0.1	*	2	24	30	30

Table 6 Average percentages of unsatisfied demand.

network	h	AFT		RAFT	
		DL	TL	DL	TL
<i>polska</i>	h^0	5.6	7.8	5.6	7.8
	βh^0	2.3	1.2	2.2	1.2
<i>nobel-us</i>	h^0	4.5	6.2	4.5	6.3
	βh^0	1.9	0.8	1.9	0.8

Table 6 illustrates efficiency of AFT and RAFT optimized for SL, when facing DL or TL. This is an important aspect since once AFT or RAFT is implemented, the thinning formula (15) or (16), respectively, can as well be used for any current α observed on-line in the network. Table 6 reports the average amount, in percent, of the resulting unsatisfied demand divided by the required demand, where the average is taken over the failure states in DL and TL, respectively.

To avoid negative flows, the percentages in Table 6 are computed by defining the flow x_{dp}^s for each path $p \in \mathcal{P}_d, d \in \mathcal{D}$, and each failure state $s \in \mathcal{S}$, in DL or TL, as the maximum of 0 and the flow value specified by (15) or (16), respectively. Note that an analogous modification would be necessary if x_{dp}^s became greater than x_{dp}^0 —this, however, did not happen. In order to compute the violation of the demand constraints for DL and TL, we insert the so obtained values of x_{dp}^s optimized for SL to formulae (1c). We note that in the considered setting, AFT and RAFT cope very well with the unforeseen states. Analogous calculations show that capacity constraints (1d) are never violated. We see that the average amount of the violated demand volume is less for TL than for DL when the demand is reduced by coefficient β . This is because the states with triple simultaneous link failures assume demand coefficients $\beta = 0.7$ while the states with double simultaneous link failures assume $\beta = 0.8$.

9. Applications, implementation and practicality of FTOP

9.1 Applications

FT is applicable to network settings where partial availability of links is typical and the logical tunnel layer can be established. An important example are WMN (wireless mesh networks, i.e., IP/MPLS over wireless networks). WMNs are composed of fixed IP routers and Internet gateways,

interconnected by wireless links based on the radio, microwave, or free space optical (FSO) technologies. WMNs are becoming an important solution for providing broadband Internet access for fixed and mobile users connected to the routers. The reason is that in many circumstances WMNs are cheaper, faster and simpler to deploy, maintain and operate than their wired counterparts (Mogre et al. 2007). This is for example the case in areas lacking network infrastructure where WiFi radio based WMNs can provide Internet access to entire communities despite the fact that only few nodes (gateways) have direct access to the Internet. Such WiFi networks employing fixed mesh routers are also considered for the smart city applications. Another example is a corporate WMN deployed in a large city using FSO links, connecting pairs of transceivers in the line-of-sight installed on the roofs of buildings. In WMN (and, for that matter, in wireless communications in general) a typical link availability state is characterized by a subset of those links that lose a fraction of their maximum capacity (achieved under perfect signal propagation) because of the current weather state that directly affects the radio channel condition. The links adapt to the detected channel condition by switching to an adequate modulation and coding scheme (Goldsmith 2005) and the current capacity is set by the scheme used. Then the IP traffic entering the end-to-end MPLS tunnels can be controlled at the tunnel-originating nodes to fit their current virtual capacity (i.e., nominal or thinned) determined according to the FT strategy.

Another example is an IP/MPLS over DWDM (Dense Wavelength Division Multiplexing) network. In the IP/MPLS layer a typical link failure state is composed of several IP links that lose a fraction of their capacity because of the effect of upwards propagation of the optical cable failure (cut) in the DWDM optical transport network layer (Pióro and Medhi 2004). An IP link loses only a part of its capacity when it is realized over diverse paths in the DWDM layer.

9.2 Implementation issues

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

This issue can be overcome by applying RAFT (see Section 7.2). With RAFT, the thinning factors of each tunnel are specified by means of an affine function depending only on the availability

coefficients of the links along the tunnel. Then the link-status message concerning a change of the availability coefficient of a particular link is sent by its originating node backwards along all the tunnels that traverse this node and use the link in question. In this way, by means of appropriately extended path-error messages in Resource Reservation Protocol – Traffic Engineering (RSVP-TE, the protocol from the TCP/IP stack that is used to establish, maintain, and erase LSPs, see Minei and Lucek (2011) and Awduche et al. (2001)), the nodes can be quickly informed about the changes in the links capacity they need to be aware of. Hence, the crucial mechanism for FT could be implemented in a network composed of IP routers interconnected by IP links with logical tunnels (LSP) realized by means of the MPLS mechanism (see Minei and Lucek (2011) and Andersson and Swallow (2003)).

Another issue is that with FT in general only a fraction of all possible link availability states can be directly considered in optimization, simply because the number of states can be exponential. Hence, the thinning factors for a non-considered state must be somehow calculated at the tunnel originating nodes, having in mind that the resulting tunnel capacity cannot overload the links and the reduction of traffic is acceptable. For the general version of FT, finding a reasonable algorithm for that would be difficult but this task becomes tractable with RAFT (and, for that matter, also for AFT): for the non-considered states, the thinning factor for a given tunnel is automatically determined by the affine function once availability coefficients for its links are known. Moreover, the affine functions, even when optimized using a limited number of states, would tend to ensure reasonable thinning factors also for the states not explicitly considered in optimization, as illustrated in Table 6 in Section 8.2. This is particularly sound when thinning is based on a special case of an affine function: $x_{dp}^s = \max\{0, x_{dp}^0 - \sum_{e \in p} z_{dp}^e (1 - \alpha_e^s)\}$, $d \in \mathcal{D}$, $p \in \mathcal{P}_d$, where all z_{dp} are nonnegative.

9.3 Practicality of FTOP and its affine version related to state description

In our optimization model each link availability coefficient α (for a given link and a given state) is a fixed parameter while the nominal capacity y of the link is an optimization variable. Using the availability coefficients to determine the state-dependent link capacity as $\alpha \cdot y$ is a sound assumption for the following reason. In reality, in most cases link capacity is modular, i.e., equal to $M \cdot y$ where y is a number (an integer variable) of modules of size M installed on the considered link. Referring to the application examples discussed in Section 9.1, in the IP/MPLS over wireless case, where the modules correspond to parallel microwave radio links or wireless optical links, coefficient α corresponds to the modulation and coding scheme applied in the considered state. In the IP/MPLS over DWDM case, α is the percentage of modules (for example, 0%, 33%, 66%, 100%) that are not lost in the considered link as a result of the failure in the DWDM layer. That is, we assume the

equal split of y modules to be realized over a given set of k disjoint DWDM paths ($k = 3$ paths in our example), whatever the value of y (provided y is divisible by k , to be precise). Note that in the paper link capacity is linearized and variables y are made continuous to avoid unnecessary complication of the optimization model.

The state description assumed for FTOP is on purpose quite general. The implementation of FT described in Section 9.2, however, assumes that the demand reduction coefficients depend only on the vector α^s (i.e., $\beta^s = \beta(\alpha^s)$, $s \in \mathcal{S}$), so that the state is in fact determined by the current vector α^s of link availability coefficients. This kind of state description, together with the assumption that the demands will accept reduction in their nominal traffic volumes in the states with significant link capacity reduction, opens a way to specify reasonable demand reduction coefficients by the network operator to avoid substantial network over-dimensioning, and hence excessive network cost. Clearly, the degree to which the preferable traffic volume is decreased should reflect the fractions of the nominal link capacity available in a given state.

A simple way for achieving reasonable reductions is to divide the set of states into several classes corresponding to several degrees of lost capacity, and to assume a certain percentage of the nominal traffic that should be carried for each demand in a given state. For example, we may distinguish two classes of states, one of “almost nominal” states with the entire (100%) nominal traffic to be carried, and the second class of “degraded states” that allow for carrying only 75%, say, of nominal traffic for each demand. A more sophisticated way, applicable in the affine version of FTOP (i.e., in AFTOP), is to relate the demand reduction coefficients to the link availability coefficients through the so called uncertainty polytope (see Section 7.3).

Another important practical issue is how to select a representative set of states of reasonable (tractable) size to be assumed for FTOP (or AFTOP). These states should, on one hand, represent a whole range of typical states, and, on the other hand, be sufficient to effectively approximate the thinning factors in the remaining feasible states (see Section 9.2). Such a selection is network dependent. For the IP/DWDM case the selection is basically simple. As it is a common practice to consider only single optical cable cuts (simultaneous cuts of more than one cable are very unlikely), the resulting multiple partial IP link failures are easy to list and include in FTOP. In the case of WMN, the situation is more complicated. For modelling link availability states implied by weather conditions in a metropolitan WMN, we could divide the nodes into disjoint groups and assume that the bad weather conditions affects either all of them or a subset of them at a time. If we distinguish only several groups then we are able to consider all the states corresponding to all subsets of the family of these groups, i.e., the partial failures (with different levels of link availability, depending for example on the rain intensity) of all nodes in the affected groups. The demand reduction

coefficients would then depend on the total number of affected nodes and (demand-wise) on the position of the demand’s end nodes. This issue, however, requires a separate study.

For everyday network operation a pure flow allocation version of FTOP (or AFTOP) becomes important. Such a problem (let us call it FTOP/FA) arises when nominal link capacities are fixed, i.e., y^0 become given parameters. FTOP/FA has to be considered whenever the nominal traffic matrix (assumed for FTOP) is significantly perturbed causing the nominal flows and the thinning factors (tunnel flow-defining affine functions in the case of AFT) not fit the new traffic matrix anymore. Hence, the nominal flows and thinning factors have to be re-optimized. In order to properly accommodate the new traffic by the network, some kind of fairness criteria (link max-min fairness or proportional fairness (see Chapter 8 in Pióro and Medhi (2004))) should be imposed on the values of $h_d^s, d \in \mathcal{D}, s \in \mathcal{S}$ (these quantities would become optimization variables on top of the tunnel flows), and reflected by an appropriate objective function used instead of (1a).

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

9.4 Why flow thickening should be avoided

Flow thickening (see Section EC.4 in the electronic companion) leads to cheaper solutions as compared to pure thinning, and when all $t_d \rightarrow \infty$ the resulting modification of FTOP becomes equivalent to the GR optimization problem. As the corresponding modification of FT allowing for thickening can be implemented in the same way as the original FT (including the affine versions, see Section 9.2), and thickening can improve traffic performance of FT, a natural question arises why we still opt for FT. The reasons are as follows. First, with FT the maximum number of flows that are adjusted when moving from state s_1 to state s_2 is equal to the number of flows that traverse links affected in state s_1 plus the number flows that traverse links affected in state s_2 . This number is quite small as compared to the number of all (non-zero) nominal flows. However, with thickening this bound is not valid and the number of adjusted flows can become significant. In general, the larger the thickening bound the more tunnel flows are adjusted, and this number is typically the largest for GR (with GR frequently virtually all flows can be adjusted even for modest changes in link capacity). As the flow adjustment process takes a non-negligible time, and

some flows can be thickened before other flows are thinned, some links can get overloaded during such a transient period inducing traffic (packet) losses. Clearly, the larger the number of adjusted flows, the higher the chance of link overloads in the transient period. For cost comparisons and flow adjustment issues of FT as compared with its variant allowing for thickening, the reader is referred to the recent papers Fouquet et al. (2014a) and Fouquet et al. (2014b).

The second reason are link overloads caused by erroneous approximation of flow thickening factors. If a certain state is not taken account in optimization, its tunnel thickening factors must be approximated, as it is done in FT for the thinning factors (see Section 9.2 and Section 9.3). However, since with thickening the nominal tunnel capacities are allowed to be exceeded in the states with reduced link capacity, wrong approximation of the thickening factors in such a state could easily lead to link overloads during the whole duration of the considered state.

To summarize, as far as unforeseen traffic losses are concerned, FT is safer than its variants admitting flow thickening (including GR).

10. Concluding remarks

Flow thinning (FT) is an original concept of a traffic protection strategy for communication networks with variable link capacity. Its affine variant, AFT, opens a way for distributed implementations of FT. As such (potentially implementable) strategies working at the logical tunnel level and reacting on-line to link capacity fluctuations have not been, to our knowledge, proposed, studying optimization models for FT is worthwhile, the more that the related multicommodity flow problems extend the area of resilient network design and are interesting from the OR viewpoint.

In the paper we have studied in-depth a basic FT optimization problem (referred to as FTOP, see Section 2), and its affine version (AFTOP, see Section 7). FTOP turns out to be \mathcal{NP} -hard and its non-compact link-path LP formulation requires path generation for which we have found a quite effectively solvable (though \mathcal{NP} -hard) pricing problem. We have also exhibited some special cases when path generation is polynomial and found a polynomially solvable algorithm for finding accurate lower bounds for FTOP. As FTOP supports FT, where the way flow can be thinning is not restricted, its solutions deliver a lower bound for the solutions of AFTOP—the optimization problem supporting potentially implementable affine versions of FT where the flows are thinned according to affine functions depending on the link availability coefficients characterizing a given network state. This makes FTOP important in evaluating traffic efficiency of AFT. Our AFTOP formulation is based on the notion of uncertainty polytope and because of that is capable of encompassing larger sets of link availability states in comparison with FTOP.

The numerical study shows that in terms of cost/traffic efficiency, AFT is almost as good as FT. FT, in turn, exhibits cost not significantly inferior to that of GR—the theoretically most cost efficient strategy, at least when (reasonable) demand reduction coefficients are allowed.

Acknowledgments

We would like to thank the anonymous referees and the associate editor for a careful reviewing process that helped to improve our first manuscript. The work of Michał Pióro was supported by the National Science Center (Poland) under grant 2015/17/B/ST7/03910 “Logical tunnel capacity control - a traffic routing and protection strategy for communication networks with variable link capacity”, while the work of Yoann Fouquet and Dritan Nace was carried out in the framework of Labex MS2T managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02). A part of this paper was written during M. Pióro’s stay at UTC also funded in the framework of Labex MS2T.

References

- Ahuja, R. K., T. L. Magnanti, J. B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ.
- Andersson, L., G. Swallow. 2003. RFC3468: The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols.
- Awduche, D., L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow. 2001. RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
- Ben-Tal, A., S. Boyd, A. Nemirovski. 2006. Extending scope of robust optimization: Comprehensive robust counterparts of uncertain problems. *Math. Program.* **107**(1-2) 63–89.
- Ben-Tal, A., L. El Ghaoui, A.S. Nemirovski. 2009. *Robust optimization*. Princeton University Press.
- Ben-Tal, A., A. Goryashko, E. Guslitzer, A. Nemirovski. 2004. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming* **99**(2) 351–376.
- Bertsimas, D., M. Sim. 2004. The price of robustness. *Operations Research* **52**(1) 35–53.
- Büsing, C., F. D’Andreagiovanni. 2012. New results about multi-band uncertainty in robust optimization. *SEA*. 63–74.
- Classen, G., D. Coudert, A.M.C.A. Koster, N. Nepomuceno. 2011. A chance-constrained model and cutting planes for fixed broadband wireless networks, in *Proc. INOC 2011. LNCS* **6701** 37–42.
- Coudert, D., P. Datta, S. Perennes, H. Rivano, M-E. Voge. 2007. Shared risk resource group: Complexity and approximability issues. *Parallel Processing Letters* **17**(6) 169–184.
- CPLEX. 2013. *IBM ILOG CPLEX 12.5 Reference Manual*. ILOG CPLEX Division, Gentilly, France.
- Dahl, G., M. Stoer. 1998. A cutting plane algorithm for multicommodity survivable network design problems. *INFORMS Journal on Computing* **10** 1–11.
- Fouquet, Y., D. Nace, M. Pióro, M. Poss, M. Żotkiewicz. 2014a. Flow adjustment - a flexible routing strategy for demand protection against multiple partial link failures. *accepted for the 4th International Conference on Advanced Communications and Computation INFOCOMP 2014*. Paris, France.

- Fouquet, Y., D. Nace, M. Pióro, M. Poss, M. Żotkiewicz. 2014b. Flow adjustment methods for survivable networks. *invited for the 16th International Telecommunications Network Strategy and Planning Symposium NETWORKS 2014*. Funchal, Madeira Island, Portugal.
- Goldsmith, A. 2005. *Wireless Communications*. Cambridge University Press.
- Grotschel, M., L. Lovasz, A. Schrijver. 1988. *Geometric Algorithms and Combinatorial Optimization*. Springer Verlag, Berlin.
- Grover, W. D. 2003. *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*. Prentice Hall.
- Harhira, H.A., S. Pierre. 2007. A mathematical model for the admission control problem in MPLS networks with end-to-end delay guarantees. *Proc. ICCCN 2007*. Honolulu, HI, USA.
- Katz, D., K. Kompella, D. Gan, D. Yeung. 2003. RFC3630: Traffic Engineering (TE) Extensions to OSPF Version 2.
- Khachian, L. 1979. A polynomial algorithm for linear programming. *Soviet Mathematics Doklady* **20** 191–194.
- Korte, B., J. Vygen. 2012. *Combinatorial Optimization Theory and Algorithms*. Springer.
- Koster, A.M.C.A., A. Zymolka, M. Jäger, R. Hülselman. 2005. Demand-wise shared protection for meshed optical networks. *Journal of Network and System Management* **13**(1) 35–55.
- Lasdon, L. 1970. *Optimization Theory for Large Systems*. MacMillan.
- Mattia, S. 2012. Robust optimization with multiple intervals. Tech. Rep. 12-07, IASI-CNR.
- Minei, I., J. Lucek. 2011. *MPLS-Enabled Applications: Emerging Developments and New Technologies*. John Wiley & Sons.
- Minoux, M. 1986. *Mathematical Programming: Theory and Algorithms*. John Wiley & Sons.
- Mogre, P.G., M. Hollic, R. Steimetz. 2007. Qos in wireless mesh networks: challenges, pitfalls, and roadmap to its realization. *Proc. NOSSDAV07*. Urbana-Champaign, IL, USA.
- Motzkin, T. S. 1951. Two consequences of the transposition theorem on linear inequalities. *Econometrica* **19** 184–185.
- Nemhauser, G. L., L. A. Wolsey. 1988. *Integer and Combinatorial Optimization*. John Wiley & Sons.
- Orlowski, S., M. Pióro. 2012. Complexity of column generation in network design with path-based survivability mechanisms. *Networks* **59** 132–147.
- Orlowski, S., M. Pióro, A. Tomaszewski, R. Wessály. 2010. SNDlib 1.0–Survivable Network Design Library. *Networks* **55**(3) 276–286.
- Ouorou, A., J.-P. Vial. 2007. A model for robust capacity planning for telecommunications networks under demand uncertainty. *Proceedings of the 6th International Workshop on Design and Reliable Communication Networks (DRCN 2007)*. 1–4.

- Pióro, M., D. Medhi. 2004. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan-Kaufman.
- Pióro, M., D. Nace, Y. Fouquet. 2013. An optimization model for communication networks with partial multiple link failures. *RNDM 2013, 5th International Workshop on Reliable Networks Design and Modeling, co-located with ICUMT 2013 Congress*. Almaty, Kazakhstan.
- Poss, M., C. Raack. 2013. Affine recourse for the robust network design problem: between static and dynamic routing. *Networks* **61**(2) 180–198.
- Shinko, I., D. Nace, Y. Fouquet. 2013. A study on a distributed rerouting scheme. *AINA 2013, Workshop NetMM*. 1461–1466. Barcelona, Spain.
- Strand, J., A.L. Chiu, R. Tkach. 2001. Issues for routing in the optical layer. *IEEE Communications Magazine* **39** 81–87.
- Tomaszewski, A., M. Pióro, M. Żotkiewicz. 2010. On the complexity of resilient network design. *Networks* **55** 109–118.
- Wessály, R. 2000. *Dimensioning survivable capacitated networks*. Cuvillier Verlag Gottingen.
- Wessály, R., S. Orlowski, A. Zymolka. 2005. Demand-wise shared protection revisited: A new model for survivable network design. *Proceedings of the 2nd International Network Optimization Conference (INOC 2005)*. 100–105.

Michał Pióro is a Professor at the Institute of Telecommunications, Warsaw University of Technology, Poland. At the same time he is a Professor at the Department of Electrical and Information Technology, Lund University, Sweden. He works on communication networks modelling, optimization and performance analysis, applying various Operations Research approaches.

Yoann Fouquet received his M.Sc. in Information Technology and Ph.D. in Operations Research from University of Technology of Compiègne, France. His research interests focus on network optimization and routing. Currently, he is a software engineer at Thales France.

Dritan Nace is a Professor at the Department of Computer Science, University of Technology of Compiègne, France. His research interests include network optimization and routing, and, more generally, combinatorial optimization.

Michael Poss is a CNRS researcher at the LIRMM laboratory, a mixed research unit that depends on both the University of Montpellier and the National Center for Scientific Research (CNRS). His current research focuses on robust combinatorial optimization.

Examples, proofs, derivations, and additional results

EC.1. List of abbreviations

Table EC.1 Paper-specific list of abbreviations

AFT	affine flow thinning
AFTOP	affine flow thinning optimization problem
AGR	affine global rerouting
DL	double-link (failure scenario), includes SL
DSP	dual separation problem
DWSP	demand-wise shared protection
ER	elastic rerouting
FT	flow thinning
FTOP	flow thinning optimization problem
FTOP ^{<}	flow thinning optimization problem with predefined admissible paths
GR	global rerouting
LBP	lower bound problem
LP	linear programming
PG	path generation
PP	pricing problem
RAFT	restricted affine flow thinning
RAGR	restricted affine global rerouting
SL	single-link (failure scenario)
SN	single-node (failure scenario)
TL	triple-link (failure scenario), includes DL

EC.2. Examples related to FTOP

EXAMPLE EC.1. Consider the network depicted in Figure EC.1. It is a 3-node, 4-link directed network with $\mathcal{V} = \{v, w, t\}$ and $\mathcal{E} = \{1, 2, 3, 4\}$. The unit link costs are all equal to 1: $\xi_e = 1, e \in \mathcal{E}$. There are two demands ($\mathcal{D} = \{1, 2\}$), demand $d = 1$ from node v to node t , and demand $d = 2$ from node w to node t , with the nominal traffic volumes $h_1^0 = h_2^0 = 1$. Each demand has two admissible paths: $\mathcal{P}_1 = \{p_{11}, p_{12}\}$ where $p_{11} = \{1\}, p_{12} = \{3, 2\}$, and $\mathcal{P}_2 = \{p_{21}, p_{22}\}$ where $p_{21} = \{2\}, p_{22} = \{4, 1\}$. We assume that links 3 and 4 are always fully available while links 1 and 2 can fail, but only one at a time. When link $e = 1$ or $e = 2$ fails, it losses half of its capacity. Thus, there are three availability states $\mathcal{S} = \{1, 2, 3\}$ and we assume that in each state the nominal traffic volumes are to be realized so that the demand coefficients are all equal to 1 ($\beta_d^s = 1, d \in \mathcal{D}, s \in \mathcal{S}$). State $s = 3$ is a maximum state ($\mathcal{S}_0 = \{3\}$) with all links fully available: $\alpha_e^3 = 1, e \in \mathcal{E}$. In state $s = 1$, link $e = 1$ fails: $\alpha_1^1 = \frac{1}{2}, \alpha_e^1 = 1, e = 2, 3, 4$. Symmetrically, in state $s = 2$, link $e = 2$ fails: $\alpha_2^2 = \frac{1}{2}, \alpha_e^2 = 1, e = 1, 3, 4$.

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

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

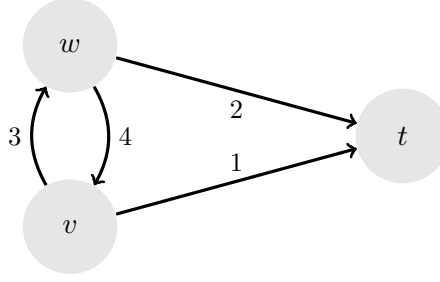


Figure EC.1 A simple network example

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

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

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

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

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

EXAMPLE EC.2. For the network depicted in Figure EC.1, the solution given in Example EC.1 is feasible and optimal for (1). Note that the flows x^0, x^1, x^2 of Example EC.1 are unique at optimality while x^3 are not. For example, the flows $x^3 = x^0$ are also optimal.

If we change the unit costs of links 3 and 4 to M , where M is a large number ($M \gg 1$), then the optimal solution of FTOP will change. The optimal cost will increase to $C = 4$ and the optimal tunnel capacities will be as follows:

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

$$y_1^0 = 2, y_2^0 = 2, y_3^0 = 0, y_4^0 = 0$$

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

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

$$1 \leq x_{11}^3 \leq 2, x_{12}^3 = 0, 1 \leq x_{21}^3 \leq 2, x_{22}^3 = 0.$$

EXAMPLE EC.3. Consider for the network described in Example EC.1 and depicted in Figure EC.1 with the two sets of admissible paths reduced to $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ where $\mathcal{P}_1 = \{p_{11}\}$ and $\mathcal{P}_2 = \{p_{21}\}$. One of the optimal solutions of the resulting instance $P(\mathcal{P})$ is as follows:

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

The corresponding optimal solution of the dual problem $D(\mathcal{P})$ (see (2)) is in turn given by:

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

and all other dual variables $\pi_1^0, \pi_2^0, \lambda_1^2, \lambda_2^1, \lambda_1^3, \lambda_2^3, \sigma_d^s, s \in \mathcal{S}, d \in \mathcal{D}$, equal to 0. Hence, applying PG, we find out that paths p_{12} and p_{22} have their total dual lengths (defined by (4)) equal to $\langle p_{12} \rangle = \langle p_{22} \rangle = -1$. Since these values are negative, both paths are added to the problem so that the set of admissible paths \mathcal{P} becomes set equal to $\hat{\mathcal{P}}$, i.e., the set of all (loop-less) paths in the network graph for the considered two demands. An optimal solution of the resulting instance $P(\hat{\mathcal{P}})$ of FTOP is given in Example EC.1.

EC.3. Polyhedron Ω and proofs of the separation and pricing propositions

EC.3.1 Projected polyhedron Ω

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

$$\pi_e^0 + \sum_{s \in \mathcal{S}_e} \alpha_e^s \pi_e^s \leq \xi_e, \quad e \in \mathcal{E} \quad (\text{EC.1a})$$

$$\sum_{e \in \mathcal{E}} \delta_{edp} \pi_e^0 \geq \sum_{s \in \mathcal{S}'} (\lambda_d^s - \sum_{e \in \mathcal{E}_s} \delta_{edp} \pi_e^s), \quad d \in \mathcal{D}, p \in \hat{\mathcal{P}}_d, \mathcal{S}' \subseteq \mathcal{S} \quad (\text{EC.1b})$$

$$\pi \geq 0, \lambda \geq 0. \quad (\text{EC.1c})$$

The above characterization follows from the fact that for a fixed path $q \in \hat{\mathcal{P}}_d$, the quantity on the right hand side of (EC.1b) computed for any given $(\pi, \lambda) \in \mathbb{R}_+^n$ attains its maximum for $\mathcal{S}' = \mathcal{S}_q(\pi, \lambda)$. This means that for the given path, the proper inequality, i.e., inequality (7), dominates the remaining inequalities in (EC.1b) for each $(\pi, \lambda) \in \mathbb{R}_+^n$. Clearly, characterization (EC.1) can be easily obtained with Fourier-Motzkin elimination (see Motzkin (1951)).

EC.3.2 Proof of Proposition 1

Consider vector $(\pi, \lambda) \in \mathbb{R}_+^n$ with π fulfilling constraints (2b) and a given path $q \in \hat{\mathcal{P}}_d$. We will show that system of inequalities (3) is infeasible if, and only if, inequality (6) holds. Clearly, this implies that (7) separates (π, λ) from Ω .

Suppose $\langle q \rangle \geq 0$, that is, $|q|^0 \geq \sum_{s \in \mathcal{S}_q(\pi, \lambda)} (\lambda_d^s - |q|^s)$. Then, obviously,

$$\sigma_{dq}^s := \begin{cases} \lambda_d^s - |q|^s, & \text{if } s \in \mathcal{S}_q(\pi, \lambda) \\ 0, & \text{if } s \in \mathcal{S} \setminus \mathcal{S}_q(\pi, \lambda) \end{cases} \quad (\text{EC.2})$$

is a feasible solution of the system of inequalities (3).

Now suppose $\langle q \rangle < 0$, that is,

$$\sum_{s \in \mathcal{S}_q(\pi, \lambda)} (\lambda_d^s - |q|^s) > |q|^0. \quad (\text{EC.3})$$

Then, assuming feasibility of (3), due to (5), (3b), (3c), (3a), and (EC.3), respectively, the following sequence of inequalities must hold:

$$0 \leq \sum_{s \in \mathcal{S}_q(\pi, \lambda)} (\lambda_d^s - |q|^s) \leq \sum_{s \in \mathcal{S}_q(\pi, \lambda)} \sigma_{dq}^s \leq \sum_{s \in \mathcal{S}} \sigma_{dq}^s \leq |q|^0 < \sum_{s \in \mathcal{S}_q(\pi, \lambda)} (\lambda_d^s - |q|^s). \quad (\text{EC.4})$$

This is a contradiction since in (EC.4) the second term is strictly less than the last term but they are identical. Thus, system (3) is infeasible.

EC.3.3 Proof of Proposition 2

Since (π, λ) is a solution of $D(\mathcal{P})$, $(\pi, \lambda) \in \mathbb{R}_+^n$ and hence, as already noticed when introducing characterization (EC.1) of the projected dual polyhedron Ω , the equality

$$\sum_{s \in \mathcal{S}_q(\pi, \lambda)} (\lambda_d^s - \sum_{e \in \mathcal{E}_s} \delta_{edq} \pi_e^s) = \max \left\{ \sum_{s \in \mathcal{S}'} (\lambda_d^s - \sum_{e \in \mathcal{E}_s} \delta_{edq} \pi_e^s) : \mathcal{S}' \subseteq \mathcal{S} \right\} \quad (\text{EC.5})$$

holds. Assume that for some $\mathcal{S}' \subseteq \mathcal{S}$, the corresponding constraint (EC.1b) is broken, i.e., $\sum_{e \in \mathcal{E}} \delta_{edq} \pi_e^0 < \sum_{s \in \mathcal{S}'} (\lambda_d^s - \sum_{e \in \mathcal{E}_s} \delta_{edq} \pi_e^s)$. Then, due to (EC.5), also the constraint $\sum_{e \in \mathcal{E}} \delta_{edq} \pi_e^0 < \sum_{s \in \mathcal{S}_q(\pi, \lambda)} (\lambda_d^s - \sum_{e \in \mathcal{E}_s} \delta_{edq} \pi_e^s)$ is broken, and, for the same reason, the degree of violation of the latter constraint, i.e., $D(q) = \sum_{s \in \mathcal{S}_q(\pi, \lambda)} (\lambda_d^s - \sum_{e \in \mathcal{E}_s} \delta_{edq} \pi_e^s) - \sum_{e \in \mathcal{E}} \delta_{edq} \pi_e^0 = -\langle q \rangle$ is the largest among all the constraints (EC.1b) corresponding to path q .

EC.4. The flow thickening extension of FTOP

A natural extension of FT, and thus of FTOP, allows for tunnel thickening limited by additional parameters $t_d \geq 1, d \in \mathcal{D}$, called the *thickening bounds*. The extension is obtained by modifying constraint (1e):

$$x_{dp}^s \leq t_d x_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, s \in \mathcal{S}. \quad (\text{EC.6})$$

Clearly, with $t_d > 1$ a (limited) path-flow thickening for demand $d \in \mathcal{D}$ is permissible, not only thinning. In fact, this modification (even when the thickening factor is made tunnel-dependent) has virtually the same properties as FTOP and can be approached essentially in the way discussed in the rest of the paper.

Flow thickening leads to cheaper solutions as compared to pure thinning, and when all $t_d \rightarrow \infty$ the resulting modification of FTOP becomes equivalent to the GR optimization problem discussed above. Yet, as discussed in Section 9.4, flow thickening may impose implementation problems.

When the thickening bounds $t_d, d \in \mathcal{D}$, are considered (constraint (1e) is substituted by (EC.6)) then constraint (2c) in the related dual problem takes the form

$$\sum_{s \in \mathcal{S}} \sigma_{dp}^s \leq \frac{1}{t_d} \sum_{e \in \mathcal{E}} \delta_{edp} \pi_e^0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d \quad (\text{EC.7})$$

while the rest of the constraints remain unchanged. When all $t_d \rightarrow \infty$, i.e., when the primal problem becomes the GR optimization problem, the right-hand side of constraint (EC.7) becomes equal to 0 which forces all dual variables $\sigma_{dp}^s, s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}_d$, to be equal to 0. Hence, constraint (2d) takes the form

$$\lambda_d^s \leq \sum_{e \in \mathcal{E}_s} \delta_{edp} \pi_e^s, \quad s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}_d \quad (\text{EC.8})$$

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

EC.5. Selected results for single-link failures

EC.5.1 Single-link failures - proof of Lemma 3

First, we observe that the total dual length of path $q \in \hat{\mathcal{P}}_d$ defined by (4) can be (in fact also in the general case of \mathcal{S}) equivalently expressed as

$$\langle q \rangle = (|q|^0 + \sum_{s \in \mathcal{S}} [q]^s) - \sum_{s \in \mathcal{S}} \lambda_d^s, \quad (\text{EC.9})$$

where the *modified dual length* $[q]^s$ of path q in state $s \in \mathcal{S}$ is defined as

$$[q]^s := \begin{cases} |q|^s, & \text{if } |q|^s \leq \lambda_d^s \\ \lambda_d^s, & \text{otherwise.} \end{cases} \quad (\text{EC.10})$$

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

$$|q|^s = \begin{cases} 0, & \text{if } s \in \mathcal{S}_0, \text{ or } s \in \mathcal{S}(e) \text{ and } e \notin q \\ \pi_e^s, & \text{if } s \in \mathcal{S}(e) \text{ and } e \in q \end{cases} \quad (\text{EC.11})$$

the modified dual path lengths are equal to:

$$[q]^s = \begin{cases} 0, & \text{if } s \in \mathcal{S}_0, \text{ or } s \in \mathcal{S}(e) \text{ and } e \notin q \\ \pi_e^s, & \text{if } s \in \mathcal{S}(e) \text{ and } e \in q \text{ and } \pi_e^s \leq \lambda_d^s \\ \lambda_d^s, & \text{if } s \in \mathcal{S}(e) \text{ and } e \in q \text{ and } \pi_e^s > \lambda_d^s. \end{cases} \quad (\text{EC.12})$$

Now we make use of the link weights $\omega(e), e \in \mathcal{E}$, specified by (13) in the formulation of Lemma 3:

$$\omega(e) = \pi_e^0 + \sum_{s \in \mathcal{S}(e)} \max\{\pi_e^s, \lambda_d^s\}. \quad (\text{EC.13})$$

By (EC.13) and (EC.12) we have

$$\sum_{e \in q} \omega(e) = |q|^0 + \sum_{e \in q} \sum_{s \in \mathcal{S}(e)} [q]^s = |q|^0 + \sum_{s \in \mathcal{S}} [q]^s, \quad (\text{EC.14})$$

where the quantity on the left-hand side is equal the length of path q with respect to the link weights defined by (EC.13) (and (13)). Hence, by (EC.9) we finally obtain the equality

$$\sum_{e \in q} \omega(e) = \langle q \rangle + \sum_{s \in \mathcal{S}} \lambda_d^s, \quad (\text{EC.15})$$

which is equivalent to (12).

EC.5.2 Single-link failures – compact FTOP formulation

A compact node-link FTOP formulation for a given single-link failure scenario $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1$, where $\mathcal{S}_1 = \bigcup_{e \in \mathcal{E}} \mathcal{S}(e)$, is as follows.

$$\min \quad C = \sum_{e \in \mathcal{E}} \xi_e y_e^0 \quad (\text{EC.16a})$$

$$\sum_{e \in \delta^+(o(d))} x_{ed}^0 - \sum_{e \in \delta^-(o(d))} x_{ed}^0 = X_d^0, \quad d \in \mathcal{D} \quad (\text{EC.16b})$$

$$\sum_{e \in \delta^+(v)} x_{ed}^0 - \sum_{e \in \delta^-(v)} x_{ed}^0 = 0, \quad d \in \mathcal{D}, v \in \mathcal{V} \setminus \{o(d), t(d)\} \quad (\text{EC.16c})$$

$$\sum_{d \in \mathcal{D}} x_{ed}^0 \leq y_e^0, \quad e \in \mathcal{E} \quad (\text{EC.16d})$$

$$X_d^0 \geq h_d^s, \quad d \in \mathcal{D}, s \in \mathcal{S}_0 \quad (\text{EC.16e})$$

$$X_d^0 - z_{ed}^s \geq h_d^s, \quad d \in \mathcal{D}, e \in \mathcal{E}, s \in \mathcal{S}(e) \quad (\text{EC.16f})$$

$$\sum_{d \in \mathcal{D}} (x_{ed}^0 - z_{ed}^s) \leq \alpha_e^s y_e^0, \quad e \in \mathcal{E}, s \in \mathcal{S}(e) \quad (\text{EC.16g})$$

$$z_{ed}^s \leq x_{ed}^0, \quad d \in \mathcal{D}, e \in \mathcal{E}, s \in \mathcal{S}(e) \quad (\text{EC.16h})$$

$$\text{all variables } x^0, X^0, z, y^0 \text{ continuous and nonnegative.} \quad (\text{EC.16i})$$

In the formulation, variables $x_{ed}^0, e \in \mathcal{E}$, are the link-flows of demand $d \in \mathcal{D}$ (realizing its nominal path-flows), while variable X_d^0 expresses the overall flow of demand d from $o(d)$ to $t(d)$ (realized by its nominal path-flows). Variables $z_{ed}^s, e \in \mathcal{E}, d \in \mathcal{D}, s \in \mathcal{S}(e)$, in turn, specify the amount of link-flow by which the nominal link-flow x_{ed}^0 is reduced in state $s \in \mathcal{S}(e)$. Constraints (EC.16b) and (EC.16c) are conservation equations for the nominal link-flows. Capacity constraint (EC.16d) assures that the nominal link loads do not exceed the maximum link capacities, and constraint (EC.16g) assures that the capacity available on link $e \in \mathcal{E}$ is not exceeded by its loads in its failure states. Demand constraints (EC.16e) and (EC.16f) take care about satisfaction of the assumed demand volumes in the maximum states and the single-link failure states, respectively. Finally, constraint (EC.16h) bounds the possible link-flow reductions.

In the case of a failure of a single link, say of link $e' \in \mathcal{E}$ in one of its failure states $s \in \mathcal{S}(e')$, any given pattern of nominal path-flows realizing the nominal link-flows $x_{ed}^0, e \in \mathcal{E}$, for a given demand

$d \in \mathcal{D}$, can be properly thinned according to the values of $z_{e'd}^s, d \in \mathcal{D}$. Let \mathcal{P}_d and \mathcal{L}_d be the sets of elementary $o(d) - t(d)$ paths and loops, respectively, assigned non-zero path/loop-flows x_{dp}^0 when realizing the nominal link-flows $x_{ed}^0, e \in \mathcal{E}$, of a given demand $d \in \mathcal{D}$ (such sets of paths and loops exist, see Theorem 8.8 in Korte and Vygen (2012)). Because $\sum_{p \in \mathcal{P}_d} x_{dp}^0 = X_d^0$, due to constraints (EC.16f) and (EC.16g), we can apply thinning for any given failure state $s \in \mathcal{S}(e')$ only to the nominal path-flows in \mathcal{P}_d (and not to the loops in \mathcal{L}_d) containing the failing link e' . Let then $\mathcal{Q}_{e'd} \subseteq \mathcal{P}_d$ denote the set of paths that contain the considered link e' , so that $\sum_{p \in \mathcal{Q}_{e'd}} x_{dp}^0 = x_{e'd}^0$. We can consistently thin down the nominal path-flows through link e' in the considered state $s \in \mathcal{S}(e')$ according to the formula

$$x_{dp}^s := x_{dp}^0 - \frac{z_{e'd}^s}{x_{e'd}^0} \cdot x_{dp}^0, \quad p \in \mathcal{Q}_{e'd}. \quad (\text{EC.17})$$

Note that possible loops in the optimal solutions can be eliminated as their capacity is not used by the state-dependent path-flows $x_{dp}^s, p \in \mathcal{P}_d, s \in \mathcal{S}$, realizing the demand volume h_d^s defined by (EC.17). Since, as discussed Section 2.2, FTOP (1) does not require non-elementary paths (and thus loops in the link-flows), formulation (EC.16) is correct. In fact, when all link unit capacity costs $\xi_e, e \in \mathcal{E}$, are strictly positive, the loops will not appear in the optimal solutions of (EC.16).

It is interesting to note that for the single-link failure scenarios the LBP formulation (14) remains to be only a lower bound for the exact formulation (EC.16). In fact, when an optimal solution $x^0, x^s, s \in \mathcal{S}, y^0$ of (14) does not contain loops in the flow x^0 then the optimal values of (14) determine the corresponding feasible solution $x^0, z^s, s \in \mathcal{S}, y^0$ of (EC.16) with the same x^0, y^0 and $z_{ed}^s := x_{ed}^0 - x_{ed}^s, d \in \mathcal{D}, e \in \mathcal{E}, s \in \mathcal{S}(e)$. Since LBP is a lower bound for FTOP, its optimum is at the same time the optimum for FTOP. However, if the link-flow x^0 contains loops, then the solution C' of LBP can be strictly smaller than the optimal solution C of FTOP, as illustrated in Example EC.4 below. This is because no counterpart of constraint (EC.16f) is present in (14) and hence we may thin the loops without thinning the realized overall demand volume X_d^0 .

EXAMPLE EC.4. Consider the network depicted in Figure EC.2. There is one demand (from o to t) and two states, $s = 1$ and $s = 2$, with link 1 and link 2 failing, respectively. In both states the failing link has the same availability coefficient α , and the demand volume to be realized is equal to h . The unit link costs are as follows: $\xi_1 = \xi_2 = \xi_3 = 1, \xi_4 = M$, where $M \gg 1$.

In the considered cost setting with $\xi_4 \gg 1$, optimal solutions of (EC.16) (and hence of FTOP) are induced by the nominal path/loop-flows of the form: $x_1^0 = f$ (on the upper path (o, v, t)), $x_2^0 = f$ (on the lower path (o, v, t)), $x_3^0 = 0$ (on the upper loop (o, v, o)), $x_4^0 = 0$ (on the lower loop (o, v, o)). There are two cases to consider.

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

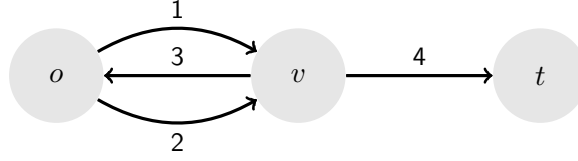


Figure EC.2 A network example with a loop in the optimum link-flow

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

The optimal solution of LBP (14), in turn, is induced by the two nominal path-flows equal to $\frac{h}{2}$, and by the two nominal loop-flows equal to $(\frac{1}{1+\alpha} - \frac{1}{2})h = \frac{(1-\alpha)h}{2(1+\alpha)}$. Then the corresponding nominal link-flows are equal to $x^0 = (\frac{h}{1+\alpha}, \frac{h}{1+\alpha}, \frac{(1-\alpha)h}{1+\alpha}, h)$, and the resulting cost is equal to $C' = \frac{(3-\alpha)h}{1+\alpha} + Mh$. In the availability state $s = 1$, the link-flows become equal to $x^1 = (\frac{\alpha h}{1+\alpha}, \frac{h}{1+\alpha}, 0, h)$, and in $s = 2$, the link-flows are symmetric and equal to $x^2 = (\frac{h}{1+\alpha}, \frac{\alpha h}{1+\alpha}, 0, h)$.

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

We observe that in the considered example the loops (o, v, o) could be eliminated by putting $x_{ed}^0 = 0$ ($d \in \mathcal{D}$ and $e \in \delta^-(o(d))$ or $e \in \delta^+(t(d))$) and $x_{ed}^s = 0$ ($s \in \mathcal{S}$, $x_{ed}^s = 0$ ($s \in \mathcal{S}, d \in \mathcal{D}$ and $e \in \delta^-(o(d))$ or $e \in \delta^+(t(d))$) in (14). Yet, this does not help much as in the general case loops will appear anyhow, only they will not involve the demands' origins and destinations.

In the case of undirected links the situation is analogous to the directed case. The standard counterpart of formulation (EC.16) for the undirected links (with two oppositely directed link-flows corresponding to each undirected link) remains correct, and the LBP formulation modified for undirected links is still not equivalent to the modified formulation (EC.16).

EC.6. Single-node failures

A single-node failures scenario consists of the states with limited availability of the links incident to one single node. Thus, a single-node failure scenario \mathcal{S} is composed of a set of maximum states and a set of single-node failure states: $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_2$, where \mathcal{S}_0 is a set of maximum states, and $\mathcal{S}_2 := \bigcup_{v \in \mathcal{V}} \mathcal{S}(v)$ is a set of single-node failure states. Each set $\mathcal{S}(v)$, if not empty, represents a set of states in which only node v fails. That is, $s \in \mathcal{S}(v)$ implies that $\alpha_e^s \leq 1$, $e \in \delta(v)$, $\alpha_e^s = 1$, $e \in \mathcal{E} \setminus \delta(v)$, and $\alpha_e^{s(v)} < 1$ for at least one $e \in \delta(v)$ (where $\delta(v) := \delta^-(v) \cup \delta^+(v)$). If node v does not fail at all, then the set $\mathcal{S}(v)$ is empty.

EC.6.1 Single-node failures – the pricing problem

In the single-node failure scenario the pricing problem (8) is also polynomial. To apply shortest path computation to solve PP, we consider, for each $d \in \mathcal{D}$, a transformed graph $\hat{\mathcal{G}}(d) = (\hat{\mathcal{V}}(d), \hat{\mathcal{E}}(d))$,

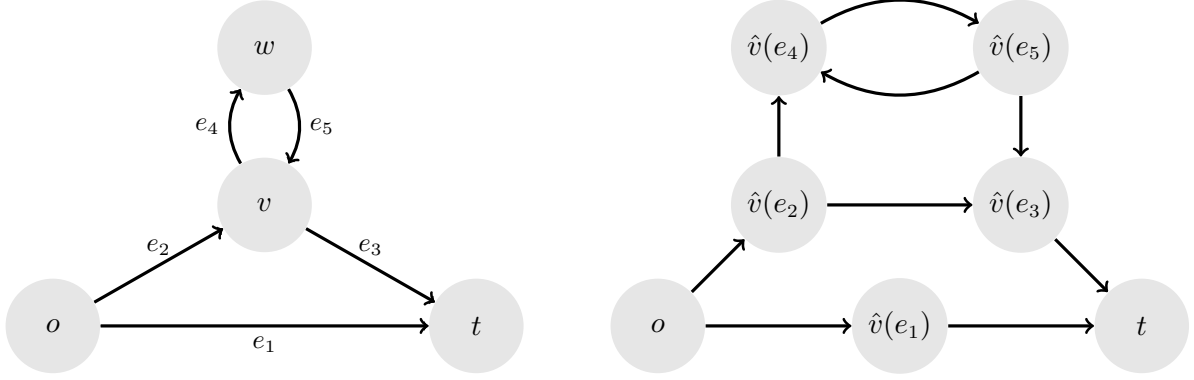


Figure EC.3 Original graph \mathcal{G} and transformed graph $\hat{\mathcal{G}}(d)$

constructed from the original network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ by exchanging, in a way, the roles of nodes and links. The transformation is illustrated in Figure EC.3 for a demand d from node o to node t . The original graph \mathcal{G} is depicted on the left-hand side while the transformed graph $\hat{\mathcal{G}}(d)$ is shown on the right-hand side.

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

$$\hat{\mathcal{V}}(d) := \{\hat{v}(e) : e \in \mathcal{E}\} \cup \{o(d), t(d)\}; \quad (\hat{v}(e), \hat{v}(f)) \in \hat{\mathcal{E}}(d) \Leftrightarrow \exists v \in \mathcal{V} \setminus \{o(d), t(d)\}, e \in \delta^-(v) \wedge f \in \delta^+(v);$$

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

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

Lemma 5 Consider a given single-node failure scenario \mathcal{S} and a fixed demand $d \in \mathcal{D}$. The total dual length of any path $q \in \hat{\mathcal{P}}_d$ is given by the formula

$$\langle q \rangle = \sum_{\hat{e} \in \hat{q}} \omega(\hat{e}) - \sum_{s \in \mathcal{S}} \lambda_d^s, \quad (\text{EC.18})$$

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

$$\omega(\hat{v}(e), \hat{v}(f)) := \pi_f^0 + \sum_{s \in \mathcal{S}(v)} \min\{\pi_e^s + \pi_f^s, \lambda_d^s\}, \quad v \in \mathcal{V} \setminus \{o(d), t(d)\}, e \in \delta^-(v), f \in \delta^+(v) \quad (\text{EC.19a})$$

$$\omega(o(d), \hat{v}(f)) := \pi_f^0 + \sum_{s \in \mathcal{S}(o(d))} \min\{\pi_f^s, \lambda_d^s\}, \quad f \in \delta^+(o(d)) \quad (\text{EC.19b})$$

$$\omega(\hat{v}(e), t(d)) := \sum_{s \in \mathcal{S}(t(d))} \min\{\pi_e^s, \lambda_d^s\}, \quad e \in \delta^-(t(d)). \quad (\text{EC.19c})$$

As in the proof of Lemma 3 (see (EC.9)), the total dual length of path $q \in \hat{\mathcal{P}}_d$ defined by (4) can be expressed as:

$$\langle q \rangle = (|q|^0 + \sum_{s \in \mathcal{S}} [q]^s) - \sum_{s \in \mathcal{S}} \lambda_d^s. \quad (\text{EC.20})$$

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

$$|q|^s = \begin{cases} 0, & \text{if } s \in \mathcal{S}_0, \text{ or } s \in \mathcal{S}(v) \text{ and } q \cap \delta(v) = \emptyset \\ \pi_e^s, & \text{if } s \in \mathcal{S}(v) \text{ and } q \cap \delta(v) = \{e\} \\ \pi_{e'}^s + \pi_{e''}^s, & \text{if } s \in \mathcal{S}(v) \text{ and } q \cap \delta(v) = \{e', e''\}. \end{cases} \quad (\text{EC.21})$$

Thus, the modified dual lengths, defined in (EC.10), are as follows:

$$[q]^s = \begin{cases} 0, & \text{if } s \in \mathcal{S}_0, \text{ or } s \in \mathcal{S}(v) \text{ and } q \cap \delta(v) = \emptyset \\ \pi_e^s, & \text{if } s \in \mathcal{S}(v) \text{ and } q \cap \delta(v) = \{e\} \text{ and } \pi_e^{s(v)} \leq \lambda_d^{s(v)} \\ \lambda_d^s, & \text{if } s \in \mathcal{S}(v) \text{ and } q \cap \delta(v) = \{e\} \text{ and } \pi_e^{s(v)} > \lambda_d^{s(v)} \\ \pi_{e'}^s + \pi_{e''}^s, & \text{if } s \in \mathcal{S}(v) \text{ and } q \cap \delta(v) = \{e', e''\} \text{ and } \pi_{e'}^{s(v)} + \pi_{e''}^{s(v)} \leq \lambda_d^{s(v)} \\ \lambda_d^s, & \text{if } s \in \mathcal{S}(v) \text{ and } q \cap \delta(v) = \{e', e''\} \text{ and } \pi_{e'}^{s(v)} + \pi_{e''}^{s(v)} > \lambda_d^{s(v)}. \end{cases} \quad (\text{EC.22})$$

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

Now we make use of the link weights $\omega(\hat{e}), \hat{e} \in \hat{\mathcal{E}}$, specified by (EC.19) in the formulation of Lemma 5:

$$\omega(\hat{v}(e), \hat{v}(f)) := \pi_f^0 + \sum_{s \in \mathcal{S}(v)} \min\{\pi_e^s + \pi_f^s, \lambda_d^s\}, \quad v \in \mathcal{V} \setminus \{o(d), t(d)\}, e \in \delta^-(v), f \in \delta^+(v) \quad (\text{EC.23a})$$

$$\omega(o(d), \hat{v}(f)) := \pi_f^0 + \sum_{s \in \mathcal{S}(o(d))} \min\{\pi_f^s, \lambda_d^s\}, \quad f \in \delta^+(o(d)) \quad (\text{EC.23b})$$

$$\omega(\hat{v}(e), t(d)) := \sum_{s \in \mathcal{S}(t(d))} \min\{\pi_e^s, \lambda_d^s\}, \quad e \in \delta^-(t(d)) \quad (\text{EC.23c})$$

and consider a path $\hat{q} = (o(d), \hat{v}(e_1), \hat{v}(e_2), \dots, \hat{v}(e_n), t(d))$. We observe that the length $\sum_{\hat{e} \in \hat{q}} \omega(\hat{e})$ of path \hat{q} with respect to the link weights (EC.23) obeys the equality

$$\sum_{\hat{e} \in \hat{q}} \omega(\hat{e}) = |q|^0 + \sum_{s \in \mathcal{S}} [q]^s \quad (\text{EC.24})$$

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

$$\sum_{\hat{e} \in \hat{q}} \omega(\hat{e}) = \langle q \rangle + \sum_{s \in \mathcal{S}} \lambda_d^s, \quad (\text{EC.25})$$

which is equivalent to (EC.18).

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

The above construction can be applied, after a slight modification, to undirected graphs. In the undirected case we first define the transformed graph as an undirect graph, similarly as in the directed case. Then we transform it to a bi-directed graph, with two oppositely directed arcs corresponding to each undirected link. The two arcs in each such pair are assigned non-symmetric arc weights, differing in the π_f^0 value that is determined by the direction of the arc (see definition (EC.19a)).

Since the term $\sum_{s \in \mathcal{S}} \lambda_d^s$ in (12) is constant, solving PP is equivalent to finding a shortest path in the transformed graph with respect to the link weights $\omega(\hat{e}), \hat{e} \in \hat{\mathcal{E}}$. Hence, PP is polynomial, and so is FTOP. A compact LP formulation for the considered case of FTOP is given in Section EC.6.2.

EC.6.2 Single-node failures – compact FTOP formulation

A compact node-link FTOP formulation for a given single-node failure scenario $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_2$, where $\mathcal{S}_2 = \bigcup_{v \in \mathcal{V}} \mathcal{S}(v)$, is as follows.

$$\min \quad C = \sum_{e \in \mathcal{E}} \xi_e y_e^0 \quad (\text{EC.26a})$$

$$\sum_{e \in \delta^+(o(d))} x_{ed}^0 - \sum_{e \in \delta^-(o(d))} x_{ed}^0 = X_d^0, \quad d \in \mathcal{D} \quad (\text{EC.26b})$$

$$\sum_{e \in \delta^+(v)} x_{ed}^0 - \sum_{e \in \delta^-(v)} x_{ed}^0 = 0, \quad d \in \mathcal{D}, v \in \mathcal{V} \setminus \{o(d), t(d)\} \quad (\text{EC.26c})$$

$$\sum_{d \in \mathcal{D}} x_{ed}^0 \leq y_e^0, \quad e \in \mathcal{E} \quad (\text{EC.26d})$$

$$X_d^0 \geq h_d^s, \quad d \in \mathcal{D}, s \in \mathcal{S}_0 \quad (\text{EC.26e})$$

$$\sum_{e^- \in \delta^-(v)} w_{e^-ed}^v = x_{ed}^0, \quad d \in \mathcal{D}, v \in \mathcal{V} \setminus \{o(d), t(d)\}, e \in \delta^+(v) \quad (\text{EC.26f})$$

$$\sum_{e^+ \in \delta^+(v)} w_{ee+d}^v = x_{ed}^0, \quad d \in \mathcal{D}, v \in \mathcal{V} \setminus \{o(d), t(d)\}, e \in \delta^-(v) \quad (\text{EC.26g})$$

$$z_{e^-e+d}^s \leq w_{e^-e+d}^v, \quad d \in \mathcal{D}, v \in \mathcal{V} \setminus \{o(d), t(d)\}, e^- \in \delta^-(v), e^+ \in \delta^+(v), s \in \mathcal{S}(v) \quad (\text{EC.26h})$$

$$X_d^0 - \sum_{e^- \in \delta^-(v)} \sum_{e^+ \in \delta^+(v)} z_{e^-e+d}^s \geq h_d^s, \quad d \in \mathcal{D}, v \in \mathcal{V} \setminus \{o(d), t(d)\}, s \in \mathcal{S}(v) \quad (\text{EC.26i})$$

$$Z_{ed}^s \leq x_{ed}^0, \quad d \in \mathcal{D}, e \in \delta^+(o(d)), s \in \mathcal{S}(o(d)) \quad (\text{EC.26j})$$

$$X_d^0 - \sum_{e \in \delta^+(o(d))} Z_{ed}^s \geq h_d^s, \quad d \in \mathcal{D}, s \in \mathcal{S}(o(d)) \quad (\text{EC.26k})$$

$$Z_{ed}^s \leq x_{ed}^0, \quad d \in \mathcal{D}, \quad e \in \delta^-(t(d)), \quad s \in \mathcal{S}(t(d)) \quad (\text{EC.26l})$$

$$X_d^0 - \sum_{e \in \delta^-(t(d))} Z_{ed}^s \geq h_d^s, \quad d \in \mathcal{D}, \quad s \in \mathcal{S}(t(d)) \quad (\text{EC.26m})$$

$$\sum_{d \in \mathcal{D}} x_{ed}^0 - \sum_{d \in \mathcal{D}(v)} \sum_{e^- \in \delta^-(v)} z_{e^-ed}^s - \sum_{d \in \mathcal{D}^+(v)} Z_{ed}^s \leq \alpha_e^s y_e^0, \quad v \in \mathcal{V}, \quad e \in \delta^+(v), \quad s \in \mathcal{S}(v) \quad (\text{EC.26n})$$

$$\sum_{d \in \mathcal{D}} x_{ed}^0 - \sum_{d \in \mathcal{D}(v)} \sum_{e^+ \in \delta^+(v)} z_{ee^+d}^s - \sum_{d \in \mathcal{D}^-(v)} Z_{ed}^s \leq \alpha_e^s y_e^0, \quad v \in \mathcal{V}, \quad e \in \delta^-(v), \quad s \in \mathcal{S}(v) \quad (\text{EC.26o})$$

$$\text{all variables } x^0, X^0, w, z, Z, y^0 \text{ continuous and nonnegative.} \quad (\text{EC.26p})$$

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

The values of transit variables $w_{e^-e^+d}^v$ can be used to define the overall flow on the paths $p \in \hat{\mathcal{P}}_d$ traversing a transit node v of a demand $d \in \mathcal{D}$ containing two particular links e^- and e^+ , where $e^- \in \delta^-(v)$ and $e^+ \in \delta^+(v)$. When node v fails, the flows on these paths can be consistently thinned using the value of $z_{e^-e^+d}^s$ instead of $z_{e'd}^s$ in a formula analogous to (EC.17). A similar observation applies to variables Z .

For a fixed $d \in \mathcal{D}$, an elementary path-flow pattern $x_{dp}^0 > 0, p \in \mathcal{P}_d$ (for some $\mathcal{P}_d \subseteq \hat{\mathcal{P}}_d$), that realizes the link-flows $x_{ed}^0, e \in \mathcal{E}$, and is consistent with the values of variables w can be found recursively as follows. Initially, we set \mathcal{P}_d equal to \emptyset , and find a (elementary) path $q = (v_0, e_1, v_1, e_2, \dots, e_{n-1}, v_{n-1}, e_n, v_n)$ in $\hat{\mathcal{P}}_d$, where $v_0 = o(d)$ and $v_n = t(d)$, such that the values of all variables $x_{e_1d}^0, w_{e_1e_2d}^{v_1}, w_{e_2e_3d}^{v_2}, \dots, w_{e_{n-1}e_nd}^{v_{n-1}}, x_{e_nd}^0$ are strictly greater than 0. Then we define $x_{dq}^0 := \min\{x_{e_1d}^0, w_{e_1e_2d}^{v_1}, w_{e_2e_3d}^{v_2}, \dots, w_{e_{n-1}e_nd}^{v_{n-1}}, x_{e_nd}^0\}$, subtract the flow x_{dq}^0 from variables $x_{e_1d}^0, w_{e_1e_2d}^{v_1}, w_{e_2e_3d}^{v_2}, \dots, w_{e_{n-1}e_nd}^{v_{n-1}}, x_{e_nd}^0$, add the so found path q to \mathcal{P}_d , and repeat the procedure until no such path q for d can be found. This observation justifies the correctness of formulation (EC.26).

It turns out that the LBP formulation becomes exact when it has loop-less optimal solutions and each node can fail at most once (i.e., $|\mathcal{S}(v)| \leq 1, v \in \mathcal{V}$). This is because only then any feasible loop-less solution of (14) determines a feasible solution of (EC.26) with the same cost. (Note that this is true for an arbitrary single-link failure scenario.) For single-node failure scenarios admitting $|\mathcal{S}(v)| > 1$ for some $v \in \mathcal{V}$, the above observation is not true anymore because of constraint (EC.26h) which implies that for each value of $w_{e^-e^+}^v$ (for v, e^-, e^+, d fixed) must be in this case an upper bound for the set of values $z_{e^-e^+}^s, s \in \mathcal{S}(v)$ with more than one element. This is illustrated by the following example.

EXAMPLE EC.5. Consider the network depicted in Figure EC.4 with 4 links $e = 1, 2, 3, 4$. The unit capacity cost of all four links is equal to 1. The single demand (with demand volume h) between nodes o and t can be realized on 4 paths: $p_1 = \{1, 3\}, p_2 = \{1, 4\}, p_3 = \{2, 3\}, p_4 = \{2, 4\}$. We consider a single-node failure scenario with 4 states $s = 1, 2, 3, 4$ characterized by the sets of failing links $\mathcal{E}_1 = \{2, 4\}, \mathcal{E}_2 = \{2, 3\}, \mathcal{E}_3 = \{1, 4\}, \mathcal{E}_4 = \{1, 3\}$. (Note that the four availability states can be considered as four different failures of node v .) The availability coefficients of all the failing links in all the states are equal to α . The demand has the nominal volume and the volumes in the states all equal to h . Note that in each state $s = 1, 2, 3, 4$ only path p_s remains unaffected.

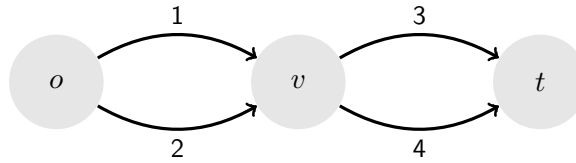


Figure EC.4 A network example

It is clear that in terms of link capacity the optimal solution of the lower bound formulation (14) is unique and symmetric. Hence, a simple cut-based argumentation implies that the common optimal value y^0 of the link capacity must fulfil the equality $y^0 + \alpha y^0 = h$, that is, $y^0 = \frac{h}{1+\alpha}$. The common value x^0 for the optimal nominal link-flows is equal to y^0 . In each state, the optimal link-flows on the links with reduced capacity are equal to $\frac{h\alpha}{1+\alpha}$, while on the fully available links—to $\frac{h}{1+\alpha}$. Hence, the optimal LBP objective function value is equal to $C' = \frac{4h}{1+\alpha}$.

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

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

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

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

We note that for the single-node failures, the relation between the exact formulation (EC.26) and the lower bound formulation (14) in the undirected case is the same as for the directed case.

EC.7. Comparing LBP and FTOP

The solution of formulation (14) is a lower bound for FTOP because all constraints of (14) are obviously fulfilled by the link-flows defined by any feasible solution $x_{dp}^0, x_{dp}^s, d \in \mathcal{D}, p \in \hat{\mathcal{P}}_d, s \in \mathcal{S}$ of (1), that is, by the link-flows

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

Conversely, an optimal solution $x_{ed}^0, x_{ed}^s, e \in \mathcal{E}, d \in \mathcal{D}, s \in \mathcal{S}$, of (14) could be translated to an optimal solution of FTOP only if there existed elementary path-flows that would realize the nominal link-flows x_{ed}^0 , and at the same time could be appropriately thinned to realize the state-dependent link-flows x_{ed}^s for each state s .

However, this is in general not possible and hence optimal LBP solutions may in this sense be infeasible for FTOP already for the single-link failure scenarios (Section EC.5.2, Example EC.4) and for the single-node failure scenarios (Section EC.6.2, Example EC.5).

We evaluated the lower bound of FTOP resulting from compact formulation (14) for state scenarios SL, DL and SN (single-node failures) for the four cases of (α, β) considered in Section 8.1.1. As the tests were made for undirected networks, we implemented a version of formulation (14) with undirected links.

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

A comparison of LBP and FTOP in terms of the network cost and the running time assuming DL is presented in Tables EC.2 and EC.3 for *polska* and *nobel-us*, respectively. The tables show the ratios C_{FTOP}/C_{LBP} and T_{FTOP}/T_{LBP} . The quantities C_{FTOP} and C_{LBP} are the optimal costs

for FTOP and for LBP, respectively, while T_{FTOP} and T_{LBP} are the respective running times. The results indicate that for the DL case the LBP cost is only marginally smaller than the FTOP cost, and that LBP becomes extremely time consuming.

Table EC.2 LBP vs. FTOP for *polska* under DL.

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

Table EC.3 LBP vs. FTOP for *nobel-us* under DL.

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

Finally let us address the question raised in Section 6: why we consider FT instead of an LBP-based strategy (i.e., a kind of link-flow thinning) despite the fact that in general solutions of FTOP are strictly more expensive than solutions of LBP. The reason is three-fold.

First, as discussed earlier in this section, in general the state-dependent link-flows optimized with LBP do not correspond to any path-flows (tunnels) that obey the assumptions of FT. Thus, the simple tunnel thinning and traffic control mechanism that with FT is performed at the traffic/tunnel originating nodes cannot be directly applied. Certainly, a link-flow thinning mechanism could be applied at the link originating nodes—this, however, would most likely lead to a substantially more complicated traffic control mechanism as compared to FT.

Second, as shown below, the cost advantage of LBP over FTOP is only marginal. In addition, except for small networks the running time required to solve LBP is significantly larger than that required by the PG algorithm of FTOP. The reason is as follows. Assuming the common case $|\mathcal{D}| \sim |\mathcal{V}|^2$ and $|\mathcal{E}| \sim |\mathcal{V}|$, in the node-link formulation (14) the number of variables and constraints is of the order of $O(|\mathcal{V}||\mathcal{D}||\mathcal{S}|) = O(|\mathcal{V}|^3|\mathcal{S}|)$. At the same time, for a given set of admissible paths \mathcal{P} with $|\mathcal{P}_d| \leq P, d \in \mathcal{D}$ (where P is bounded and does not increase with the size of the network), the FTOP formulation (1) requires $O(P|\mathcal{D}||\mathcal{S}|) = O(|\mathcal{V}|^2|\mathcal{S}|)$ variables and constraints, that is $|\mathcal{V}|$ times less.

Third, as shown in Example EC.4, in general the optimal nominal link-flow x^0 delivered by LBP must contain loops. This can cause serious problems for traffic routing protocols.

EC.8. Uncertainty polytopes

EC.8.1 Budgeted uncertainty polytope with controlled relaxation

The objective of this appendix is to generalize the FT uncertainty polytope from Section 7.3 to arbitrary linear programs subject to uncertainty. Hence, in what follows we consider a linear constraint

$$\sum_{i \in \mathcal{I}} w_i x_i \leq b, \tag{EC.27}$$

where $\mathcal{I} := \{1, 2, \dots, I\}$, $x \in \mathbb{R}^I$ is a vector of optimization variables, and $w \in \mathbb{R}^I$ is a vector of coefficients, being the parameter defining instances of (EC.27). Robust optimization considers uncertainty imposed by parameter w by assuming that w can take any value in a given compact convex (uncertainty) set \mathcal{S} . Thus, in robust optimization, constraint (EC.27) is replaced by the infinite set of constraints

$$\sum_{i \in \mathcal{I}} w_i x_i \leq b, \quad w \in \mathcal{S}.$$

This framework was extended by Ben-Tal et al. (2006) who proposed to control also what happens when the uncertain parameter w takes values outside the uncertainty set \mathcal{S} . Namely, they use a larger set, \mathcal{S}_+ , that contains all “physically possible” realizations of the uncertain parameter, defined as the sum of \mathcal{S} and a closed convex cone \mathcal{C} , $\mathcal{S}_+ := \{w' = w + \lambda : w \in \mathcal{S}, \lambda \in \mathcal{C}\}$, and they allow for a controlled violation of the constraint when w takes values in \mathcal{S}_+ :

$$\sum_{i \in \mathcal{I}} w_i x_i \leq b + \gamma \|w, \mathcal{S}\|, \quad w \in \mathcal{S}_+,$$

where $\gamma \geq 0$ is a given sensibility parameter and $\|w, \mathcal{S}\|$ denotes the distance of w from \mathcal{S} .

Here we propose an alternative construction for \mathcal{S}_+ . Rather than building \mathcal{S}_+ from general convex sets \mathcal{S} and \mathcal{C} , we focus on extending the budgeted uncertainty polytope introduced in Bertsimas and Sim (2004), remarkably popular in the fields of integer programming, combinatorial optimization and network optimization. The budgeted uncertainty polytope assumes that each parameter w_i can deviate from its nominal value \bar{w}_i by a value not greater than \hat{w}_i and that at most N of the parameters can simultaneously take extreme values. Recent papers by Büsing and D’Andreagiovanni (2012) and Mattia (2012) have independently proposed an extension of such polytopes where more parameters playing the role of N are used. Such extended budgeted uncertainty polytopes are defined as follows.

Suppose $\mathcal{K} := \{1, 2, \dots, K\}$ and assume a given finite set $\mathcal{W}_i := \{\hat{w}_i^1, \hat{w}_i^2, \dots, \hat{w}_i^K\}$ of deviations levels for each parameter w_i , and a finite set $\mathcal{N} := \{N^1, N^2, \dots, N^K\}$ of positive integers that satisfy conditions $1 \leq N^k \leq |\mathcal{I}|$, $k \in \mathcal{K}$ and $\sum_{k \in \mathcal{K}} N^k \geq |\mathcal{I}|$. Using an approach similar to the one from Section 7.3, we first introduce an auxiliary polytope \mathcal{S}' that is defined as the set of all vectors $u = (u^+, u^-)$ for $u^+, u^- \in \mathbb{R}_+^{I \times K}$ fulfilling the conditions

$$\sum_{k \in \mathcal{K}} (u_i^{+k} + u_i^{-k}) \leq 1, \quad i \in \mathcal{I}; \quad \sum_{i \in \mathcal{I}} (u_i^{+k} + u_i^{-k}) \leq N^k, \quad k \in \mathcal{K}.$$

The uncertainty polytope \mathcal{S}^B from Mattia (2012) is then obtained through the affine mapping

$$w(u)_i = \bar{w}_i + \sum_{k \in \mathcal{K}} \hat{w}_i^k (u_i^{+k} - u_i^{-k}), \quad i \in \mathcal{I}.$$

The asymmetric polytope from Büsing and D'Andreagiovanni (2012) is defined analogously, by considering distinct positive and negative deviations levels.

For our construction, we suppose that the above set \mathcal{S}^B has been defined to be as large as possible so that it contains all physically possible realizations of the uncertain parameter w . Hence \mathcal{S}^B plays the role of set \mathcal{S}_+ introduced by Ben-Tal et al. (2006). Since enforcing constraint $\sum_{i \in \mathcal{I}} w_i x_i \leq b$ for all $w \in \mathcal{S}^B$ would be too conservative, we select a subset of \mathcal{S}^B for which the constraint must hold strictly. Outside of the subset, we allow a controlled violation of $\sum_{i \in \mathcal{I}} w_i x_i \leq b$. This is achieved by considering b as another uncertain parameter whose reference value is \bar{b} , and introducing a finite set $\{\hat{b}^1, \hat{b}^2, \dots, \hat{b}^K\}$ of its nonnegative deviations. The robust constraint becomes

$$\sum_{i \in \mathcal{I}} w_i x_i \leq b, \quad (w, b) \in \mathcal{S},$$

where \mathcal{S} is obtained through the following (linear) mappings:

$$w(u)_i = \bar{w}_i + \sum_{k \in \mathcal{K}} \hat{w}_i^k (u_i^{+k} - u_i^{-k}), \quad i \in \mathcal{I}; \quad b(u) = \bar{b} + \sum_{k \in \mathcal{K}} \hat{b}^k U^k, \quad (\text{EC.28})$$

where $U^k = \sum_{e \in \mathcal{I}} (u_i^{+k} + u_i^{-k})$, $k \in \mathcal{K}$. The subset of indices of \mathcal{K} whose deviations \hat{b}^k are equal to 0 corresponds to the classical uncertainty set for which the constraint must hold firmly. The remaining indices correspond to an extension of the uncertainty set, \mathcal{S}^B , for which some controlled relaxation of the constraint is permitted.

We finally note that the polytope defined through (EC.28) leads to a linear programming formulation, being a straightforward generalization of formulation (22).

EC.8.2 Uncertainty polytopes for single, double, and n -link failures

Observe that in polytope $\mathcal{S}'(\mathcal{N})$ defined by (18) the vertices are binary. This is because constraints (18) are implied by a coefficient matrix identical to the one used for the assignment problem constraints. The latter matrix is known to be totally unimodular (Nemhauser and Wolsey 1988).

This actually means that for each vertex (α, h) of the uncertainty polytope $\mathcal{S}(\mathcal{N})$ defined by (19), $\alpha_e \in \{a^1, a^2, \dots, a^K\}$, $e \in \mathcal{E}$. Because of that, we can model the single-link failure scenarios and, more generally, n -link failure scenarios, as the vertices of the corresponding, appropriately defined uncertainty polytopes.

Single-link failures (with availability coefficient a) are modeled by assuming $\mathcal{K} = \{1, 2\}$ with $a^1 = a, a^2 = 1$ and $N^1 = 1, N^2 = |\mathcal{E}| - 1$. The so defined polytope $\mathcal{S}(\mathcal{N})$ contains all the single-link failure states but not the nominal state. For each single-link failure, $h_d = h_d^0(1 - b^1)$, $d \in \mathcal{D}$. If we wished to include the nominal state into $\mathcal{S}(\mathcal{N})$ then we would change N^2 to $|\mathcal{E}|$ and then for the nominal state ($U^1 = 0, U^2 = |\mathcal{E}|$), the vector h would be equal to h^0 .

The double-link failures case has several subcases. The simplest is when we assume the same failure level a for both failing links. Then we put $\mathcal{K} = \{1, 2\}$ with $a^1 = a, a^2 = 1$ and $N^1 = 2, N^2 = |\mathcal{E}| - 2$. The so defined polytope $\mathcal{S}(\mathcal{N})$ contains all double-link failure states, but not the nominal state nor the single-link failures. For each double-link failure, $h_d = h_d^0(1 - 2b^1), d \in \mathcal{D}$. If we wished to include single-link failures (with availability level a) then we would change N^2 to $|\mathcal{E}| - 1$. Then for the single-link failures ($U^1 = 1, U^2 = |\mathcal{E}| - 1$), the vector h would be equal to $h^0(1 - b^1)$. Moreover, by putting $N^2 = |\mathcal{E}|$ we would include the nominal state to $\mathcal{S}(\mathcal{N})$ as well. Finally, when we assume two different failure levels, $a^1 < a^2$, for the two failing links we put $\mathcal{K} = \{1, 2, 3\}$, $0 < a^1 < a^2 < a^3 = 1$, and $N^1 = N^2 = 1$ and $N^3 = |\mathcal{E}| - 2$. This setting makes $\mathcal{S}(\mathcal{N})$ contain only double-link failures with $h = h^0(1 - b^1 - b^2)$. Adding all single-link failures (with the demand requirement dependent on the availability level of the failing link) and the nominal state can be done analogously.

Note that when both nominal state and the double-link failure states belong to the uncertainty polytope, then, necessarily, also all single-link failures must belong to it. Hence, if we do not wish to consider single-link failures, we need to treat the nominal state, and for that matter any other state outside the considered uncertainty polytope, in an individual way, as discussed after Proposition 4 in Section 7.4.

The n -link failures case (with $n \geq 3$) can be treated analogously. It is important that when we decide to include all the k -link failures (with $k < n$) into the constructed polytope, then all the i -link failures (with $k < i < n$) will be included automatically.

EC.8.3 Proof of Proposition 4

In order to derive the reformulation stated in Proposition 4 we will make use of the dual formulations of linear programs defined over $\mathcal{S}(\mathcal{N})$, i.e., linear programs from the class

$$\max_{(\alpha, h) \in \mathcal{S}(\mathcal{N})} F_c(\alpha, h), \quad (\text{EC.29})$$

where $F_c(\alpha, h) := \sum_{\bar{e} \in \mathcal{E}} c'_{\bar{e}} \alpha_{\bar{e}} + \sum_{d \in \mathcal{D}} c''_d h_d$ for any given vector $c \in \mathbb{R}^{|\mathcal{E}| + |\mathcal{D}|}$, $c = (c', c'')$, $c' \in \mathbb{R}^{|\mathcal{E}|}$, $c'' \in \mathbb{R}^{|\mathcal{D}|}$.

Lemma 6 *The optimal objective value F_c^* of (EC.29) is bounded and equal to $W^* + \sum_{d \in \mathcal{D}} c''_d h_d^0$, where W^* is the optimal objective value of the following linear program in variables $\tau := (\tau_{\bar{e}}, \bar{e} \in \mathcal{E})$ and $\theta := (\theta_k, k \in \mathcal{K})$:*

$$\min W = \sum_{\bar{e} \in \mathcal{E}} \tau_{\bar{e}} + \sum_{k \in \mathcal{K}} N^k \theta_k \quad (\text{EC.30a})$$

$$\tau_{\bar{e}} + \theta_k \geq c'_{\bar{e}} a^k - \sum_{d \in \mathcal{D}} c''_d h_d^0 b^k, \quad \bar{e} \in \mathcal{E}, k \in \mathcal{K} \quad (\text{EC.30b})$$

$$\theta \geq 0. \quad (\text{EC.30c})$$

Proof. Clearly, F_c^* is bounded since $S(\mathcal{N})$ is bounded and nonempty. Now we note that the considered problem (EC.29) can equivalently be reformulated as:

$$\max_{u \in S'(\mathcal{N})} F_c(u), \quad (\text{EC.31})$$

where $S'(\mathcal{N})$ is defined by (18), and the objective function

$$F_c(u) := \sum_{\bar{e} \in \mathcal{E}} c'_e \left(\sum_{k \in \mathcal{K}} a^k u_{\bar{e}}^k \right) + \sum_{d \in \mathcal{D}} c''_d \left(h_d^0 \left(1 - \sum_{k \in \mathcal{K}} b^k \left(\sum_{\bar{e} \in \mathcal{E}} u_{\bar{e}}^k \right) \right) \right) \quad (\text{EC.32})$$

is obtained using (20a) and (20b). After transformation we get

$$F_c(u) = \sum_{d \in \mathcal{D}} c''_d h_d^0 + \sum_{\bar{e} \in \mathcal{E}} \sum_{k \in \mathcal{K}} (c'_e a^k - \sum_{d \in \mathcal{D}} c''_d h_d^0 b^k) u_{\bar{e}}^k, \quad (\text{EC.33})$$

where the first term on the right-hand side is constant. Hence, instead of (EC.31) we can consider its modification

$$\max \quad F'_c(u) = \sum_{\bar{e} \in \mathcal{E}} \sum_{k \in \mathcal{K}} (c'_e a^k - \sum_{d \in \mathcal{D}} c''_d h_d^0 b^k) u_{\bar{e}}^k \quad (\text{EC.34a})$$

$$[\tau_{\bar{e}}] \quad \sum_{k \in \mathcal{K}} u_{\bar{e}}^k = 1, \quad \bar{e} \in \mathcal{E} \quad (\text{EC.34b})$$

$$[\theta_k \geq 0] \quad \sum_{\bar{e} \in \mathcal{E}} u_{\bar{e}}^k \leq N^k, \quad k \in \mathcal{K} \quad (\text{EC.34c})$$

$$u \geq 0. \quad (\text{EC.34d})$$

It is easy to check that (EC.30), with the dual variables specified on the left-hand sides of constraints (EC.34b) and (EC.34c), defines the problem dual to (EC.34), and this proves Lemma 6. \square

Proof of Proposition 4 Denote the polyhedron in the (τ, θ) -space $\mathbb{R}^{|\mathcal{E}|+|\mathcal{K}|}$ defined by (EC.30b) and (EC.30c) for a given vector $c = (c', c'') \in \mathbb{R}^{|\mathcal{E}|+|\mathcal{D}|}$ with $D(c)$, and define expressions

$$f(\tau, \theta, c) := \sum_{\bar{e} \in \mathcal{E}} \tau_{\bar{e}} + \sum_{k \in \mathcal{K}} N^k \theta_k + \sum_{d \in \mathcal{D}} c''_d h_d^0. \quad (\text{EC.35})$$

Let $A(\mathcal{N})$ denote the projection of the uncertainty polytope $S(\mathcal{N})$ onto the α -space $\mathbb{R}^{|\mathcal{E}|}$. Formulation (1), modified for $S := S(\mathcal{N})$ and the affine decision rules $x_{dp}^s = z_{dp}^0 + \sum_{\bar{e} \in \mathcal{E}_{dp}} z_{dp}^{\bar{e}} \alpha_{\bar{e}}^s$, $\alpha \in A(\mathcal{N})$, is as follows:

$$\min \quad C = \sum_{e \in \mathcal{E}} \xi_e y_e^0 \quad (\text{EC.36a})$$

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} x_{dp}^0 \leq y_e^0, \quad e \in \mathcal{E} \quad (\text{EC.36b})$$

$$\sum_{p \in \mathcal{P}_d} (z_{dp}^0 + \sum_{\bar{e} \in \mathcal{E}_{dp}} z_{dp}^{\bar{e}} \alpha_{\bar{e}}) \geq h_d, \quad d \in \mathcal{D}, \quad (\alpha, h) \in S(\mathcal{N}) \quad (\text{EC.36c})$$

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} (z_{dp}^0 + \sum_{\bar{e} \in \mathcal{E}_{dp}} z_{dp}^{\bar{e}} \alpha_{\bar{e}}) \leq \alpha_e y_e^0, \quad e \in \mathcal{E}, \alpha \in A(\mathcal{N}) \quad (\text{EC.36d})$$

$$z_{dp}^0 + \sum_{\bar{e} \in \mathcal{E}_{dp}} z_{dp}^{\bar{e}} \alpha_{\bar{e}} \leq x_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, \alpha \in A(\mathcal{N}) \quad (\text{EC.36e})$$

$$z_{dp}^0 + \sum_{\bar{e} \in \mathcal{E}_{dp}} z_{dp}^{\bar{e}} \alpha_{\bar{e}} \geq 0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, \alpha \in A(\mathcal{N}) \quad (\text{EC.36f})$$

$$x^0, y^0 \text{ continuous and nonnegative, } z \text{ continuous.} \quad (\text{EC.36g})$$

Assuming $z_{dp}^{\bar{e}} := 0, \bar{e} \in \mathcal{E} \setminus \mathcal{E}_{dp}$, we can transform (EC.36) to a more convenient form:

$$\min \quad C = \sum_{e \in \mathcal{E}} \xi_e y_e^0 \quad (\text{EC.37a})$$

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} x_{dp}^0 \leq y_e^0, \quad e \in \mathcal{E} \quad (\text{EC.37b})$$

$$\sum_{\bar{e} \in \mathcal{E}} (-\sum_{p \in \mathcal{P}_d} z_{dp}^{\bar{e}}) \alpha_{\bar{e}} + h_d \leq \sum_{p \in \mathcal{P}_d} z_{dp}^0, \quad d \in \mathcal{D}, (\alpha, h) \in S(\mathcal{N}) \quad (\text{EC.37c})$$

$$\sum_{\bar{e} \in \mathcal{E}} (\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} z_{dp}^{\bar{e}} - \Delta(\bar{e}, e) y_e^0) \alpha_{\bar{e}} \leq -\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} z_{dp}^0, \quad e \in \mathcal{E}, \alpha \in A(\mathcal{N}) \quad (\text{EC.37d})$$

$$\sum_{\bar{e} \in \mathcal{E}} z_{dp}^{\bar{e}} \alpha_{\bar{e}} \leq x_{dp}^0 - z_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, \alpha \in A(\mathcal{N}) \quad (\text{EC.37e})$$

$$\sum_{\bar{e} \in \mathcal{E}} (-z_{dp}^{\bar{e}}) \alpha_{\bar{e}} \leq z_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, \alpha \in A(\mathcal{N}) \quad (\text{EC.37f})$$

$$x^0, y^0 \text{ continuous and nonnegative, } z \text{ continuous.} \quad (\text{EC.37g})$$

Now, due to Lemma 6, we can substitute the left-hand side of each of the constraints (EC.37c)–(EC.37f) by an appropriate expression (EC.35) and an appropriate constraint (EC.30b). In detail, the substitutions are as follows.

Constraint (EC.37c): $c^1 := (c', c''), c' := (-\sum_{p \in \mathcal{P}_d} z_{dp}^{\bar{e}}, \bar{e} \in \mathcal{E}), c'' := (\Delta(d, \bar{d}), \bar{d} \in \mathcal{D})$

$$f(\tau_d^1, \theta_d^1, c^1) \leq \sum_{p \in \mathcal{P}_d} z_{dp}^0, \quad d \in \mathcal{D} \quad (\text{EC.38a})$$

$$(\tau_d^1, \theta_d^1) \in \text{D}(c^1), \quad d \in \mathcal{D}. \quad (\text{EC.38b})$$

Constraint (EC.37d): $c^2 := (c', c''), c' := (\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} z_{dp}^{\bar{e}} - \Delta(\bar{e}, e) y_e^0, \bar{e} \in \mathcal{E}), c'' := 0$

$$f(\tau_{\bar{e}}^2, \theta_{\bar{e}}^2, c^2) \leq -\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} z_{dp}^0, \quad e \in \mathcal{E} \quad (\text{EC.39a})$$

$$(\tau_{\bar{e}}^2, \theta_{\bar{e}}^2) \in \text{D}(c^2), \quad \bar{e} \in \mathcal{E}. \quad (\text{EC.39b})$$

Constraint (EC.37e): $c^3 := (c', c''), c' := (z_{dp}^{\bar{e}}, \bar{e} \in \mathcal{E}), c'' := 0$

$$f(\tau_{dp}^3, \theta_{dp}^3, c^3) \leq x_{dp}^0 - z_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d \quad (\text{EC.40a})$$

$$(\tau_{dp}^3, \theta_{dp}^3) \in \text{D}(c^3), \quad d \in \mathcal{D}, p \in \mathcal{P}_d. \quad (\text{EC.40b})$$

Constraint (EC.37f): $c^A := (c', c'')$, $c' := (-z_{dp}^{\bar{e}}, \bar{e} \in \mathcal{E})$, $c'' := 0$

$$f_{c^A}(\tau_{dp}^A, \theta_{dp}^A, c^A) \leq z_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d \quad (\text{EC.41a})$$

$$(\tau_{dp}^A, \theta_{dp}^A) \in D(c^A), \quad d \in \mathcal{D}, p \in \mathcal{P}_d. \quad (\text{EC.41b})$$

These substitutions lead directly to formulation (22). \square

We finally note that we can explicitly extend formulation (22) for a given finite set of states \mathcal{S} , $\mathcal{S} \cap \mathcal{S}(\mathcal{N}) = \emptyset$. This is achieved by adding the following constraints to formulation (22).

$$\sum_{p \in \mathcal{P}_d} (z_{dp}^0 + \sum_{\bar{e} \in \mathcal{E}_{dp}} z_{dp}^{\bar{e}} \alpha_{\bar{e}}^s) \geq h_d, \quad d \in \mathcal{D}, s \in \mathcal{S} \quad (\text{EC.42a})$$

$$z_{dp}^0 + \sum_{\bar{e} \in \mathcal{E}_{dp}} z_{dp}^{\bar{e}} \alpha_{\bar{e}}^s \leq x_{dp}^0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, s \in \mathcal{S} \quad (\text{EC.42b})$$

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \delta_{edp} (z_{dp}^0 + \sum_{\bar{e} \in \mathcal{E}_{dp}} z_{dp}^{\bar{e}} \alpha_{\bar{e}}^s) \leq \alpha_e y_e^0, \quad e \in \mathcal{E}, s \in \mathcal{S} \quad (\text{EC.42c})$$

$$z_{dp}^0 + \sum_{\bar{e} \in \mathcal{E}_{dp}} z_{dp}^{\bar{e}} \alpha_{\bar{e}}^s \geq 0, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, s \in \mathcal{S}. \quad (\text{EC.42d})$$

EC.9. Number of paths generated by the PG algorithm

Figure EC.5 illustrates an important aspect of the PG algorithm, namely the evolution of the number of added paths and the network cost for the consecutive iterations. Two α cases for SL with $\beta = 1$ are considered for *pdh* and *polska*. For all these cases the final optimum is obtained after only 5-7 PG iterations.

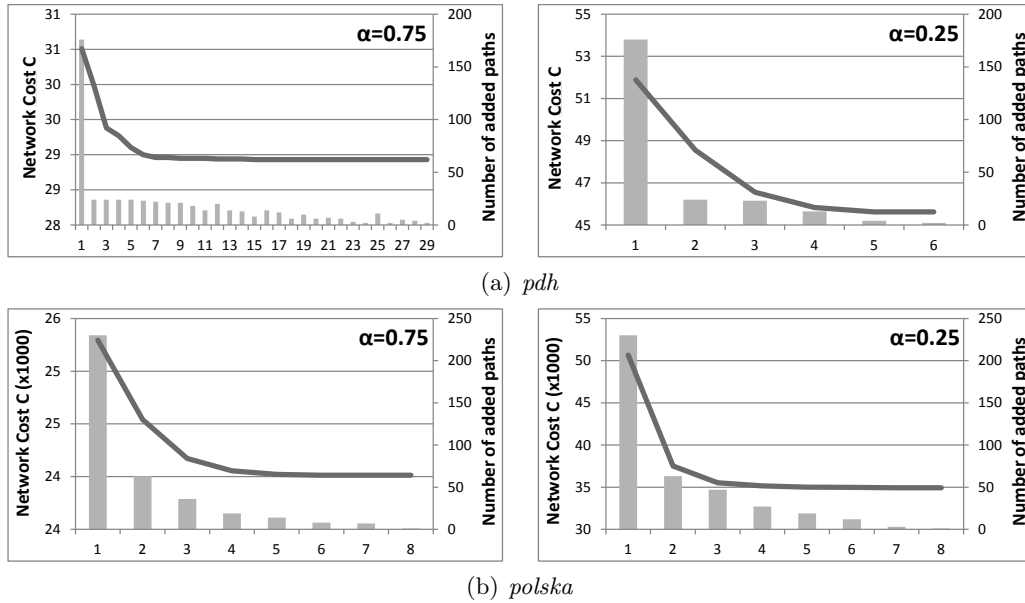


Figure EC.5 Value of the objective function and number of paths added at each iteration.