



HAL
open science

Path Generation for Affine Flow Thinning

Michal Pióro, Ilya Kalesnikau, Michael Poss

► **To cite this version:**

Michal Pióro, Ilya Kalesnikau, Michael Poss. Path Generation for Affine Flow Thinning. Electronic Notes in Discrete Mathematics, 2018, 64, pp.355 - 364. 10.1016/j.endm.2018.02.010 . hal-01768634

HAL Id: hal-01768634

<https://hal.science/hal-01768634>

Submitted on 30 Mar 2021

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

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

Path Generation for Affine Flow Thinning

Michał Pióro^{a,b,1,2} Ilya Kalesnikau^{a,1,3} Michael Poss^{c,4}

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

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

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

Abstract

Flow thinning (FT) is a traffic routing and protection strategy for communication networks whose links experience fluctuations in available capacity (as, e.g., in wireless networks). To cope with this phenomenon, end-to-end traffic demands are assigned dedicated tunnels (e.g., MPLS tunnels) whose nominal capacity is subject to thinning in order to account for variable capacity of the links, fluctuating below their nominal values. Consequently, the instantaneous traffic sent between the demand's end nodes must accommodate to the current total capacity available on its dedicated tunnels. In the paper we present a path generation-based approach to network dimensioning for a practical version of FT applying affine flow thinning – a fairly complicated issue not yet considered. We derive a relevant pricing problem and present a numerical study illustrating efficiency of the optimization algorithm.

Keywords: Survivable networks, variable link capacity, linear and mixed-integer programming, multicommodity flows, affine routing, path generation.

1 Introduction

Flow thinning (FT), a concept introduced in [6], is 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. Each end-to-end traffic demand is assigned a set of logical tunnels whose total capacity is dedicated to carry the demand's

¹ This work was supported by NCN (Poland) [grant 2015/17/B/ST7/03910].

² Email: michal.pioro@tele.pw.edu.pl

³ Email: ikalesni@mion.elka.pw.edu.pl

⁴ Email: michael.poss@lirmm.fr

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. FT (a generalization of the *demand-wise shared protection strategy* [2,8] devised for total link failures) is relevant, among other applications, for wireless mesh networks utilizing MPLS tunnels.

The multicommodity flow optimization problem for FT is \mathcal{NP} -hard and its non-compact linear programming formulation requires path generation (PG) (see [1,4]) using a pricing problem to find the paths to be added to the problem in the consecutive steps of the PG algorithm. In this paper we consider an *affine* version of FT, referred to as AFT, where the capacity of each tunnel is thinned according to a tunnel-dependent linear function whose arguments are link failure coefficients. In particular, we study the pricing problem (PP) for AFT – the issue not investigated in [6]. Our motivation is that PP for the affine case turns out to be more complicated than its counterpart presented in [6] for the general form of FT where thinning is not restricted. In the next sections we formulate the affine version of the FT optimization problem, derive the relevant PP, and present a numerical study illustrating efficiency of the PG algorithm, and effectiveness of the considered version of affine FT as compared to the effectiveness of FT.

2 Notation

The (undirected) network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is composed of the set of nodes \mathcal{V} and the set of links \mathcal{E} , where each link represents an undirected pair $\{v, w\}$ of nodes for some $v, w \in \mathcal{V}$, $v \neq w$. We mention here that considerations of the paper are applicable to directed networks as well.

The nominal capacity of link $e \in \mathcal{E}$ is denoted by y_e^0 and its unit capacity cost by $\xi(e)$ (cost of the network is given by $C = \sum_{e \in \mathcal{E}} \xi(e)y_e^0$). Yet, the capacity of links is variable, and typically not all the links reach their maximum (nominal) capacity simultaneously. Thus, at any time instant, nominal link capacities 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. To consider this, we specify a set of link availability states \mathcal{S} . Each state $s \in \mathcal{S}$ is characterized by the link *availability coefficients* $\alpha(e, s)$ ($0 \leq \alpha(e, s) \leq 1$), $e \in \mathcal{E}$. By definition, the capacity of link e available in state s is equal to $y_e^s := \alpha(e, s)y_e^0$. In the

sequel we use link *failure coefficients* $\beta(e, s) := 1 - \alpha(e, s)$ instead of alphas.

Traffic demands are represented by the set \mathcal{D} . Each demand $d \in \mathcal{D}$ is characterized by an undirected pair $\{o(d), t(d)\}$ composed of the demand's end nodes (origin and termination), and given traffic volumes $h(d, s)$ to be realized in the considered link availability states $s \in \mathcal{S}$. Traffic volumes and link capacities are expressed in the same units. Each demand d is assigned a set of admissible paths $\mathcal{P}(d)$ (called the path-list) composed of selected elementary paths between $o(d)$ and $t(d)$. Paths in $\mathcal{P}(d)$, used to realize the demand (traffic) volumes, are assigned nominal 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 between $o(d)$ and $t(d)$, will be denoted by $\hat{\mathcal{P}}(d)$, $d \in \mathcal{D}$, with $\hat{\mathcal{P}} := \bigcup_{d \in \mathcal{D}} \hat{\mathcal{P}}(d)$.

Since we assume elementary paths, the paths can 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 $\delta(e, d, p)$, $e \in \mathcal{E}$, $d \in \mathcal{D}$, $p \in \mathcal{P}_d$, where $\delta(e, d, p) = 1$ if $e \in p$, and $\delta(e, d, p) = 0$ otherwise. 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.

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

3 ATOP – Affine Thinning Optimization Problem

For given admissible path-lists $\mathcal{P}(d)$, $d \in \mathcal{D}$, the considered problem (ATOP) is denoted by $P(\mathcal{P})$ (where $\mathcal{P} = \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$). ATOP can be formulated as the following link-path linear programme (LP) involving variables x_{dp}^0 ($d \in \mathcal{D}$, $p \in \mathcal{P}(d)$), x_{dp}^s ($d \in \mathcal{D}$, $p \in \mathcal{P}(d)$, $s \in \mathcal{S}$), z_{dp}^e ($d \in \mathcal{D}$, $p \in \mathcal{P}(d)$, $e \in \mathcal{E}$) and y_e^0 ($e \in \mathcal{E}$). In the formulation, $\mathcal{E}(d, p) \subseteq \mathcal{E}$ is a (predefined) subset of links

whose failure coefficients influence the linear (affine to be precise) thinning formula for path $p \in \mathcal{P}(d)$. Dual variables indicated in the square brackets will be used in (4).

Problem P(\mathcal{P}):

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

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

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

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

$$[\varphi_{dp}^s] x_{dp}^s = x_{dp}^0 - \sum_{e \in \mathcal{E}(d,p)} \beta(e, s) z_{dp}^e, \quad d \in \mathcal{D}, \quad p \in \mathcal{P}(d), \quad s \in \mathcal{S} \quad (1e)$$

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

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

Note that constraint (1d) is, for each link $e \in \mathcal{E}$, written down only for $s \in \mathcal{S}(e)$, and not for all $s \in \mathcal{S}$. This is because for any state $s \in \mathcal{S} \setminus \mathcal{S}(e)$ (in which the capacity of the considered link e is fully available, $\alpha(e, s) = 1$) constraint (1d) is implied by (1b) and (1e). Note also that in general feasibility of ATOP is assured only if $\alpha(e, s) > 0$ for all $e \in \mathcal{E}$ and $s \in \mathcal{S}$.

The thinning formula (1e) can be generalized in a natural way to:

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

where all z_{dp}^e are continuous and *unconstrained in sign* (note that z_{dp}^0 provides an additional “degree of freedom” in the thinning formula), and $x_{dp}^s \leq x_{dp}^0$, $d \in \mathcal{D}$, $p \in \mathcal{P}_d$, $s \in \mathcal{S}$. As discussed in [6], typical examples of the sets $\mathcal{E}(d, p)$ are $\mathcal{E}(d, p) = p$, $\mathcal{E}(d, p) = \bigcup_{v \in \mathcal{V}(p)} \delta(v)$ (where $\mathcal{V}(p)$ is the set of nodes traversed by path p , and $\delta(v)$ is the set of links incident with v), and $\mathcal{E}(d, p) = \mathcal{E}$.

Finally, the most general thinning is achieved when only the restriction

$$x_{dp}^s \leq x_{dp}^0, \quad d \in \mathcal{D}, \quad p \in \mathcal{P}(d), \quad s \in \mathcal{S} \quad (3)$$

is assumed and no affine formula as (2) is applied. The so obtained generalization of ATOP was studied in [6] under the name FTOP.

We end this section with giving the formulation dual to ATOP (1):

Problem D(\mathcal{P}):

$$W(\mathcal{P}) = \max \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} h(d, s) \lambda_d^s \quad (4a)$$

$$\pi_e^0 + \sum_{s \in \mathcal{S}(e)} \alpha(e, s) \pi_e^s \leq \xi(e), \quad e \in \mathcal{E} \quad (4b)$$

$$\sum_{s \in \mathcal{S}} \varphi_{dp}^s \leq \sum_{e \in \mathcal{E}} \delta(e, d, p) \pi_e^0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (4c)$$

$$\lambda_d^s \leq \varphi_{dp}^s + \sum_{e \in \mathcal{E}(s)} \delta(e, d, p) \pi_e^s, \quad s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}(d) \quad (4d)$$

$$\sum_{s \in \mathcal{S}} \beta(e, s) \varphi_{dp}^s \geq 0, \quad d \in \mathcal{D}, p \in \mathcal{P}(d), e \in \mathcal{E}(d, p) \quad (4e)$$

$$\pi^0, \pi^s, \lambda^s \geq 0 \text{ and continuous; } \varphi \text{ continuous.} \quad (4f)$$

4 Path Generation and the Pricing Problem

The link-path LP formulation (1) of ATOP is noncompact because of exponentially many routing paths in the maximum path-list $\hat{\mathcal{P}}$. Although the complexity of ATOP is open as yet, we may suspect that the problem is difficult, taking into account that we are not aware of its compact LP formulations, and that FTOP – the generalization of ATOP assuming (3) instead of (1e) – is \mathcal{NP} -hard [6]. Since in practice formulation (1) cannot be solved directly (as we are not able to include all elementary paths in the path-lists), to consider all paths in $\hat{\mathcal{P}}$ we need to apply path generation (PG, see [1,4,7]) – a classical technique in multicommodity flow networks related to column generation in linear programming (cf. [3]). 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 variables and constraints to the problem formulation (i.e., we generate new columns and rows in (1)). This is done by solving an appropriate pricing problem (PP) using, as parameters, optimal dual variables, i.e., an optimal solution (λ^*, π^*) of the current dual D(\mathcal{P}).

PP for ATOP turns out to be more complicated than its FTOP counterpart derived in Section 4 of [6]. PP for ATOP is derived below.

Consider a given demand $d \in \mathcal{D}$ and a given path $q \in \hat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$. Introducing a new path $q \in \hat{\mathcal{P}}(d) \setminus \mathcal{P}(d)$ to the dual (4) introduces both new (we call them *local* in the sequel) dual variables φ^s , $s \in \mathcal{S}$, and new constraints (4c)–(4e). It can happen that some of the new constraints (4c)–(4e) will be violated by the optimal dual solution (λ^*, π^*) of D(\mathcal{P}) for any values

of the local variables. In such a case some of the new constraints will separate (λ^*, π^*) from the dual polyhedron of $D(\mathcal{P} \cup \{q\})$. The sum of such violations is expressed through the quantity $P(q)$ defined by the following LP.

$$P(q) = \min Z + \sum_{s \in \mathcal{S}} Y_s + \sum_{e \in \mathcal{E}(d, q)} X_e \quad (5a)$$

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

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

$$[b_e \geq 0] \quad X_e \geq - \sum_{s \in \mathcal{S}} \beta(e, s) \varphi^s, \quad e \in \mathcal{E}(d, q) \quad (5d)$$

$$Z \in \mathbb{R}_+; Y_s \in \mathbb{R}_+, s \in \mathcal{S}; X_e \in \mathbb{R}_+, e \in \mathcal{E}(d, q); \varphi^s \in \mathbb{R}, s \in \mathcal{S}, \quad (5e)$$

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

Since we are interested in maximizing the value of $P(q)$, we cannot use a minimization objective as in (5a). Therefore we consider the problem dual to (5), formulated using the dual variables specified in the square brackets in constraints (5b)-(5d) (in (6) dual variables a_s , $s \in \mathcal{S}$, are eliminated):

$$P(q) = \max -g|q|^0 + \sum_{s \in \mathcal{S}} (g - \sum_{e \in \mathcal{E}(d, q)} \beta(e, s) b_e) (\lambda_d^{s*} - |q|^s) \quad (6a)$$

$$g \leq 1; b_e \leq 1, e \in \mathcal{E}(d, q); \sum_{e \in \mathcal{E}(d, q)} \beta(e, s) b_e \leq g, s \in \mathcal{S} \quad (6b)$$

$$g \in \mathbb{R}_+; b_e \in \mathbb{R}_+, e \in \mathcal{E}. \quad (6c)$$

PP consists in finding, for each $d \in \mathcal{D}$, a path q that maximizes (6a). PP is derived from (6) by introducing binary variables u_e , $e \in \mathcal{E}$, and constraints (7b) (so that the set $\{e \in \mathcal{E} : u_e = 1\}$ specifies a path), and by eliminating bi-linearities $b_e u_e$, $g u_e$, $b_e u_e u_{e'}$ that will appear. For undirected networks and $\mathcal{E}(d, p) = p$, the resulting PP is as follows (below the superscript $*$ in π^* , λ^* is skipped, and $\delta(v)$ denotes the set of links incident to $v \in \mathcal{V}$):

$$\max - \sum_{e \in \mathcal{E}} \pi_e^0 G_e + (\sum_{s \in \mathcal{S}} \lambda_d^s) g - \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}} \pi_e^s G_e + \sum_{s \in \mathcal{S}} \sum_{e' \in \mathcal{E}} \sum_{e \in \mathcal{E}} \pi_{e'}^s \beta(e, s) T_{e'e} - \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}} \lambda_d^s \beta(e, s) B_e \quad (7a)$$

$$\sum_{e \in \delta(v)} u_e = 1, v \in \{o(d), t(d)\}; \sum_{e \in \delta(v)} u_e = 2z_v, v \in \mathcal{V} \setminus \{o(d), t(d)\} \quad (7b)$$

$$g \leq 1; b_e \leq 1, e \in \mathcal{E}; \sum_{e \in \mathcal{E}} \beta(e, s) B_e \leq g, s \in \mathcal{S} \quad (7c)$$

$$B_e \leq u_e, B_e \leq b_e, B_e \geq b_e + u_e - 1, e \in \mathcal{E} \quad (7d)$$

$$G_e \leq g, G_e \leq u_e, G_e \geq g + u_e - 1, e \in \mathcal{E} \quad (7e)$$

$$T_{e'e} \leq u_{e'}, T_{e'e} \leq u_e, T_{e'e} \leq b_e, T_{e'e} \geq b_e + u_{e'} + u_e - 2, e', e \in \mathcal{E} \quad (7f)$$

$$g \in \mathbb{R}_+; b_e, B_e, G_e \in \mathbb{R}_+, e \in \mathcal{E}; T_{e'e} \in \mathbb{R}_+, e, e' \in \mathcal{E} \quad (7g)$$

$$u_e \in \{0, 1\}, e \in \mathcal{E}; z_v \in \{0, 1\}, v \in \mathcal{V} \setminus \{o(d), t(d)\}. \quad (7h)$$

An optimal solution u^* of (7) defines the path $q := \{e \in \mathcal{E} : u_e^* = 1\}$ that should be added to ATOP (provided $P(q) > 0$).

Note that any feasible set of links $\mathcal{U} := \{e \in \mathcal{E} : u_e = 1\}$ will necessarily contain an elementary path between $o(d)$ and $t(d)$ and, possibly, a set of disjoint isolated loops. The path is elementary because variables $z_v, v \in \mathcal{V}$, are binary (note that assuming these variables to be integer-valued would allow paths to contain loops as well). Such loops may appear in optimal PP solutions for several reasons. One of them is that there is a tradeoff related to the number of elements in the set \mathcal{U} : the more elements the more flexible formulae (1e), and the less elements the lower link capacities.

To eliminate the (isolated) loops we introduce the bi-directed version $\mathcal{G}' = (\mathcal{V}, \mathcal{A})$ of the original undirected network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of (directed) arcs is defined as $\mathcal{A} := \{(v, w), (w, v) : \{v, w\} \in \mathcal{E}\}$, i.e., each undirected link $e \in \mathcal{E}$ is substituted by two oppositely directed arcs $a'(e)$ and $a''(e)$. Below, $\delta^-(v)$ denotes the set of all arcs incoming to node v , and $\delta^+(v)$ – the set of all arcs outgoing from node v ($v \in \mathcal{V}$). Adding aggregated arc-flow variables $f_a \geq 0$, $a \in \mathcal{A}$, and the following constraints to (7) will eliminate the loops in question.

$$\sum_{a \in \delta^-(t(d))} f_a = \sum_{v \in \mathcal{V} \setminus \{o(d), t(d)\}} z_v \quad (8a)$$

$$\sum_{a \in \delta^+(o(d))} f_a = \sum_{a \in \delta^-(o(d))} f_a \quad (8b)$$

$$\sum_{a \in \delta^+(v)} f_a = \sum_{a \in \delta^-(v)} f_a + z_v, \quad v \in \mathcal{V} \setminus \{o(d), t(d)\} \quad (8c)$$

$$f_{a'(e)} + f_{a''(e)} \leq (|\mathcal{V}| - 2)u_e, \quad e \in \mathcal{E}. \quad (8d)$$

In the next section we will also consider the case $\mathcal{E}(d, p) = \mathcal{E}$, $d \in \mathcal{D}, p \in \mathcal{P}(d)$. Then formulation (7) becomes a bit simpler since variables B_e become equal to b_e , and variables $T_{e'e}$ express the product $u_{e'} \cdot b_e$ rather than $u_{e'} \cdot u_e \cdot b_e$.

5 Numerical Study

Below we present numerical results for a network with 12 nodes, 18 (undirected) links and 66 (undirected) demands. The network is fully described in SNDlib (sndlib.zib.de, see [5]) under the name *polska*. We use the data from the first instance (i.e., *polska*-D-B-M-N-C-A-N-N) and assume the link unit capacity cost ξ_e equal to its `<module_cost>` divided by its `<module_capacity>` for the first pair of these values. Thus, for the first link (between Gdansk and Warsaw) $\xi_1 = 156/155 = 1.0645$, and the entire vector ξ of the unit capacity costs is as follows: 1.0645, 1.7548, 1.0064, 1.2000, 1.7548, 1.5290, 1.3419,

1.1677, 1.3419, 1.6129, 2.0903, 2.0903, 1.6129, 1.0645, 1.9677, 0.9161, 1.2581, 1.8968 (the order of links is the same as in the SNDlib file).

In Table 1 we present the results illustrating time efficiency of the PG algorithm for ATOP and FTOP, and cost efficiency of ATOP as compared with FTOP (referred to as F-T). Two cases of ATOP are considered: A-S with $\mathcal{E}(d, p) = p$, and A-F with $\mathcal{E}(d, p) = \mathcal{E}$. All the reported calculations were executed on a PC-class computer (Windows 10 64-bit, 8 GB RAM, Processor Intel Core i5-3210M, 4 logical processors, 2.5GHz) using CPLEX 12.4.0.0.

Table 1 shows the results for three failure scenarios (in all of them, the failing links loose half of their capacity, i.e, if link e fails in state s , its link availability coefficient is $\alpha(e, s) = 0.5$):

- Single link (SL) failure scenario $\mathcal{S}1$ contains the state $s(0)$ (all links fully available), and all the states with exactly one link failing (note that $|\mathcal{S}1| = |\mathcal{E}| + 1$). 100% traffic protection is assumed so that $h(d, s) := h(d)$, $d \in \mathcal{D}$, $s \in \mathcal{S}1$, and $h(d)$ are the demand volumes described in SNDlib.
- Double link (DL) failure scenario $\mathcal{S}2$ contains $\mathcal{S}1$ (with no reduction of demand volumes) and all the states with exactly two links failing ($|\mathcal{S}2| = |\mathcal{S}1| + |\mathcal{E}|(|\mathcal{E}| - 1)/2$). For the states in $\mathcal{S}2 \setminus \mathcal{S}1$ the demand volumes are reduced to 95% of the volumes assumed for the states in $\mathcal{S}1$.
- Triple link (TL) failure scenario $\mathcal{S}3$ contains $\mathcal{S}2$, and all the states with exactly three links failing ($|\mathcal{S}3| = |\mathcal{S}2| + |\mathcal{E}|(|\mathcal{E}| - 1)(|\mathcal{E}| - 2)/6$). For $\mathcal{S}2$ the demand reductions are as in DL, and for $\mathcal{S}3 \setminus \mathcal{S}2$ the demand volumes are reduced to 90% of the $\mathcal{S}1$ volumes.

For PG, the initial path list contains one path for each demand – the shortest path with respect to the link unit costs (66 paths in total). Note that assigning the entire demand volume $h(d, s(0))$ for each demand $d \in \mathcal{D}$ to such a shortest path will result in the optimally dimensioned network when protection is not considered. For the examined network example this cost is equal to 30275.5.

In Table 1 the columns presenting the results describe: initial cost (C^0), final cost (C^*), cost increase of ATOP with respect to FTOP (ΔC), final number of paths $|\mathcal{P}^*|$, number of iterations (iter), total computation time (total), time per iteration (t/iter), percentage of the total computation time spent in the master problem (t/MP), and percentage of the total computation time spent in the pricing problem (t/PP). It should be noted that the computational time for A-S is usually greater than for A-F due to the fact that the PP corresponding to A-S contains additional binary variables with respect to the PP corresponding to A-F because $\mathcal{E}(d, p) = p$ for A-S and $\mathcal{E}(d, p) = \mathcal{E}$ for A-F. For the same reason, the network cost achieved for A-F is always less

Table 1
PG algorithm results for *polska*

		C^0	C^*	ΔC	$ \mathcal{P}^* $	iter	total	t/iter	t/MP	t/PP
SL	F-T	60551	40236	-	259	9	6s	0.6s	9.1%	90.9%
	A-F	60551	40236	0	274	9	2m3s	14s	0.7%	99.3%
	A-S	60551	40236	0	276	9	2m22s	16s	0.4%	99.6%
DL	F-T	60551	44496	-	278	8	1m2s	8s	61.6%	38.4%
	A-F	60551	56521	27%	194	10	2m37s	16s	7.3%	92.7%
	A-S	60551	56521	27%	309	10	4m20s	26s	8.2%	91.8%
TL	F-T	60551	48951	-	297	9	30m59s	3m27s	81.8%	18.2%
	A-F	60551	58609	19%	165	6	5m22s	54s	42.1%	57.9%
	A-S	60551	58609	19%	228	9	12m35s	1m24s	25.2%	74.8%

than or equal to that for A-S. (In the reported cases these costs are equal but this is not the case in general). Finally, let us note that, again for the same reason, A-S needs to be considered as its potential network implementation is much simpler than for A-F.

A general conclusion from the presented study is that the PG process is pretty effective, which is especially important with respect to PP – a MIP formulation of a potentially hard optimization problem.

6 Comments and Concluding Remarks

Although affine flow thinning (AFT) requires more link capacity than the original version (FT) with no constraints on thinning (except (3)), the affine versions are important for three reasons. First, as FT, AFT must be aware of the current link states $\alpha(\cdot, s)$; yet it will use the (optimized) affine functions for thinning the nominal flows. Such thinning, contrary to FT, is applicable to all possible states, 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 of [6]).

Second, AFT in its version with the thinning dependent only on the failure coefficients β along the path realizing the given flow (and for that matter, in the version taking into account the failure coefficients of all the links e belonging to the sets $\delta(v)$ for all the nodes v traversed by the considered path) is much more practical than FT in the network implementation aspect. In such a case disseminating link capacity information is, as discussed in Section 9.2 of [6], not a major issue, as the path-originating node thins the corresponding

path-flow only on the basis of the current links' capacities along the path.

Third, from the computational efficiency viewpoint, the main interest in AFT stems from its capability of handling large sets \mathcal{S} of link availability states, provided \mathcal{S} is given as the set of vertices (extreme points) of a polytope (called *uncertainty polytope*) described by a polynomial number of variables and constraints. This is explained in Section 7.3 of [6].

In the paper we have discussed an important issue in optimization of affine thinning – path generation. We have derived an appropriate formulation of the pricing problem and shown by numerical examples that the path-generation resolution process of the basic optimization problem is quite effective. An issue left for future studies is path generation for the general affine thinning case defined by (2) (which requires less link capacity than the two versions considered in Table 1), and also for the case when the link availability states are specified by means of the uncertainty polytope.

References

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.
- [2] A.M.C.A. Koster, A. Zymolka, M. Jäger, and R. Hülserman. Demand-wise shared protection for meshed optical networks. *Journal of Network and System Management*, 13(1):35–55, 2005.
- [3] M. Minoux. *Mathematical Programming: Theory and Algorithms*. John Wiley & Sons, 1986.
- [4] S. Orlowski and M. Pióro. Complexity of column generation in network design with path-based survivability mechanisms. *Networks*, 59(1):132–147, 2012.
- [5] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski. SNDlib 1.0 – survivable network design library. *Networks*, 55(3):276–286, 2010.
- [6] M. Pióro, Y. Fouquet, D. Nace, and M. Poss. Optimizing flow thinning protection in multicommodity networks with variable links capacity. *Operations Research*, 64(2):273–289, 2016.
- [7] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan-Kaufman, 2004.
- [8] R. Wessäly, S. Orlowski, A. Zymolka, A.M.C.A. Koster, and C.G. Gruber. Demand-wise shared protection revisited: A new model for survivable network design. In *Proc. 2nd International Network Optimization Conference (INOC 2005)*, pages 100–105, 2005.