An optimization model for robust FSO network dimensioning

Dritan Nace, Michal Pióro, Michael Poss, Fabio D'andreagiovanni, Ilya Kalesnikau, Marinela Shehaj, Artur Tomaszewski

An optimization model for robust FSO network dimensioning

Dritan Nace[a,*], Michał Pióro[b], Michael Poss[c], Fabio D'Andreagiovanni[d,a], Ilya Kalesnikau[b], Marinela Shehaj[a], Artur Tomaszewski[b]

[a]*Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253, Compiègne, France*
[b]*Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland*
[c]*UMR CNRS 5506 LIRMM, Université de Montpellier, 161 rue Ada, Montpellier, France*
[d]*French National Center for Scientific Research (CNRS), France*

Abstract

FSO (Free Space Optics) is a well established wireless optical transmission technology considered as an alternative to radio communications, for example in metropolitan wireless mesh networks. An FSO link is established by means of a laser beam connecting the transmitter and the receiver placed in the line of sight. A major disadvantage of FSO links (with respect to fiber links) is their sensitivity to weather conditions such as fog, rain and snow, causing substantial loss of the transmission power over the optical channel due mostly to absorption and scattering. Thus, although the FSO technology allows for fast and low-cost deployment of broadband networks, its operation will be affected by this sensitivity, manifested by substantial losses in links' transmission capacity with respect to the nominal capacity. Therefore, a proper approach to FSO network dimensioning should take such losses into account so that the level of carried traffic is satisfactory under all observed weather conditions. In the paper we describe such an approach. We introduce a relevant dimensioning problem and present a robust optimization algorithm for such enhanced dimensioning. A substantial part of the paper is devoted to present a numerical study of two FSO network instances that illustrates the promising effectiveness of the proposed approach.

Keywords: Resilient and survivable networks, free space optics, variable link capacity, linear and mixed-integer programming, robust optimization, multicommodity flows.

1. Introduction

The considerations of this paper are devoted to dimensioning communication networks resilient with respect to multiple partial link failures. The main application area we have in mind are networks that apply Free Space Optics (FSO) – a well established broadband wireless optical transmission technology where the communication links are provided by means of a laser beam sent from the transmitter to the receiver placed in the line of sight.

*Corresponding author
Email addresses: dritan.nace@hds.utc.fr (Dritan Nace), m.pioro@tele.pw.edu.pl (Michał Pióro), michael.poss@lirmm.fr (Michael Poss), d.andreagiovanni@hds.utc.fr (Fabio D'Andreagiovanni), i.kalesnikau@tele.pw.edu.pl (Ilya Kalesnikau), mshehaj@hds.utc.fr (Marinela Shehaj), a.tomaszewski@tele.pw.edu.pl (Artur Tomaszewski)

FSO links are considered as an alternative to radio links for example in metropolitan Wireless Mesh Networks (WMN).

FSO networks exhibit several important advantages: transmission range of several kilometers, high transmission bandwidth, secure communication, quick and easy deployment, lower cost (as compared with the fiber optical technology), immunity to electromagnetic interference, license-free long-range operation (contrary to radio communications). A disadvantage is vulnerability of the FSO links to weather conditions such as fog, rain, and snow (and pollution, for that matter), causing substantial loss of the transmission power over optical channel, mostly due to absorption and scattering. This makes the problem of network dimensioning important, and, as a matter of fact, difficult.

Typically, a given weather condition affects a subset of the network links and each affected link looses a portion (fraction) of its nominal capacity (i.e., of the bit-rate realized when weather is fine). Such a fraction is called the link failure (degradation) ratio that can assume some value between 0 (no capacity loss) and 1 (total capacity loss). As we will see in Section 2, typical values of the link failure ratio are $0.25, 0.50$ and 1. In consequence, each particular weather condition that may occur defines a failure state referred to as the multiple partial link failure state.

The above discussion motivates the network optimization problem studied in this paper: how to dimension the network links at the lowest cost and at the same time assure the traffic demand satisfaction at an acceptable level for all observed/predicted weather conditions. As we will demonstrate, the problem can be approached by the methods of robust optimization [1].

The impact of weather conditions on the wireless networks transmission capacity has been studied to some extent, but the majority of works have considered failure modeling for a single region failure (some investigations for the case of weather perturbations occurring simultaneously in different regions can be found in [2]). In particular, paper [3] discusses three measures of the WMN survivability for a regional failure scenario assuming circular failure areas with random location of failure epicenters. As observed in [3], from the network topology viewpoint, networks with nodes covering a square area in a regular way have a better performance in terms of the total traffic surviving a regional failure. It is also shown that in the case of heavy rain storms, using information on forecasted attenuation of links based on radar measurements allows for periodic updates of the network topology in advance, and thus dealing efficiently with these phenomena. In [4], a mixed-integer programming (MIP) model for network reconfiguration in case of unfavorable weather conditions is presented. The model looks for alternative routes for rerouting some of the disturbed traffic, while reducing the interference between adjacent transmission links. Other studies, such as [5], have considered the relevance and impact of specific weather factors on the FSO links capacity. In Section 2, we will present a short study on such an impact in terms of simultaneously affected FSO links.

As already mentioned, the paper introduces a (new) optimization model for robust dimensioning of FSO networks in order to achieve resilience with respect to multiple partial link failures. Similar problems have been widely investigated in the literature but for a given list of failure states, see for instance [6] and the list of literature therein. More recently, robust optimization approaches to resilient network dimensioning have been developed in [7, 8, 9] (under traffic demand uncertainty), and in [10] (under signal propagation uncertainty in

wireless networks). Yet, to the best of our knowledge, no models have been developed for uncertainty concerning available capacity on transmission links, i.e., for the case relevant to networks employing FSO. Also, as majority of work on traffic restoration and resilient network dimensioning has been done for wired networks, multiple partial link failures are virtually not considered despite common appearance of this phenomenon in modern wireless networks that cope with it by means of adjustable modulation and coding schemes (MCS). Multiple partial link failures were addressed in [6] (for the so called Unrestricted Reconfiguration mechanism, commonly called Global Rerouting [11]), and more recently in [12] (for the so called oblivious routing) and in [13] (see below). An optimization approach, based on disjoint path routing, to FSO networks with multiple partial failures is presented in [14].

Paper [13], along with [15], are the starting point for this work. In those two papers we have studied a specific traffic protection mechanism, called *Flow Thinning* (FT). The main feature of FT is handling partial failures without any traffic rerouting at all. In FT, each traffic demand is equipped with a set of dedicated routing paths whose flows are thinned accordingly to the current availability of network link capacity (for more details on FT see [13]). Although the context of the present study and of [13] and [15] is the same, as we deal with cost-efficient dimensioning of communication networks resilient to multiple link failures, the following differences main can be observed. First, the current paper assumes the Global Rerouting mechanism instead of FT. (Recall that Global Rerouting allows for establishing path flows in an unrestricted way from scratch when a failure state occurs – this feature makes the cheapest possible mechanism traffic protection mechanism in terms of the cost of link capacity.) Next, in the present study we focus on specific features of the FSO networks such as full-duplex links, realistic link failure scenarios, etc. Finally, now we concentrate on estimating the reference failure set by means of an approximate compact set of failure states, which opens a way to a considerable reduction of computation time.

The contribution of this work consists in studying a network dimensioning problem taking into account resilience with respect to multiple partial link failures. In contrast to the previous investigations, we deal with the problem involving a very large number of failure states, described by the so called uncertainty set (an instance of the budgeted uncertainty set formally described in [1]) for which we propose an optimization model together with a cut generation solution algorithm, and use it for a numerical study on robust dimensioning of a specific FSO network.

The paper is organized as follows. In Section 2 we discuss the issues of modeling weather conditions based on meteorological records for a given time period, and the associated link capacity degradation. In Section 3 we present a mathematical formulation of the robust network dimensioning problem that is central to our investigations. Section 4 develops a solution approach to the problem based on a cut generation algorithm, and is followed by by Section 5 discussing computational complexity of the considered optimization problem. Then, in Section 6, modifications and enhancements of our solution approach are discussed. Next, in Section 7, results of extensive numerical studies illustrating the efficiency of our approach to robust FSO network dimensioning are presented. Finally, in Section 8, we give concluding remarks and outline the future work.

2. Modeling of link availability states

As already mentioned, the optimization approach developed in this paper is motivated by the necessity of dealing with degradation of FSO links caused by weather conditions. Just like radio technology, FSO suffers from a major limitation – sensitivity of the data transmission quality to weather conditions. In fact, in case of bad weather conditions, the FSO data transmission quality may be severely affected if no special actions are undertaken. A solution is to control the current signal modulation and coding scheme (MCS) applied at the transmitter of the affected FSO link to secure correct data transmission. This, however, results in decreasing effective transmission data rates. In the sequel, the fraction of the maximal attainable (i.e., nominal) link capacity (bit-rate) lost due to a change of its MCS is called *link failure ratio*.

In the paper a link represents a set of parallel FSO transmission systems. If not stated explicitly otherwise, the systems are assumed to be full duplex, so that a system between node v and node w is composed of two pairs of transceivers. One such pair provides transmission over a light beam from v to w, and the other over a light beam from w to v. Each such light beam carries the data with a given nominal bit-rate of M Gbps, which is typically of the order of 1–20 Gbps [16, 17]. The quantity M is called *link capacity module*.

In our considerations we assume four operation modes for each link. The first (basic) mode realizes the nominal link capacity, and is applied in normal (good) weather conditions using the 16-QAM (quadrature-amplitude modulation, see [18]) MCS. The second mode corresponds to weather conditions worse than normal when the scheme is switched to 4-QAM – this assures approximately 75% of the nominal link capacity (failure ratio 0.25). When the weather conditions are even more degraded, the third mode is applied by switching to the QPSK scheme (quadrature phase-shift keying, see [19]) scheme, losing approximately 50% of the nominal link capacity (failure ratio 0.5). The last mode corresponds to harsh weather conditions that make correct data transmission impossible (failure ratio 1).

In order to consider link capacity degradation in optimization modeling, we need to estimate the link failure ratios for various weather conditions that can occur during the network lifetime. Having distinguished the four operation modes, we can determine what is the proper operation mode for each FSO link in a given weather state (note that in general different links see different weather conditions in a given weather state). To do this we apply the formulae given in [20] and deduce the vector of the link failure ratios for each of the considered weather states. The set of all such (different) vectors constitutes the *reference failure set* denoted by $\mathcal{S}$. In our state modeling methodology we analyse the weather records observed over a time period, typically over one year. For each hourly period (out of $365 \times 24 = 8760$ periods) we translate the observed weather conditions into the link failure ratios – in effect each of the hourly period is assigned its link failure ratio vector. That is, when at a given hourly period some geographical area is affected by bad weather, the corresponding (link availability) state is characterized by assigning appropriate failure ratios to the links in this area, for example the ratios equal to 0.25 or 0.5, depending on the degree of degradation the affected links experience. Next, we identify the subset of periods with different link failure ratio vectors to form the so called reference failure set (denoted with $\mathcal{S}$ below). The so formed reference failure sets are used in the numerical study presented in Section 7.

In the paper we also consider a special kind of virtual failure sets, called K-sets (see Section 4.3), parameterized by an integer value K, where K is less than or equal to the number of all links in the network. For a given K, the K-set contains all states corresponding to all combinations of K, or less, simultaneously affected links. It is additionally assumed that when a link is affected, it is always affected with the same failure ratio. For example, the set of single link failure states frequently considered in the literature is a specific case of K-set, where $K = 1$ and the link failure ratio is set to 1 for all links. In our optimization approach the K-sets are used as an approximation of the reference failure set $\mathcal{S}$. The approach consists in using K-sets (for a set of selected values of parameter K) as an input (instead of the true reference failure set) to the robust network dimensioning problem formulated in Section 3, and then testing the so obtained optimal link capacities (called the robust solution) on the true reference failure state. Clearly, the larger the K value, the more severe failure states the K-set contains, resulting in general in more robust (and at the same time likely more costly) solutions.

3. Problem specification

3.1. Notation

The considered network is modeled by means of a bi-directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{E})$ composed of the set of nodes $\mathcal{V}$, the set of (directed) arcs $\mathcal{A} \subseteq \mathcal{V}^2 \setminus \{(v, v) : v \in \mathcal{V}\}$, and the set of (bi-directed) links $\mathcal{E} \subseteq \mathcal{V}^{|2|}$ (where $\mathcal{V}^{|2|}$ denotes the family of all 2-element subsets of the set of nodes $\mathcal{V}$).

Since the graph is bi-directed, each arc $a = (v, w) \in \mathcal{A}$ has its oppositely directed counterpart $a' = (w, v) \in \mathcal{A}$. For a given arc $a = (v, w)$, its originating node v will be denoted by $o(a)$ and its terminating node by $t(a)$, i.e., $o(a) = v$ and $t(a) = w$. Furthermore, for each node $v \in \mathcal{V}$ we define $\delta^+(v) := \{a \in \mathcal{A} : o(a) = v\}$ (the set of arcs outgoing from node v) and $\delta^-(v) := \{a \in \mathcal{A} : t(a) = v\}$ (the set of links incoming to node v). In the sequel, the number of nodes $|\mathcal{V}|$ and the number of arcs $|\mathcal{A}|$ will be denoted by V and A, respectively.

To each pair of two oppositely directed arcs $a = (v, w), a' = (w, v)$ there corresponds a bi-directed link $e = \{v, w\} \in \mathcal{E}$ denoted by $e(a)$ (or $e(a')$). Thus, the set of links is defined as $\mathcal{E} := \{e(a) : a \in \mathcal{A}\}$. The two oppositely directed arcs corresponding to link e will be denoted by $a(e)$ and $a'(e)$, hence for $e = \{v, w\}$, $a(e) = (v, w)$ and $a'(e) = (w, v)$. Clearly, the number of links $E = |\mathcal{E}|$ is two times less than the number of arcs: $A = 2E$.

Capacity of links is modular and each module provides the bit-rate of M Gbps separately for each of the two oppositely directed arcs related to the link – this assumption corresponds to full duplex transmission. The capacity of a link is composed of several modules. When the link capacity is optimized, the number of modules installed on link $e \in \mathcal{E}$ is denoted by y_e. Thus, y_e is a non-negative integer variable whose value specifies the link capacity equal to My_e. When the link capacity is fixed, it is denoted by $c(e)$, where $c(e) = Mn(e)$ for some fixed non-negative integer $n(e)$. As a consequence, each of the two oppositely directed arcs of link e has capacity My_e or $Mn(e)$. To each link $e \in \mathcal{E}$ there corresponds a non-negative module cost $\xi(e)$ (a parameter). Hence, the total cost of the network is given by $\sum_{e\in\mathcal{E}} \xi(e)y_e$ or $\sum_{e\in\mathcal{E}} \xi(e)n(e)$. In the sequel, notation $y := (y_e, e \in \mathcal{E})$ and $c := (c(e), e \in \mathcal{E})$ will be used.

Traffic demands are represented by the set $\mathcal{D}$ (the number of demands $|\mathcal{D}|$ will be denoted by D). Each demand $d \in \mathcal{D}$ is represented by an ordered pair of nodes $(o(d), t(d))$ (its origin and termination) and the volume $h(d)$ (a parameter) that has to be realized from $o(d)$ to $t(d)$. The demand volumes and link capacity modules are expressed in the same units.

The set of network links is subject to multiple partial failures (link degradations). A particular configuration of such degradations is referred to a network failure state, and the set of failure states is denoted $\mathcal{S}$ (the number of states $|\mathcal{S}|$ is denoted by S). When a link is degraded then it loses a fraction of its capacity specified by a failure ratio (between 0 and 1) from a given set of possible failure ratios $\mathcal{R}$, for example $\mathcal{R} = \{0, 0.25, 0.5, 1\}$. In consequence, each failure state $s \in \mathcal{S}$ can be identified with a vector of the link failure ratios $\beta(s) = (\beta(e, s),\ e \in \mathcal{E})$, where $\beta(e, s) \in \mathcal{R}$ is the fraction of capacity lost on link e in state s. Thus, in state $s \in \mathcal{S}$ the available capacity of link $e \in \mathcal{E}$ is equal to $\big(1 - \beta(e, s)\big)My_e$ (or $\big(1 - \beta(e, s)\big)c(e)$).

Finally, we assume that in any failure state $s \in \mathcal{S}$ each demand $d \in \mathcal{D}$ can be routed in a bifurcated way along all possible paths from $o(d)$ to $t(d)$, and the resulting traffic flows in particular states are independent of each other. This means that the network applies the so called *Global Rerouting* or *Unrestricted Reconfiguration* mechanism [6, 11].

3.2. Problem formulation

The problem considered in the paper is formulated as a mixed-integer program (MIP) in the node-arc notation (using arc-flow variables $x := (x^s_{ad},\ a \in \mathcal{A}, d \in \mathcal{D}, s \in \mathcal{S})$ and the link capacity variables y). The notation is summarized in Table 1.

parameters	**description**
$\mathcal{V}$	set of nodes v ($v \in \mathcal{V}$)
$\mathcal{A}$	set of (directed) arcs a ($a \in \mathcal{A}$)
$\mathcal{E}$	set of (bi-directed) links e ($e \in \mathcal{E}$, composed of two oppositely directed arcs)
$\delta^+(v)(\delta^-(v))$	set of arcs outgoing from (incoming to) node $v \in \mathcal{V}$
$a(e)$, $a'(e)$	two oppositely directed arcs corresponding to link $e \in \mathcal{E}$
$e(a)$	bi-directed link e corresponding to arc $a \in \mathcal{A}$
$c(e)$	capacity of link $e \in \mathcal{E}$ (when fixed and given)
M	capacity of the link capacity module
$\xi(e)$	cost of one capacity module on link $e \in \mathcal{E}$
$\mathcal{D}$	set of traffic demands d ($d \in \mathcal{D}$)
$\mathcal{S}$	set of failure states s ($s \in \mathcal{S}$)
$\mathcal{R}$	given set of possible failure ratios ($\mathcal{R} = \{0, 0.25, 0.5, 1\}$)
$\beta(e, s)$	fraction of capacity lost on link $e \in \mathcal{E}$ in state $s \in \mathcal{S}$ ($\beta(e, s) \in \mathcal{R}$)
variables	**description**
x^s_{ad}	flow carrying traffic of demand d on arc a in state s (no-negative continuous)
y_e	number of modules installed on link e (non-negative integer)
λ^v_d, π_a	dual variables

Table 1: Basic parameters and variables of the optimization model.

Note that in this paper we use the following convention for vector notation. When a represents a vector of given parameters, then $a = (a(1), a(2), \ldots, a(n))$ (so that a denotes an n-element vector, and $a(i)$ is one of its elements). Yet, when a denotes a vector of variables, then $a = (a_1, a_2, \ldots, a_n)$. Alternatively, when the set of indices is represented by $\mathcal{N}$ (i.e., $\mathcal{N} = \{1, 2, \ldots, n\}$), then we write $a = (a(i),\ i \in \mathcal{N})$ or $a = (a_i,\ i \in \mathcal{N})$, respectively.

The formulation of the the optimization problem in question is as follows.

Problem P($\mathcal{S}$) (main problem in node-arc formulation):

$$C(\mathcal{S}) = \min \textstyle\sum_{e\in\mathcal{E}} \xi(e) y_e \tag{1a}$$

$$\textstyle\sum_{a\in\delta^+(v)} x^s_{ad} - \sum_{a\in\delta^-(v)} x^s_{ad} = \begin{cases} h(d), & \text{if } v = o(d) \\ -h(d), & \text{if } v = t(d), \\ 0, & \text{otherwise} \end{cases} \quad d \in \mathcal{D}, v \in \mathcal{V}, s \in \mathcal{S} \tag{1b}$$

$$\textstyle\sum_{d\in\mathcal{D}} x^s_{ad} \le \big(1 - \beta(e(a), s)\big) M y_{e(a)}, \quad a \in \mathcal{A},\ s \in \mathcal{S} \tag{1c}$$

$$x \ge 0 \text{ and continuous};\ y \ge 0 \text{ and integer}. \tag{1d}$$

Above, x^s_{ad} is the flow on arc a dedicated to carry the traffic of demand d in state s. The objective of P($\mathcal{S}$), i.e., minimizing the cost of links, is specified in (1a). Constraint (1b) represents the flow conservation equation for each demand d at each node v in each state s, assuring the realization of $h(d)$ for all d in each state s. Finally, the capacity constraint (1c) ensures that the capacity of link e is not exceeded in any state s.

Because of modular links, problem P($\mathcal{S}$) is $\mathcal{NP}$-hard even for a polynomial number of states in set $\mathcal{S}$ (see [6]). Moreover, the direct approach to P($\mathcal{S}$) requires solving linear programs involving a large number of arc-flow variables x (linear relaxations of (1) assuming continuous y) during the branch-and-bound process performed by the MIP solver. For this reason, we have developed a more efficient approach to P($\mathcal{S}$). The approach, based on Benders' decomposition [21], is presented in the next sections.

4. Solving P($\mathcal{S}$) by cut generation

4.1. Feasibility testing

Let $\mathcal{Y}$ denote the set of all capacity vectors feasible for (1) and suppose we wish to test whether a given capacity vector $c = (c(e),\ e \in \mathcal{E})$ is in $\mathcal{Y}$. Because the demand routing in a particular failure state is independent of the demand routing in the remaining states, we can perform the *feasibility test* in question by checking the feasibility separately for each state by means of the following linear program.

Problem F(c, s) (feasibility of c in state s):

$$O(c, s) = \min \textstyle\sum_{e\in\mathcal{E}} z_e \tag{2a}$$

$$[\lambda^v_d] \quad \textstyle\sum_{a\in\delta^+(v)} x_{ad} - \sum_{a\in\delta^-(v)} x_{ad} = \begin{cases} h(d), & \text{if } v = o(d) \\ -h(d), & \text{if } v = t(d), \\ 0, & \text{otherwise} \end{cases} \quad d \in \mathcal{D}, v \in \mathcal{V} \tag{2b}$$

$$[\pi_a \ge 0] \quad \textstyle\sum_{d\in\mathcal{D}} x_{ad} \le \big(1 - \beta(e(a), s)\big) c(e(a)) + z_{e(a)}, \quad a \in \mathcal{A} \tag{2c}$$

$$x, z \geq 0 \text{ and continuous,} \tag{2d}$$

where $O(c,s)$ expresses the minimum of the sum of links' overloads measured by the components of the vector $z := (z_e,\ e \in \mathcal{E})$. The test is valid since c is feasible for s if, and only if, $O(c,s) = 0$. If the result of the test is negative (i.e., $O(c,s) > 0$) we need to find an inequality that separates c from $\mathcal{Y}$. This can be done by considering the dual to (2) (whose variables $\lambda := (\lambda_d^v,\ v \in \mathcal{V}, d \in \mathcal{D})$ and $\pi := (\pi_a,\ a \in \mathcal{A})$ correspond to the primal constraints (2b) and (2c), respectively).

Problem D(c,s) (dual to F(c,s)):

$$W(c,s) = \max\Big\{ \textstyle\sum_{d\in\mathcal{D}} \lambda_d^{t(d)} h(d) - \sum_{e\in\mathcal{E}} (\pi_{a(e)} + \pi_{a'(e)})\big(1 - \beta(e,s)\big)c(e) \Big\} \tag{3a}$$

$$\pi_{a(e)} + \pi_{a'(e)} \leq 1,\ e \in \mathcal{E}; \qquad \lambda_d^{o(d)} = 0,\ d \in \mathcal{D} \tag{3b}$$

$$\lambda_d^{t(a)} - \lambda_d^{o(a)} \leq \pi_a, \quad a \in \mathcal{A}, d \in \mathcal{D} \tag{3c}$$

$$\pi \geq 0 \text{ and continuous, } \lambda \text{ continuous.} \tag{3d}$$

Let λ^*, π^* be an optimal solution of problem D(c,s). Since $W(c,s) = O(c,s)$ (where $O(c,s)$ is defined by (2a)), the inequality that separates c from $\mathcal{Y}$ (provided $W(c,s) > 0$) is as follows:

$$\textstyle\sum_{e\in\mathcal{E}} (\pi^*_{a(e)} + \pi^*_{a'(e)})\big(1 - \beta(e,s)\big) M y_e \geq \sum_{d\in\mathcal{D}} {\lambda_d^{t(d)}}^* h(d). \tag{4}$$

Since $\sum_{e\in\mathcal{E}} (\pi^*_{a(e)} + \pi^*_{a'(e)})\big(1 - \beta(e,s)\big)c(e) = \sum_{d\in\mathcal{D}} {\lambda_d^{t(d)}}^* h(d) - W(c,s)$ (and $W(c,s) > 0$), $My = c$ does not fulfil (4).

Note that to make sure that $c \in \mathcal{Y}$, we in general need to perform the feasibility test (3) individually for all states s in $\mathcal{S}$.

4.2. A cut generation algorithm for a given explicit list of states $\mathcal{S}$

The iterative algorithm for solving problem (1) for an arbitrary (but given) explicit list of states $\mathcal{S}$ is presented below. It results from application of the classical Benders' decomposition method [21] to P$(\mathcal{S})$ (this particular application can be found in [6]). In each iteration the master problem involving only the capacity variables y is solved and then its optimal solution y^* is tested for feasibility with respect to P(S). If the test is positive, the algorithm is stopped and y^* is optimal for (1). If not, new inequalities deduced from the feasibility tests are added to the master problem and the algorithm is reiterated. (Below, the notation $y \in \Omega$ means that y fulfills all inequalities in the set of inequalities Ω.)

Algorithm 1: cut generation algorithm for an explicit list of states $\mathcal{S}$

Step 0: $\Omega := \{y \geq 0\}$.

Step 1: Solve the master problem:

$$\text{minimize} \qquad \textstyle\sum_{e\in\mathcal{E}} \xi(e) y_e \tag{5a}$$

$$\text{subject to} \qquad y \in \Omega \quad \text{and integer,} \tag{5b}$$

and put $c = (My^*_e, e \in \mathcal{E})$.

Step 2: For each $s \in \mathcal{S}$ solve the feasibility test (3) and if $W(c,s) > 0$, then add inequality (4) to Ω.

Step 3: If no inequalities have been added to Ω in Step 2, then stop (c is the capacity vector optimal for problem (1)). Otherwise go to Step 1.

To solve a given instance of problem P($\mathcal{S}$) we use a modified (two-phase) version of the above algorithm. First, in Phase 1, we apply Algorithm 1 to the linear relaxation of P($\mathcal{S}$), i.e., we assume continuous instead of integer y in (5b) which makes the master problem (5) a linear program. When Phase 1 is completed, we start Phase 2: we bring back the integrality requirement on y to (5b) in the master problem and continue.

4.3. Feasibility testing for K-sets

It is important to observe that Algorithm 1 can become ineffective when the set $\mathcal{S}$ contains a large number of states, as this leads to an excessive number of feasibility tests to be performed in each iteration. A way to deal with issue is proposed below.

Let us consider a special case of the family of state sets denoted by $\mathcal{F}$. A set $\mathcal{S}$ belongs to $\mathcal{F}$ when each link $e \in \mathcal{E}$ is degraded (if at all) with the same failure rate $\beta(e)$, in all states s in $\mathcal{S}$: $\beta(e,s) \in \{\beta(e), 0\}$, $s \in \mathcal{S}$. A particular example of such a set is the state set $\mathcal{S}(K)$ determined by the so called *K-set*. A K-set, denoted with $\mathcal{U}(K)$, is defined, for any integer $K = 1, 2, \ldots, E$, as the set of binary vectors $\mathcal{U}(K) := \{u = (u(e),\, e \in \mathcal{E}) : \sum_{e\in\mathcal{E}} u(e) \le K,\, u(e) \in \{0,1\}, e \in \mathcal{E}\}$. Each $u \in \mathcal{U}(K)$ determines a state $s(u) \in \mathcal{S}(K)$ by putting $\beta(e, s(u)) := \beta(e)u(e)$, $e \in \mathcal{E}$. Thus, the support of u determines the set of links failing in $s(u)$, wherein each such link fails with ratio $\beta(e)$. Note that from the viewpoint of problem P($\mathcal{S}$) formulated in (1), every set S in $\mathcal{F}$ is dominated by the set $\mathcal{S}(K)$ induced by the K-set with $K = \max\{|supp(\beta(s))| :\, s \in \mathcal{S}\}$, where $|supp(\beta(s))|$ denotes the number of non-zero elements in the failure ratio vector $\beta(s)$. The domination in question means that each feasible solution of P($\mathcal{S}(K)$) is feasible for P($\mathcal{S}$). Thus the optimal solution of P($\mathcal{S}(K)$) will be resilient with respect to all states in $\mathcal{S}$. For convenience, the optimization problem (1) corresponding to a K-set $\mathcal{U}(K)$ will be from now on denoted by P(K) instead of P($\mathcal{S}(K)$).

Note that the number $U(K) := \sum_{k=0}^{K} \binom{E}{k}$ of the states in $\mathcal{U}(K)$ grows exponentially with the number of links E, provided K increases linearly with E, for example when $K = \lfloor \frac{E}{2} \rfloor$. Thus, although problem P(K) is linear, it is in general non-compact as the number of variables and constraints is proportional to $U(K)$ and hence exponential with E. Therefore, the direct approach to P(K) by means of an LP solver is not applicable for large networks, and neither is Benders' decomposition applied in Algorithm 1 presented in the previous section (to make sure that $c \in \mathcal{Y}$, Algorithm 1 needs to perform test (3) individually for all states s in $\mathcal{S}$). For this reason, we have developed a more efficient approach to P(K). The approach is similar to Benders' decomposition used in Algorithm 1. The difference is that for the state set $\mathcal{S}(K)$ determined by a K-set (for some fixed K), the feasibility test can be made efficient by finding the maximum $W(c) := \max_{s\in\mathcal{S}(K)} W(c,s)$. The adjusted test makes use of binary variables $u := (u_e,\, e \in \mathcal{E})$ that correspond to the vectors $u = (u(e),\, e \in \mathcal{E}) \in \mathcal{U}(K)$, and is as follows.

Problem G(c) (maximum violation):

$$W(c) = \tag{6a}$$

$$= \max\Big\{ \sum_{d\in\mathcal{D}} \lambda_d^{t(d)} h(d) - \sum_{e\in\mathcal{E}} (\pi_{a(e)} + \pi_{a'(e)}) c(e) + \sum_{e\in\mathcal{E}} \beta(e) c(e) (\pi_{a(e)} + \pi_{a'(e)}) u_e \Big\}$$

$$\pi_{a(e)} + \pi_{a'(e)} \le 1,\ e \in \mathcal{E}; \qquad \lambda_d^{o(d)} = 0,\ d \in \mathcal{D} \tag{6b}$$

$$\lambda_d^{t(a)} - \lambda_d^{o(a)} \le \pi_a,\ \ a \in \mathcal{A}, d \in \mathcal{D} \tag{6c}$$

$$\sum_{e\in\mathcal{E}} u_e \le K, \tag{6d}$$

$$\pi \ge 0 \text{ and continuous; } \lambda \text{ continuous; } u \text{ binary.} \tag{6e}$$

Above, any vector of binary variables $u := (u_e,\ e \in \mathcal{E})$ fulfilling constraints (6d) defines a state $s(u) \in \mathcal{S}(K)$ with the failure ratio vector $(\beta(e, s(u)) = \beta(e)u_e,\ e \in \mathcal{E})$. Moreover, for any fixed feasible u the value of the objective function maximized over λ, π is equal to $W(c, s)$ for the so defined $s(u)$. Since in (6) we are maximizing also over u, we finally observe that $W(c) = \max_{s\in\mathcal{S}(K)} W(c, s)$, as required.

Note that if we define the set $\widetilde{\mathcal{U}}(K) := \{u = (u(e),\ e \in \mathcal{E}) : \sum_{e\in\mathcal{E}} u(e) \le K, 0 \le u(e) \le 1,\ e \in \mathcal{E}\}$ (i.e., the convex hull of $\mathcal{U}(K)$, called the *uncertainty polytope* [22]), then any given capacity vector c is feasible for P(K) if, and only if, it is feasible for the state set $\bar{\mathcal{S}}(K) := \{s(u) :\ u \in \bar{\mathcal{U}}(K)\}$ characterized by the following vectors of failure ratios:

$$\beta(s(u)) := (\beta(e)u(e),\ e \in \mathcal{E}),\ u \in \widetilde{\mathcal{U}}(K).$$

This follows from the fact that the version of G(c) with λ and π fixed and u relaxed, i.e., with $0 \le u_e \le 1$ instead of $u_e \in \{0, 1\}$, $e \in \mathcal{E}$, is an LP problem with a totally unimodular matrix of coefficients (recall that K is a positive integer).

Certainly, to obtain a proper mixed-integer programming formulation we need to take care of bi-linearities in (6) induced by multiplications of variables $(\pi_{a(e)} + \pi_{a'(e)}) \cdot u_e$, $e \in \mathcal{E}$, with u_e binary and the value of $(\pi_{a(e)} + \pi_{a'(e)})$ between 0 and 1. To get rid of these bi-linearities we introduce additional (continuous) variables U_e, $e \in \mathcal{E}$, that in the optimal solution will be equal to $(\pi_{a(e)} + \pi_{a'(e)}) \cdot u_e$. This is done in the next formulation.

Problem GMIP(c) (MIP version of G(c)):

$$W(c) = \max\Big\{ \sum_{d\in\mathcal{D}} \lambda_d^{t(d)} h(d) - \sum_{e\in\mathcal{E}} (\pi_{a(e)} + \pi_{a'(e)}) c(e) + \sum_{e\in\mathcal{E}} \beta(e) c(e) U_e \Big\} \tag{7a}$$

$$\pi_{a(e)} + \pi_{a'(e)} \le 1,\ e \in \mathcal{E}; \qquad \lambda_d^{o(d)} = 0,\ d \in \mathcal{D} \tag{7b}$$

$$\lambda_d^{t(a)} - \lambda_d^{o(a)} \le \pi_a,\ \ a \in \mathcal{A}, d \in \mathcal{D} \tag{7c}$$

$$\sum_{e\in\mathcal{E}} u_e \le K, \tag{7d}$$

$$U_e \le \pi_{a(e)} + \pi_{a'(e)},\ U_e \le u_e,\ \ e \in \mathcal{E} \tag{7e}$$

$$\pi \ge 0 \text{ and continuous; } \lambda, U \text{ continuous; } u \text{ binary.} \tag{7f}$$

Note that the (otherwise necessary) condition $U_e \ge \pi_{a(e)} + \pi_{a'(e)} + u_e - 1$ is omitted in formulation (7) since the values U_e in the vector $U := (U_e,\ e \in \mathcal{E})$ are maximized by objective (7a). Now let λ^*, π^*, U^* (and u^*) be an optimal solution of (7). If $W(c) > 0$ then the capacity vector c is infeasible for the main problem P(K) formulated in (1), and the following (Benders) inequality (written in variables y)

$$\sum_{e\in\mathcal{E}} (\pi^*_{a(e)} + \pi^*_{a'(e)})(1 - \beta(e)u^*_e) M y_e - \sum_{d\in\mathcal{D}} \lambda_d^{t(d)^*} h(d) \ge 0 \tag{8}$$

separates c from the set of capacity vectors feasible for P(K). Note that c breaks the above inequality by $W(c) = \max_{s\in\mathcal{S}(K)} O(c,s)$, i.e., by the sum of the link overloads (minimized over the demand routing) in the "worst state" $s \in \mathcal{S}(K)$.

4.4. A cut generation algorithm for $\mathcal{S}(K)$

The iterative algorithm for solving problem (1) for link availability states specified through a given K-set is presented below. As for Algorithm 1 for a fixed list of states discussed in Section 4.2, in each iteration the master problem involving only the capacity variables y is solved and then its optimal solution y^* is tested for feasibility with respect to P(K). If the test is positive, the algorithm is stopped and y^* is optimal for (1). If not, a new inequality deduced from the feasibility test is added to the master problem and the algorithm is reiterated. (Recall that notation $y \in \Omega$ means that y fulfills all inequalities in the set of inequalities Ω.)

Algorithm 2: cut generation algorithm for the set of states $\mathcal{S}(K)$

Step 0: $\Omega := \{y \geq 0\}$.

Step 1: Solve the master problem:

$$\begin{aligned} &\text{minimize} && \textstyle\sum_{e\in\mathcal{E}} \xi(e) y_e && \text{(9a)}\\ &\text{subject to} && y \in \Omega \quad \text{and integer}, && \text{(9b)} \end{aligned}$$

and put $c = (My_e^*, e \in \mathcal{E})$.

Step 2: Solve the feasibility test (7). If $W(c) \leq 0$, then stop (c is the capacity vector optimal for problem (1)).

Step 3: Otherwise, add inequality (8) to Ω and go to Step 1.

As with Algorithm 1, in the computations we apply the two-phase version of Algorithm 2.

5. Computational complexity of P($\mathcal{K}$)

When the set of states $\mathcal{S}$ is determined by means of K-sets, i.e., when $\mathcal{S} = \mathcal{S}(K)$, already the linear relaxation of problem P(K) (linear relaxation of P(K) assumes continuous link capacities) becomes difficult. This, as discussed below, is suggested by co$\mathcal{NP}$-completeness of the decision version of the feasibility test (6). (For a presentation of the computational complexity theory, and the notion of co$\mathcal{NP}$-completeness in particular, see [23].)

For the purpose of this section we will assume that the network graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{E})$ described in Section 3.1 is composed of undirected rather than bi-directed links (and call such a graph *undirected*). As for the bi-directed case, the set $\mathcal{A}$ of arcs is composed of the directed arcs corresponding to the (undirected) links $\mathcal{E}$ (there are two oppositely directed arcs corresponding to each link). In graph $\mathcal{G}$ the capacity $c(e)$ (or y_e when the link capacity is an optimization variable) of a link $e \in \mathcal{E}$ is undirected and shared by the (directed) flows using this link in both directions. In other words, the load of link e is the sum of the loads of the two oppositely directed arcs associated with link e, i.e., $a(e)$ and $a'(e)$, that compete

for the undirected link capacity. In the undirected graph setting, problem P($\mathcal{S}$) is as follows.

Problem P($\mathcal{S}$) (main problem for undirected graphs):

$$C(\mathcal{S}) = \min \sum_{e\in\mathcal{E}} \xi(e) y_e \tag{10a}$$

$$\sum_{a\in\delta^+(v)} x^s_{ad} - \sum_{a\in\delta^-(v)} x^s_{ad} = \begin{cases} h(d), & \text{if } v = o(d) \\ -h(d), & \text{if } v = t(d), \\ 0, & \text{otherwise} \end{cases} \quad d\in\mathcal{D}, v\in\mathcal{V}, s\in\mathcal{S} \tag{10b}$$

$$\sum_{d\in\mathcal{D}} (x^s_{a(e)d} + x^s_{a'(e)d}) \le \big(1-\beta(e,s)\big) M y_e, \quad e\in\mathcal{E},\ s\in\mathcal{S} \tag{10c}$$

$$x \ge 0 \text{ and continuous; } y \ge 0 \text{ and integer.} \tag{10d}$$

Although the above formulation requires some (straightforward) adjustments in the feasibility tests, the undirected link assumption does not affect the essence of the optimization algorithm presented above. In particular, the feasibility test formulated in (2) adjusted for P($\mathcal{S}$) formulated in (10) takes the following form.

Problem F(c, s) (feasibility of c in state s for undirected graphs):

$$O(c,s) = \min \sum_{e\in\mathcal{E}} z_e \tag{11a}$$

$$\sum_{a\in\delta^+(v)} x_{ad} - \sum_{a\in\delta^-(v)} x_{ad} = \begin{cases} h(d), & \text{if } v = o(d) \\ -h(d), & \text{if } v = t(d), \\ 0, & \text{otherwise} \end{cases} \quad d\in\mathcal{D}, v\in\mathcal{V} \tag{11b}$$

$$\sum_{d\in\mathcal{D}} (x_{a(e)d} + x_{a'(e)d}) \le \big(1-\beta(e,s)\big) c(e) + z_e, \quad e\in\mathcal{E} \tag{11c}$$

$$x, z \ge 0 \text{ and continuous,} \tag{11d}$$

For a given undirected network graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{E})$, parameter K ($1 \le K \le |\mathcal{E}|$), and vectors $h = (h(d), d\in\mathcal{D})$ and $\beta = (\beta(e), e\in\mathcal{E})$, let $\mathcal{C}(\mathcal{S}(K))$ be the set (a polyhedron) of all (continuous) link capacity vectors c that are feasible for all states $s\in\mathcal{S}(K)$, i.e., for which $O(c,s) = 0$ for all $s\in\mathcal{S}(K)$ (see (11a)). The separation problem for polyhedron $\mathcal{C}(\mathcal{S}(K))$, corresponding to the feasibility test (11) and referred to as DSPS/IN, is defined as the following decision problem:

DSPS/IN: does a given link capacity vector c belong to polyhedron $\mathcal{C}(\mathcal{S}(K))$?

Note that DSPS/IN is in class co$\mathcal{NP}$ of decision problems since its complement, i.e., the separation problem of the form

DSPS/OUT: is a given link capacity vector c outside polyhedron $\mathcal{C}(\mathcal{S}(K))$?

is in class $\mathcal{NP}$. This is true because if c is a YES instance of DSPS/OUT, then any state $s\in\mathcal{S}(K)$ for which c is infeasible can be used as a certificate for verifying this in polynomial time.

Now let us consider a decision problem similar to DSPS/IN for an undirected network graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{E})$ where $\mathcal{D}$ represents a set of *single source demands* having their origins

in a fixed vertex $w \in \mathcal{V}$, i.e., $o(d) = w$, $t(d) \in \mathcal{V} \setminus \{w\}$, $d \in \mathcal{D}$. For a given parameter K ($1 \leq K \leq |\mathcal{D}|$), let $\mathcal{H}(K)$ be the set of (binary) demand volume vectors h defined as:

$$\mathcal{H}(K) := \{h = (h(d),\, d \in \mathcal{D}) :\ \textstyle\sum_{d\in\mathcal{D}} h(d) \leq K,\ h(d) \in \{0,1\},\ d \in \mathcal{D}\}.$$

Further, let $\mathcal{C}(\mathcal{H}(K))$ be a polyhedron of all link capacity vectors c that are feasible for demand volume vectors $h \in \mathcal{H}(K)$. Note that in the considered setting link capacities are not subject to failures, and each vector h is realized by means of its individual routing, just like in the DSPS/IN setting each state s is realized by its individual routing, see (10b). Finally, define the following decision (separation) problem referred to as DSPH/IN:

DSPH/IN: does a given link capacity vector c belong to polyhedron $\mathcal{C}(\mathcal{H}(K))$?

It is shown in Section 2 of [24] that DSPH/IN is co$\mathcal{NP}$-complete, provided the *unique games conjecture* holds. (The conjecture has already been used to prove hardness for different problems that have resisted previous attempts, see [25] for an introduction to the conjecture). We will make use of this fact to show that DSPS/IN is co$\mathcal{NP}$-complete (provided the unique games conjecture holds) by constructing a polynomial transformation of the instances of DSPH/IN to instances of DSPS/IN.

Proposition 1: DSPH/IN polynomially transforms to DSPS/IN.

Proof. Consider an instance of problem DSPH/IN for a given link capacity vector $c = (c(e),\, e \in \mathcal{E})$ and an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{E})$ with the set $\mathcal{D}$ of single-source demands and (uncertain) demand volumes described by the set $\mathcal{H}(K)$. Our transformation uses an augmented (with respect to $\mathcal{G}$) undirected graph $\mathcal{G}' = (\mathcal{V}, \mathcal{A}', \mathcal{E}')$. The set of undirected links is defined as follows: $\mathcal{E}' = \mathcal{E} \cup \mathcal{E}''$, where $\mathcal{E}'' := \bigcup_{d\in\mathcal{D}} \{e(d)\}$ and $e(d) := \{w, t(d)\}$. We assume that the sets $\mathcal{E}''$ and $\mathcal{E}$ are disjoint – this means that when, for a given $d \in \mathcal{D}$, the link between nodes w and $t(d)$ already exists in the original set of links $\mathcal{E}$, an extra parallel link $e(d)$ is created. Next, we assume that $\mathcal{D}' = \mathcal{D}$, i.e., the origins and destinations of the demands in the augmented graph $\mathcal{G}'$ are unchanged. Yet, for $\mathcal{G}'$, the demand volumes are fixed and $h'(d)$ is equal to 1 for each $d \in \mathcal{D}'$. Finally, we define $\beta(e) = 0$, $e \in \mathcal{E}$, $\beta(e) = 1$, $e \in \mathcal{E}''$, and $c'(e) = c(e)$, $e \in \mathcal{E}$, $c'(e) = 1$, $e \in \mathcal{E}''$.

We claim that c (the vector of link capacities in graph $\mathcal{G}$) is feasible for all demand volume vectors h in $\mathcal{H}(K)$ if, and only if, c' (the vector of link capacities in graph $\mathcal{G}'$) is feasible for all states s in $\mathcal{S}(K)$. In other words, $c \in \mathcal{C}(\mathcal{H}(K))$ if, and only if, $c' \in \mathcal{C}(\mathcal{S}(K))$; this implies the proposition we are proving.

To prove the claim we first note that the assumption $\beta(e) = 0$, $e \in \mathcal{E}$, implies that c' is feasible for all states induced by $\mathcal{U}(K)$ (i.e, the K-set considered for $\mathcal{G}'$; the definition of $\mathcal{U}(K)$ is given in Section 4.3) if, and only if, c' is feasible for all states induced by the set $\widehat{\mathcal{U}}(K) := \{u \in \mathcal{U}(K) :\ u(e) = 0,\, e \in \mathcal{E}\}$, i.e., for those states in $\mathcal{U}(K)$ that affect only the extra links (in $\mathcal{E}''$), and not the original links of graph $\mathcal{G}$ (in $\mathcal{E}$). Next, we define the bijection $\varphi : \widehat{\mathcal{U}}(K) \mapsto \mathcal{H}(K)$, where $\varphi(u) = h = (h(d),\, d \in \mathcal{D})$ is, for each $u \in \widehat{\mathcal{U}}(K)$, defined as $h(d) := 1 - u(e(d))$, $d \in \mathcal{D}$. In other words, $\varphi(u)(d) = h(d) = 1$ if, and only if, link $e(d) = \{w, t(d)\} \in \mathcal{E}''$ fails (totally, since by assumption, $\beta(e(d)) = 1$ for all $d \in \mathcal{D}$) in state $u \in \widehat{\mathcal{U}}(K)$.

As shown below, for each $u \in \widehat{\mathcal{U}}(K)$, routing the demand volume vector $h = \varphi(u)$ in the link capacity c available in graph $\mathcal{G}$ is equivalent to routing the demand volume vector $h' = (h'(d),\, d \in \mathcal{D}')$ (where, by definition, all $h'(d)$ are equal to 1) in the link capacity c' available in graph $\mathcal{G}'$ in the state $s(u)$ determined by a given $u \in \widehat{\mathcal{U}}(K)$. Since the mapping $\varphi : \widehat{\mathcal{U}}(K) \mapsto \mathcal{H}(K)$ is a bijection, this actually means that $c \in \mathcal{C}(\mathcal{H}(K))$ if, and only if, $c' \in \mathcal{C}(\mathcal{S}(K))$ (as claimed).

To complete the proof let us assume that the link capacity vector c of links $\mathcal{E}$ is feasible for routing a given vector $h = (h(d),\, d \in \mathcal{D}) \in \mathcal{H}(K)$ in graph $\mathcal{G}$. Then a feasible, with respect to the link capacity vector c' of links in $\mathcal{E}'$, routing of the demand vector h' (with all entries equal to 1) in state $u = \varphi^{-1}(h)$ in graph $\mathcal{G}'$ is obvious: each demand $d \in \mathcal{D}'$ for which $h(d) = 0$ is routed over the extra direct link $e(d)$ (note that the capacity of each such link in state u is equal, by construction, to 1), while the remaining demands (for which $h(d) = 1$) are routed as before, i.e., using only the capacity c of the links in $\mathcal{E}$.

Now, let us assume that the link capacity vector c' of the links in $\mathcal{E}'$ is feasible for routing the demand volume vector h' (with all entries equal to 1) in graph $\mathcal{G}'$ in state $s(u)$ for a given $u \in \widehat{\mathcal{U}}(K)$. Let $\mathcal{F}'$ denote a set of path-flows in graph $\mathcal{G}'$ resulting from the arc-flows (see (11b)) that specify the feasible routing in question. (Note that the directed arcs are not needed to describe the set $\mathcal{F}'$ as paths are characterized simply by the sets of undirected links they traverse.) Without loss of generality, we can assume that in $\mathcal{F}'$, for each given $d \in \mathcal{D}'$, the path-flows between w and $t(d)$ that realize $h'(d)$ are undirected and contain no loops. We will show that there exits a feasible routing specified by an analogous set $\mathcal{F}''$ of path-flows with the following property holding for each demand $d \in \mathcal{D}'$: d is routed entirely on the extra link $e(d) = \{w, t(d)\}$ when $u(e(d)) = 0$; otherwise, when $u(e(d)) = 1$, d is routed entirely over graph $\mathcal{G}$. Indeed, consider a demand $d \in \mathcal{D}'$ with $u(e(d)) = 0$ and assume that a part of its volume, say g (where $0 < g \leq h'(d) = 1$) is routed by means of flows in $\mathcal{F}'$ realized on paths not-containing link $e(d)$. In such a case, the part equal to $1 - g$ of the capacity of link $e(d)$ (note that $c(e(d)) = 1$) is occupied by a flow in $\mathcal{F}'$ realizing demand d (from w to $t(d)$), and the remaining part (equal to g) is used by the path-flows in $\mathcal{F}'$ realizing demands other than d. Assume that the total volume of these path-flows is g' (where $g' \leq g$). Suppose that we disconnect the segment of each such path-flow consisting of link $e(d)$. As the result, the part of $1 - g$ of the capacity of link $e(g)$ becomes idle and hence we can re-switch the path-flows in $\mathcal{F}'$ realizing d around link $e(d)$ back to $e(d)$ making the link fully occupied by demand d. After doing that, capacity equal to g on paths between w and $t(d)$ not containing $e(d)$ becomes idle and we can use it to move the path-flow segments removed from link $e(d)$ to them, in this way restoring the previously disconnected path-flows. (Note that this operation may create loops but they can be easily eliminated.) The above described procedure must be executed one-by-one for all demands in $\mathcal{D}'$. Finally, consider the demand volume vector $h = \varphi(u)$ and observe that when the process is completed, the demands in $\mathcal{D}'$ with $h(d) = 0$ are realized on the extra links $e(d)$ exhausting their capacity, so that the demand in $\mathcal{D}'$ with $h(d) = 1$ must be routed in graph $\mathcal{G}$. Thus, $\mathcal{F}''$ defines a feasible routing for h in $\mathcal{G}$. □

Corollary: If the unique games conjecture holds, the feasibility test based on formulation (11) (used in Algorithm 2) is $\mathcal{NP}$-hard.

Proof. Assume that the unique games conjecture is true. Then, Proposition 1 implies that DSPS/IN is co$\mathcal{NP}$-complete since the decision problem DSPH/IN (whose co$\mathcal{NP}$-completeness is proven in [24]) can be polynomially transformed to DSPS/IN. It follows that DSPS/OUT, as the complement of DSPS/IN, is $\mathcal{NP}$-complete (see Theorem 15.31 in [23]). Thus, the problem formulated in (11), as an optimization version of DSPS/OUT, is $\mathcal{NP}$-hard. □

Finally we note that $\mathcal{NP}$-hardness of the test (11) does not necessarily mean that the problem represented by the linear relaxation of formulation P(K) is $\mathcal{NP}$-hard. In fact, it only means, by equivalence of separation and optimization (see Section 3 of Chapter I.6 in [26]), that the family of linear programming problems defined on polyhedron $\mathcal{C}(\mathcal{S}(K))$ (linear relaxation of the master problem (9) with $\Omega = \mathcal{C}(\mathcal{S}(K))$ is a member of this family) contains $\mathcal{NP}$-hard linear programming problems. Yet, in general, there can be polynomial problems in the considered family as well. (For a discussion of this issue in the network optimization context see [27].) Nevertheless, $\mathcal{NP}$-hardness of (11) makes $\mathcal{NP}$-hardness of P(K) (also in its basic, bi-directional form) likely.

6. Modifications and improvements

6.1. Modifications of the optimization model

As shown below, the node-arc formulation (1) of P($\mathcal{S}$) can be easily modified in order to capture other types of network graphs besides the so far considered case assuming the bi-directed graph with full-duplex link capacity. Also, the aggregated node-arc formulations (see [6]) can be applied to (1) and all the modifications discussed in Sections 6.1.1 and 6.1.2. Such formulations use aggregated arc-flows of the form x^s_{av} (related to the ordinary arc-flows through equality $x^s_{ad} = \sum_{d\in\mathcal{D},t(d)=v} x^s_{ad}$, $a \in \mathcal{A}, v \in \mathcal{V}, s \in \mathcal{S}$) expressing the total flow of traffic destined to node v on arc a in state s. Finally, link-path formulations can be used, as shown in Section 6.1.3

6.1.1. Directed links

In this case the network graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ is directed (rather than bi-directed) so it is not necessary that each arc has its oppositely directed counterpart. Besides, the link capacity is directed so each arc has its individual capacity y_a (or $n(a)$). To consider this, the objective (1a) and constraint (1c) are (respectively) modified as follows:

$$C(\mathcal{S}) = \min \textstyle\sum_{a\in\mathcal{E}} \xi(a) y_a, \qquad \sum_{d\in\mathcal{D}} x^s_{ad} \le \big(1 - \beta(a,s)\big) M y_a, \quad a \in \mathcal{A},\ s \in \mathcal{S},$$

where $\xi(a)$ and $\beta(a,s)$ are module costs and failure ratios, respectively, defined for individual arcs (although if two oppositely directed arcs connect the same pair of nodes it is natural to assume that in all states in $\mathcal{S}$ they have equal failure ratios).

Formulations of the feasibility test, its dual, and the maximum violation problem corresponding to the so modified formulation of (1) are obtained in essentially the same as formulations (2), (3), and (6), respectively.

6.1.2. Undirected links

In this case the network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is composed of the set of nodes $\mathcal{V}$ and the set of (undirected) links $\mathcal{E} \subseteq \mathcal{V}^{|2|}$. The extension of (1) is is obtained by substituting each undirected edge $e = \{v, w\} \in \mathcal{E}$ by two oppositely directed arcs $a(e) = (v, w)$ and $a'(e) = (w, v)$ (that specify an auxiliary bi-directed graph used to route the arc-flows) and defining the load of edge $\{v, w\}$ as the sum of the loads of arcs (v, w) and (w, v). This is expressed by substituting constraint (1c) with

$$\textstyle\sum_{d\in\mathcal{D}}(x^s_{a(e)d} + x^s_{a'(e)d}) \le \big(1 - \beta(e, s)\big)My_e, \;\; e \in \mathcal{E},\; s \in \mathcal{S}.$$

The so adjusted formulation of P($\mathcal{S}$) works for directed demands (as assumed in Section 3.1) as well for undirected demands. In the latter case, each demand $d \in \mathcal{D}$, connecting nodes v and w, say, should be made directed through selecting one of v, w for the originating node $o(d)$ and the other for the terminating node $t(d)$.

As before, formulations of the feasibility test, its dual, and the maximum violation problem for the so modified formulation are essentially the same as for (1).

6.1.3. Link-path formulations

All the cases of problem P($\mathcal{S}$) considered in this paper can be also treated by means of link-path formulations involving path-flows (for a given list of predefined allowable routing paths) instead of arc-flows. The link-path formulation corresponding to node-arc formulation (1) is as follows.

Problem P($\mathcal{S}$) (main problem in link-path formulation):

$$C(\mathcal{S}) = \min \textstyle\sum_{e\in\mathcal{E}} \xi(e)y_e \tag{13a}$$

$$\textstyle\sum_{p\in\mathcal{P}(d)} x^s_{dp} = h(d), \;\; d \in \mathcal{D},\; s \in \mathcal{S} \tag{13b}$$

$$\textstyle\sum_{d\in\mathcal{D}} \sum_{p\in\mathcal{P}(d)} \delta(a, d, p)x^s_{dp} \le \big(1 - \beta(e(a), s)\big)My_{e(a)}, \;\; a \in \mathcal{A},\; s \in \mathcal{S} \tag{13c}$$

$$x \ge 0 \text{ and continuous; } y \ge 0 \text{ and integer.} \tag{13d}$$

Above, x^s_{dp} is the flow on path $p \in \mathcal{P}(d)$ dedicated to carry the traffic of demand $d \in \mathcal{D}$ in state $s \in \mathcal{S}$, where $\mathcal{P}(d)$ is a predefined list of elementary directed paths in graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ from node $o(d)$ (origin of demand d) to node $t(d)$ (termination of d). (Recall that an elementary path does not traverse any node more than once.) The left-hand side of the capacity constraint (13c) expresses the load of arc a using the given binary arc-path incidence coefficients $\delta(a, d, p)$ ($\delta(a, d, p) = 1$ if, and only if, arc a belongs to path $p \in \mathcal{P}(d)$). Clearly, the arc-flow x^s_{ad} induced by a given set of path-flows realizing demand d is equal to $\sum_{p\in\mathcal{P}(d)} \delta(a, d, p)x^s_{dp}$. The inverse relationship cannot be unambiguously specified, as a given set of arc-flows may induce different sets of path-flows. Yet, a set of path-flows corresponding to a given set of arc-flows can be easily obtained (see [23]).

Although the link-path formulation leads to a dual polytope different from (3b)-(3d), the dual function (3a) used for testing feasibility of the link capacity vector c remains essentially the same. The link-path formulation is valid for both bi-directed, directed and undirected networks, but requires path generation for finding the lists of allowable paths necessary to find the optimal solution when all possible paths are considered.

6.2. Valid inequalities

In fact, the Benders cutting plane approach may need a large number of Benders' inequalities to converge to optimum. This may pose an efficiency issue since test (7) (called also the separation problem) performed in Step 2 of our algorithm contains binary variables. One way to deal with this issue is to try to speed up the convergence of the algorithm by using additional valid inequalities on top of the Benders cuts in the master problem solved in Step 1 of the algorithm.

A valid inequality can be obtained by substituting the last sum in (6a) by the quantity $K \cdot B$, where $B := \frac{\sum_{e\in\mathcal{E}} \beta(e)c(e)(\pi_{a(e)}+\pi_{a'(e)})}{E}$ is the average value of the terms $\beta(e)c(e)(\pi_{a(e)} + \pi_{a'(e)})$, $e \in \mathcal{E}$. The corresponding test would thus be as follows:

$$\max \left\{ \sum_{d\in\mathcal{D}} \lambda_d^{t(d)} h(d) - \sum_{e\in\mathcal{E}} (\pi_{a(e)} + \pi_{a'(e)})c(e) + \tfrac{K}{E} \sum_{e\in\mathcal{E}} \beta(e)c(e)(\pi_{a(e)} + \pi_{a'(e)}) \right\} \quad (14a)$$

$$\pi_{a(e)} + \pi_{a'(e)} \leq 1, \quad e \in \mathcal{E} \qquad \lambda_d^{o(d)} = 0, \quad d \in \mathcal{D} \quad (14b)$$

$$\lambda_d^{t(a)} - \lambda_d^{o(a)} \leq \pi_a, \quad a \in \mathcal{E}, d \in \mathcal{D} \quad (14c)$$

$$\pi \geq 0 \text{ and continuous, } \lambda \text{ continuous.} \quad (14d)$$

When the resulting maximum is positive, the following valid inequality

$$\sum_{d\in\mathcal{D}} \lambda_d^{t(d)^*} h(d) - \sum_{e\in\mathcal{E}} (\pi^*_{a(e)} + \pi^*_{a'(e)})y_e + \tfrac{K}{E} \sum_{e\in\mathcal{E}} (\pi^*_{a(e)} + \pi^*_{a'(e)})\beta(e)y_e \leq 0 \quad (15)$$

is obtained. Note that (14) is a linear programming problem while formulation (7) requires binary variables,

Similar valid inequalities can be obtained for subsets $\mathcal{E}'$ of links with $k := |\mathcal{E}'| \leq K$. In the corresponding feasibility test the objective is as follows:

$$\begin{aligned}\max \Big\{ & \sum_{d\in\mathcal{D}} \lambda_d^{t(d)} h(d) - \sum_{e\in\mathcal{E}} (\pi_{a(e)} + \pi_{a'(e)})c(e) + \\ & + \tfrac{K-k}{E-k} \sum_{e\in\mathcal{E}\setminus\mathcal{E}'} \beta(e)c(e)(\pi_{a(e)} + \pi_{a'(e)}) + \sum_{e\in\mathcal{E}'} \beta(e)c(e)(\pi_{a(e)} + \pi_{a'(e)}) \Big\}. \end{aligned} \quad (16a)$$

Another useful class of valid inequalities are the cutset inequalities, first introduced in the robust context in [9]. Given a partition $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ of the nodes, the cutset inequality associated with the partition states that the amount of capacity installed on the arcs going from $\mathcal{V}_1$ to $\mathcal{V}_2$ should not be less than the sum of the volumes of the demands from $\mathcal{V}_1$ to $\mathcal{V}_2$. Defining $\mathcal{E}(\mathcal{V}_1, \mathcal{V}_2) = \{a \in \mathcal{A} : o(a) \in \mathcal{V}_1 \wedge t(a) \in \mathcal{V}_2\}$ and $\mathcal{D}(\mathcal{V}_1, \mathcal{V}_2) = \{d \in \mathcal{D} : o(d) \in \mathcal{V}_1 \wedge t(d) \in \mathcal{V}_2\}$, the inequality is formally defined as

$$\min_{s\in\mathcal{S}} \sum_{a\in\mathcal{E}(\mathcal{V}_1,\mathcal{V}_2)} \big(1 - \beta(e(a), s)\big)c(e(a)) \geq \sum_{d\in\mathcal{D}(\mathcal{V}_1,\mathcal{V}_2)} h(d) \quad (17)$$

While the exact separation of (17) requires solving mixed-integer programs, this can also be done heuristically. For instance, in [28] the nodes are randomly partitioned into two subsets and then a local search is performed, picking up one node and moving it to the other subset until there is no more improvement in the violation. If no violated inequality has been found, another initial partition is considered up to a maximum of 5 iterations.

In this context, it is noteworthy that recent implementations of Benders' decomposition have witnessed improvements when interior points instead of extreme points are separated in the algorithm, see [29].

6.3. K-set extension

The model presented and tested in the current paper takes into account one (but link-dependent) failure ratio level $\beta(e)$ for each link $e \in \mathcal{E}$. In reality, however, links may be affected with different levels of failure ratios, depending on the modulation and coding scheme applied in response to current weather conditions. As mentioned in Section 2, these ratios can for example be equal to 0, 0.25, 0.5, and 1.

To admit such multiple failure ratio levels we can assume that there are K different levels $0 \geq \beta(k) \leq 1$, $k \in \mathcal{K} = \{1, 2, \ldots, K\}$, and that in any considered state s at most $N(k)$ links can experience failure ratio $\beta(k)$, where $1 \leq N(k) \leq E$, $k \in \mathcal{K}$. The corresponding polytope is specified as follows.

$$\sum_{k \in \mathcal{K}} u_e^k = 1, \;\; e \in \mathcal{E}, \quad \sum_{e \in \mathcal{E}} u_e^k \leq N(k), \; k \in \mathcal{K},$$

where the binary variable $u_e^k = 1$ if, and only if, link e is degraded with the failure ratio $\beta(k)$.

The so obtained state characterization can be used to solve the problem analogous to P(K) by appropriately modifying formulations (6) and (7). In fact, an even more general version of the K-set (referred to as the uncertainty polytope) was considered in [13].

7. Numerical studies

In this section we will describe two case studies: a test network example taken from the network instances library SNDlib [30], and a realistic network instance prepared especially for this paper. All the experiments were performed on an Intel (R) Core (TM) i7- 4610M, 3.0 GHz computer with 16 GB of RAM. The algorithms code was written in Java and the optimization problems were solved with IBM ILOG CPLEX 12.6 (both using Concert Technology) running with the default setting.

7.1. Results for a test network

The discussion presented in this section aims at illustrating various aspects of the proposed robust approach based on Algorithm 2 on a reasonable (but actually not real) FSO network example (a slightly different variant of the considered example was studied in [31]). To speed up the calculations we have considered only the linear relaxation of P($\mathcal{S}$), that is we have used only Phase 1 of Algorithm 2.

7.1.1. Experiment layout

We have tested the proposed algorithm on a network example derived from the instance called *polska* that can be found in SNDlib (available at *sndlib.zib.de*). The original instance from SNDlib connects $V = 12$ nodes (metropolitan areas) and is composed of $E = 18$ undirected links. There are $D = 66$ undirected traffic demands expressed in Mbps. In our study we have examined a bi-directed version of *polska*, i.e., the basic network type considered in this paper, with $A = 2E = 36$ arcs and $D = 132$ demands, where each original (undirected) demand d has been split into two oppositely directed demands d', d'' linking the same nodes as d with $h(d') = h(d'') = \frac{h(d)}{2}$. The link module M is assumed to be equal to 1 Mbps, and the cost of one module on each link to 1 (i.e., $\xi(e) = 1$, $e \in \mathcal{E}$).

The reference set of failure states $\mathcal{S}$ for the *polska* study was obtained as described in Section 2. We analysed the weather states recorded over a one-year period (from August 16, 2014, to August 15, 2015) available at *www.worldweatheronline.com*. For the sake of simplicity only the weather conditions observed in the 12 metropolitan areas connected by the network were taken into account, assuming that the link connecting two given areas is affected by the worse of the weather condition in the two areas. The observed conditions were translated into the multiple partial link failure states corresponding to all hourly periods in the considered one-year time horizon, which gives $365 \times 24 = 8760$ "hourly" states in total. In this set of hourly states (denoted with $\widetilde{\mathcal{S}}$) there are 60.77% states with no affected links at all (these are the nominal states). It also happens that in the majority of states in $\widetilde{\mathcal{S}}$ the 4-QAM modulation and coding scheme is applied on each affected link instead of 16-QAM (i.e., the MCS used on a link in good weather conditions). This reduces the nominal capacity of the affected links by 25% (link failure ratio 0.25) – there are 39.15% of such states in $\widetilde{\mathcal{S}}$. In the remaining 0.08% of the hourly states, the MCS of some affected links is changed from 16-QAM to QSPK which reduces the nominal capacity by 50% (link failure ratio 0.5). Finally, there is a very small number of hourly states with a few links affected in 100% (link failure ratio 1 – total loss of communication). In terms of the average hourly state there are 89.3% of non-affected links, 10.7% of links with the failure rate 0.25, 0.017% links with the failure rate 0.5, and only 0.003% of links totally failed (failure ratio 1). Finally, we observed that out of all the 8760 hourly states there are only 159 states that are mutually different – these states constitute the reference failure set $\mathcal{S}$ for *polska*.

In the reported numerical experiments we first solve (using Algorithm 2) the instances of problem P(K) (1) for all K-sets ($K = 0, 1, \ldots, 18$) assuming the failure ratio $\beta(e) = 0.25$ for all 18 links when they fail. After that, for each K, we calculate the volume of traffic that cannot be carried in the hourly states in the reference failure states list $\widetilde{\mathcal{S}}$ that are outside the set of states $\mathcal{S}(K)$, i.e., in the hourly states outside the set of states induced by the considered K-set assumed for robust optimization.

7.1.2. Algorithm efficiency

Below we present the first set of results computed for network *polska* by means of Algorithm 2, the cut generation-based optimization algorithm for K-sets described in Section 4.4. As already mentioned, we consider the linear relaxation of problem P(K) that assumes continuous link capacities y and hence the algorithm is limited to Phase 1. The results include the minimum cost for P(K) (Figure 1, and the number of Benders' cuts (8) generated to reach the optimal robust solution together with the total computation time in seconds (Figure 2).

Figure 1 shows that the cost of the network becomes stable (and equal to 14 128 cost units) already for $K = 9$. This means that the additional capacity required to protect the traffic for all possible 9 (or less) simultaneous link failures (with the failure ratio 0.25) is already sufficient for all more severe simultaneous failures of K links where $K > 9$. In fact this additional capacity is equal to exactly 25% of the capacity required for unprotected network (case $K = 0$, with the cost equal to 10 596). This is not surprising since the network protected for the simultaneous failure of all 18 links with the failure ratio 0.25 (case $K = 18$) is obtained from the unprotected network by simply increasing the capacity of each link by 25%.

Concerning the time needed to obtain optimal solutions through the proposed cutting-

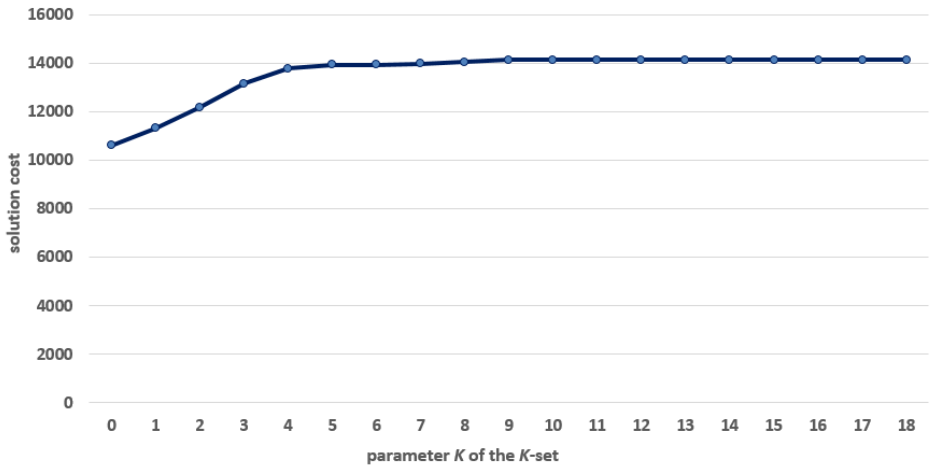

Figure 1: network *polska* – cost for different K-sets obtained with Algorithm 2 for continuous link capacities case (Phase 1).

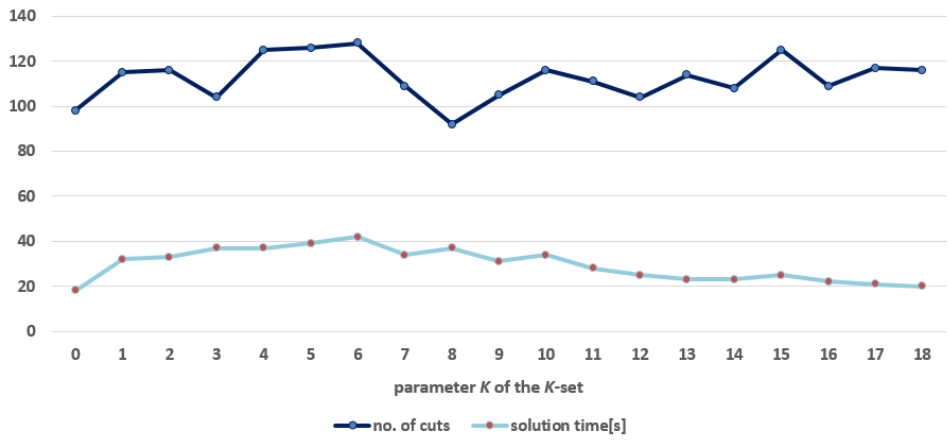

Figure 2: network *polska* – computation time and number of cuts achieved with Algorithm 2 for continuous link capacities case (Phase 1).

plane algorithm, we note that the computation time of problem P(K) for all K is reasonable. However, we may expect that for large networks the algorithm, in its current form, can exhibit excessive computation times. Thus, inclusion of additional valid inequalities (like those proposed in Section 6.2) could become important for speeding up the algorithm.

The upper graph shown in Figure 2 shows that there is no particular relationship between the number of the generated Benders' cuts (determined by the appropriate number left to the vertical axis) and the value of parameter K. The time required by Algorithm 2 to optimize the network (shown by the lower graph and determined by the numbers left to the vertical axis interpreted as seconds), however, does reveal such a relationship: the solution time is larger for intermediate values of K than for the values of K close to 1 or 18. This is because the separation time, i.e., the time spent in testing, which constitutes majority of time required by Phase 1 of Algorithm 2, is lower where K becomes closer to 0 or 18 as there are less combinations of failing links to be considered in the separation procedure.

This phenomenon was also observed in [7] and the references therein.

To end this section, we mention that instead of using Phase 1 of Algorithm 2, for $K = 0$ the linear relaxation of problem (1) can be solved directly, just by allocating the whole demand volume $h(d)$ for each $d \in \mathcal{D}$ to its shortest (with respect to the costs $\xi(e)$, $e \in \mathcal{E}$) path. Note that the so obtained optimal link capacity vector y^* will specify an optimal solution also for the case $K = |\mathcal{E}|$ if the individual link capacity y^*_e for each link $e \in \mathcal{E}$ is divided by $1 - \beta(e)$ (provided all $\beta(e) < 1$). Moreover, the optimal demand routing x^* for $K = 0$ is optimal for $K = |\mathcal{E}|$ as well. The computational time required by the so described (shortest path allocation) approach is negligible.

7.1.3. Robustness with respect to the hourly states

Table 2 reports, for each K, the results of testing the robustness of the optimal link capacity vector $y^* = (y_e,\, e \in \mathcal{E})$ obtained for P(K) against all (hourly) states in the set $\widetilde{\mathcal{S}}$ of all 8760 hourly states. Certainly, formulation of P(K) assures that the optimal capacity vector is sufficient to carry all the traffic demands $h(d)$, $d \in \mathcal{D}$, in each state in $\mathcal{S}(K)$, i.e., in each state induced by $\mathcal{U}(K)$. Yet, for the states in the reference failure set $\mathcal{S}$ (recall that there are 159 states in $\mathcal{S}$) outside $\mathcal{S}(K)$)) this is not guaranteed. Hence, for each such state $s \in \mathcal{S} \setminus \mathcal{S}(K)$ we have computed the volume of traffic that cannot be satisfied in this state, using a linear programming formulation similar to (2). In the formulation, link capacities are set to $c(e) := \big(1 - \beta(e,s)\big)y^*_e$, $e \in \mathcal{E}$, and the routing is optimized in order to minimize the total traffic loss, i.e., $\sum_{d\in\mathcal{D}} z_d$, where $0 \le z_d \le h(d)$, and $h(d) - z_d$ is the actual total traffic carried for demand d is the considered state. In Table 2 column "average traffic carried [%]" is the average percentage of total traffic carried (with respect to the total traffic demand) over all of the 8760 states in $\widetilde{\mathcal{S}}$, and "not covered states [%]" is the percentage of the states in $\widetilde{\mathcal{S}}$ for which some part of the offered traffic is not realized.

K	average carried traffic [%]	not covered states [%]
0	97.36	39.23
1	98.86	26.06
2	99.64	9.41
3	99.88	3.08
4	99.96	1.26
5	99.99	0.19
6	99.99	0.21
7	99.99	0.09
8	99.99	0.24
9	99.99	0.09
10	99.99	0.08
11	99.99	0.08
...	...	...
18	99.99	0.08

Table 2: network *polska* – coverage of the states outside $\mathcal{S}(K)$.

The first important observation is that, already for the network dimensioned for $K = 1$, it

is possible to guarantee robustness for almost 74% of the hourly states (i.e., only about 26% of the states in $\widetilde{\mathcal{S}}$ experience carried traffic degradation) with the percentage of carried traffic equal to 98.86%. With increasing K, the percentage of the states with no traffic degradation continues to increase until $K = 5$, the case for which our computations reveal the second interesting fact: the network dimensioned for $K = 5$ is already robust against all the states in $\mathcal{S}(18)$, i.e., for the states with all $E = 18$ links or less affected with a failure ratio equal to 0.25 – this effect is clearly seen in Table 2. This means that taking into account all $\binom{18}{5}$ states with exactly 5 links affected with the failure ratio 0.25 makes the network robust to all multiple partial link failure states with $\beta(e) \equiv 0.25$. This "saturation effect", for which an intermediate factor of robustness may already offer full protection against all deviations in the input data can be observed also in other (telecommunication) applications of robust optimization (see for example [7] and its list of references). A similar discussion can be found in [32] in the context of demand uncertainty.

Table 2 reveals that any K is not sufficient to cover all the states in $\widetilde{\mathcal{S}}$ and at least 0.08% of hourly states cannot be fully covered when the failure ratio in $\mathcal{S}$ is assumed to be 0.25. This is due to the existence of two "nasty" states in the reference failure set $\mathcal{S}$ that include several links affected with failure ratio 0.50 and even links whose entire capacity is lost. For those states the traffic matrix cannot be realized even for large values of K. As a consequence, protection for these two peculiar states, corresponding to about the 0.08% of all states, can never be granted, even imposing protection against the state with all links failing with failure ratio 0.25. Nevertheless, Table 2 gives some evidence that considering K-sets for network dimensioning instead of all the states in the reference failure set $\mathcal{S}$ can be effective in terms of the capability of traffic handling in the states outside those induced by the K-set. This is important, as the list of all possible states is in general not known or difficult to retrieve from available historical data.

We mention here that in Section 2 of the conference version of this paper (i.e., in [31]), a reasonable ad-hoc method for estimating the proper value of K to be used in network dimensioning is described. By the proper value of K we mean the minimum K that ensures the maximum achievable coverage (as $K = 7$ in Table 2). The method works for both the directed and undirected cases (see Section 6.1), but not for the bi-directional links case assumed in the above discussed study, and, for that matter, in the study presented below.

7.2. Study of Paris Metropolitan Area Network

In this section we present results of applying our robust optimization approach to a realistic FSO network instance. We call this instance PMAN (Paris Metropolitan Area network) since it was created using the real data for the Paris metropolitan area (summarized in Appendix A). However, although all the data used to design the PMAN instance are real, the instance itself is not real, and has been elaborated only for the purpose of this paper. This time, as we consider a realistic network instance, we consider the MIP version of problem P($\mathcal{S}$) with modular links (i.e., integer y). Because of that the optimization process involves both phases of Algorithm 2.

7.2.1. Experiment layout

In the PMAN case study we have considered the FSO transport system equipment provided by LightPointe Company [17], and more precisely the LightPointe's AireLink 80 10Gig

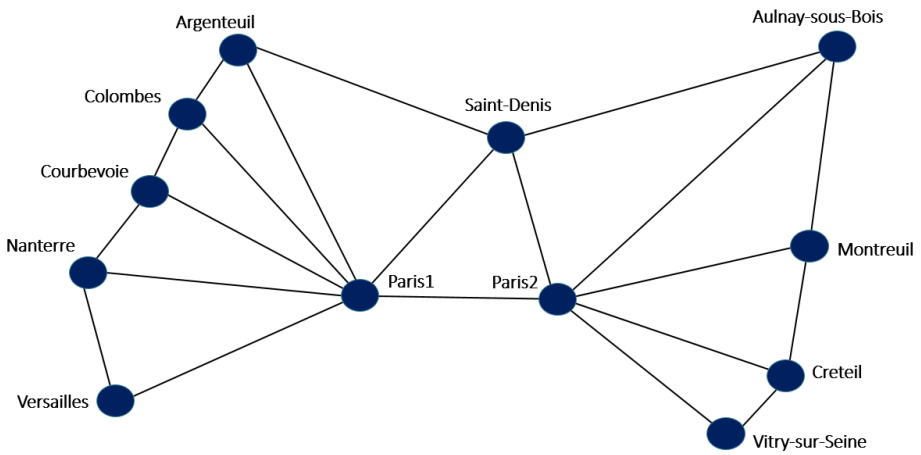

Figure 3: Paris Metropolitan Area Network.

system. The system requires one radio at each side of the FSO link and achieves 10 Gbps full duplex data transmission rate, which implies the link capacity module M equal to 10 Gbps (in each direction). As the considered system allows for the maximum distance of 15 km between two antennas linked by a light beam under clear weather conditions, the 15 km link range is assumed as well. As for *polska*, the cost of one (10 Gbps) module on each link is equal to 1 (i.e., $\xi(e) = 1$, $e \in \mathcal{E}$). This means that the cost of the network defined in (1a) expresses the total number of capacity modules (i.e., FSO systems) installed in the network.

The topology of PMAN is depicted in Figure 3. The network is composed of $V = 12$ (core) nodes and covers an area of around 250 km^2. Two nodes (Paris1 and Paris2) are located in the inner Paris area, and the remaining 10 nodes represent autonomous cities around it. The set of links is composed of all possible links whose end nodes are within 15 km in the line of sight (the distances between node/cities are given in Table A.4), resulting in $E = 21$ links (and $A = 42$ arcs) in total.

The traffic matrix for PMAN includes all $D = V(V-1) = 132$ directed traffic demands (expressed in [Gbps]) and is given in Table A.6 in Appendix A. The traffic matrix was calculated using formula (A.1) discussed in Appendix A.

We considered the weather conditions observed during all hours in a one year period, this time from January 1, 2016 until December 31, 2016 available at *www.worldweatheronline.com*. As for *polska*, the observed conditions were translated into the corresponding failure states. In effect, we obtained the set of all $366 \times 24 = 8784$ hourly failure states (2016 was a leap-year) – this full set of hourly states will be denoted by $\widetilde{\mathcal{S}}$. It happens that among these 8784 states, there are 1253 (14.26%) of the so called *disconnected states*. Each such state contains a set of links with failure ratio equal to 1 (i.e., totally failing) that is a cut that splits the network into separate (connected) components so that some of the demands cannot be realized at all.

Besides, in the study we will consider a subset of $\widetilde{\mathcal{S}}$, namely the set $\widetilde{\mathcal{S}}'$ of all $8784-1253 = 7531$ *connected states* obtained by deleting the disconnected states from $\widetilde{\mathcal{S}}$. Among the 7531 connected hourly states in the reduced set $\widetilde{\mathcal{S}}'$, 68.46% are nominal states with no degraded links. As for *polska*, the majority of the remaining states are characterized only by the 16-

QAM to 4-QAM change in MCS that reduces the nominal capacity of the affected links by 25% (link failure ratio 0.25) – there are 20.95% of such states in $\widetilde{\mathcal{S}}'$. In the remaining 10.59% of states there are links degraded by the failure ratio 0.5 and/or 1.

As far as the average connected hourly state in $\widetilde{\mathcal{S}}'$ is concerned, there are 94.7% of non-affected links, 4.3% of links with the failure ratio 0.25, 0.5% of links with the failure ratio 0.5, and 0.5% of totally failed links (failure ratio 1). Finally, we observed that out of all the 7531 connected hourly states there are only 206 states that are mutually different – these states constitute the reference failure set $\mathcal{S}$ for PMAN.

7.2.2. Algorithm efficiency

Table 3 presents numerical results illustrating the efficiency of Algorithm 2 for PMAN. The results are analogous to those for *polska* – the main difference is that for PMAN the modular links are considered, while for *polska* the link capacities were assumed to be continuous. This is reflected in Table 3 where the group of three columns under the common name "Phase 1" describe the optimal solutions (for the K-sets with $K = 0, 1, \ldots, 21$) of the linear relaxation of the considered MIP problem (1), obtained by means of Phase 1 of Algorithm 2. The column "Phase 1+2" shown in the Table, in turn, describes the computational time for modular solutions for PMAN obtained through applying Phase 2 after Phase 1 in Algorithm 2.

The solutions of the linear relaxation of P(K) obtained with Phase 1 behave similarly as the solutions for *polska*. The cost becomes stable already for $K \geq 6$, and the computation time, after initial increase and achieving maximum for $K = 11$, starts to decrease (with some exceptions).

As far as the results for the modular links case are concerned, we first note that for $K = 0$ (only the nominal state considered) the number of additional cuts (added in Phase 2) and the computation time become substantial. This is implied by a particular nature of the polyhedron of all feasible y vectors, different than that of the corresponding polyhedra for $K \geq 1$. In the modular case the behavior of the solutions for $K \geq 1$ is basically similar to that of the linear relaxation assumed in Phase 1. In this case the cost stabilizes at $K = 3$ while the maximum computation time is observed for $K = 17$.

We have also optimized the modular case for $K = 0$ and $K = 1$ directly, by applying the CPLEX MIP solver to formulation (1) with an explicit list of states. Clearly, for $K = 0$ such a list contains only the nominal state while for $K = 1$ it contains all single link degradation states with only one link degraded at a time (with the failure ratio 0.25). In the first case it took only 5 seconds to reach the optimal solution with $C^* = 32$ while in the second case it took as much as 1313 seconds reach the optimal solution with $C^* = 35$. This illustrates the fact that for $K \geq 1$ the cut-based method of Algorithm 2 is superior to the direct approach.

Finally, let us note that most of the algorithm's execution time (roughly 90%) has been spent in the cut generation subproblem, i.e., in the feasibility test solved in Step 2 of Algorithm 2. This confirms the remark at the beginning of Section 6.2 stating that extra valid inequalities are worth considering to speed-up the convergence of the algorithm.

7.2.3. Robustness with respect to the hourly states

Figures 4 and 5 show to what extent the network with modular links dimensioned for the consecutive K-sets (with the number of modules given in column 5 of Table 3) is able

K	Phase 1			Phase 1+2		
	C^*	no. of cuts	time [s]	C^*	no. of cuts	time [s]
0	25.2	92	13.0	32	2213	1398.1
1	30.5	101	20.3	35	106	21.9
2	32.2	116	23.6	36	140	28.7
3	33.1	143	35.0	37	143	36.5
4	33.4	133	31.0	37	135	31.5
5	33.5	134	31.0	37	135	32.0
6	33.6	148	35.0	37	150	35.4
7	33.6	147	36.0	37	148	36.6
8	33.6	153	37.0	37	157	38.8
9	33.6	154	37.0	37	159	39.0
10	33.6	147	34.0	37	163	37.0
11	33.6	166	41.0	37	185	45.0
12	33.6	152	35.0	37	172	39.0
13	33.6	156	31.0	37	262	54.0
14	33.6	172	34.0	37	248	47.1
15	33.6	174	31.2	37	256	48.0
16	33.6	150	26.0	37	242	42.0
17	33.6	168	29.0	37	315	62.0
18	33.6	171	28.0	37	260	43.0
19	33.6	170	27.0	37	257	45.0
20	33.6	174	27.0	37	251	42.0
21	33.6	167	24.2	37	236	36.5

Table 3: PMAN – results of robust optimization.

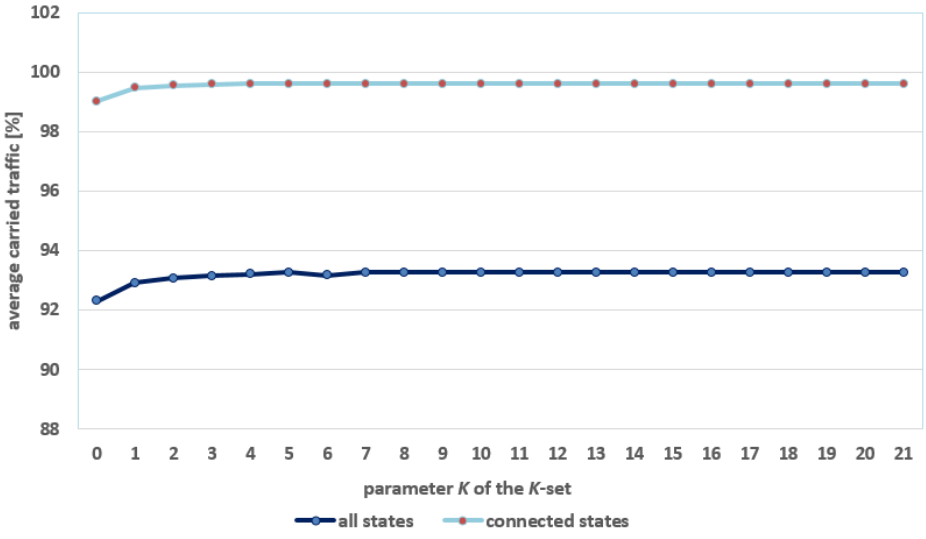

Figure 4: PMAN – states that are not covered with Algorithm 2.

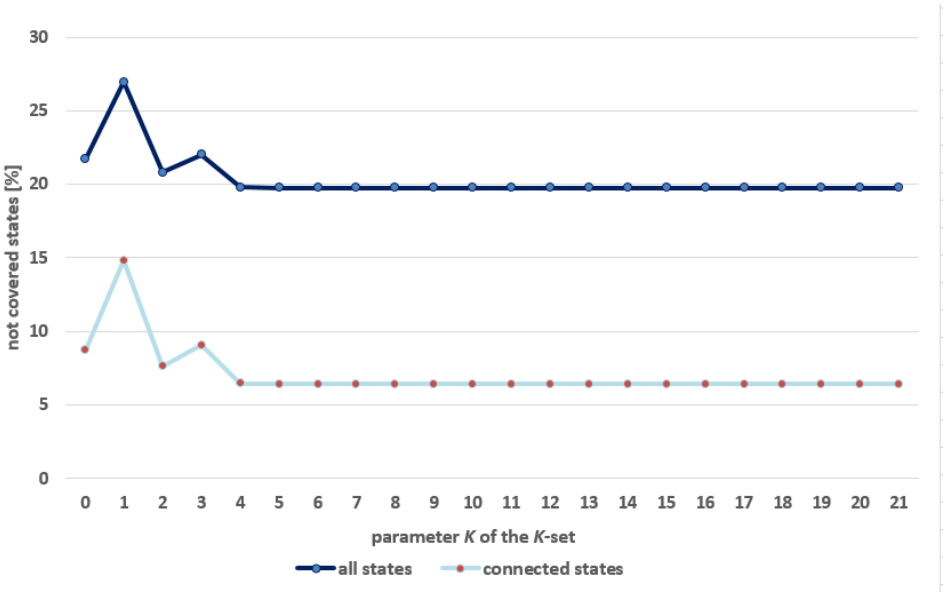

Figure 5: PMAN – results of robust optimization obtained with Algorithm 2.

to cover all 8784 hourly states (labelled with "all states", i.e., the states in $\widetilde{\mathcal{S}}$), and for all connected hourly states (labeled with "connected states", i.e., the states in $\widetilde{\mathcal{S}}'$). The figures for "average carried traffic [%]" and "not covered states [%]" were obtained in the same way as for *polska* (see Section 4.3). The figures reveal that the coverage for the connected states case is very good already for $K = 0$ (this is due to links' modularity since, typically, not all modules are fully utilized), and becomes stable for $K \geq 5$. These results are quite promising, as we may expect that the network dimensioned for a K-set even with a small value of K will cover, to a reasonable extent, also the states that were not observed in the historical data or are difficult to extract from such data.

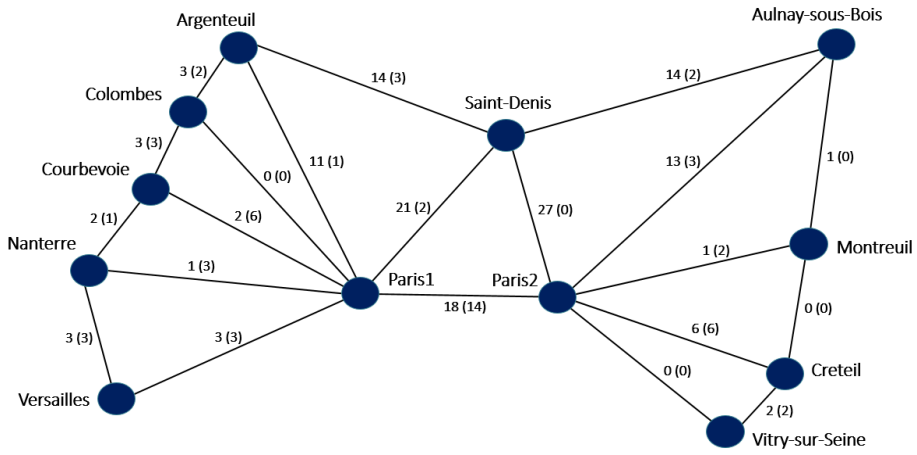

Figure 6: PMAN – link capacity.

7.2.4. An exact solution

For PMAN we have also computed the exact solution covering the whole states in the reference failure set $\mathcal{S}$ (with 206 distinct states). We have achieved that by solving the MIP formulation (1) of P(S) directly, and not using Algorithm 1. Such an optimal solution, obtained in 407 seconds, has the minimum cost equal 145 (recall that the assumed module costs $\xi(e) = 1$, $e \in \mathcal{E}$, imply that this cost is equal to the total number of modules installed on the links). This is a large difference as compared to the cost equal to 37 obtained for the K-sets with $K \geq 3$ that cover 99.60% of traffic and 19.75% of states. In fact, this is mainly due to the large amount of traffic traversing the link between Paris1 and Paris2 that is relatively often affected, sometimes totally. To restore the traffic lost on this link, a substantial additional (protection) capacity is necessary not only on this link but also on the routes around it. This effect is shown in Figure 6, where the pair of figures (one in brackets) next to a given link specifies its capacity expressed as the number of 10 Gbps modules. The first figure (not in brackets) is the link capacity when all links are the FSO links, while the second (in brackets) is the link capacity when Paris1-Paris2 is a fiber link and the remaining links are FSO links.

The above result shows that building a 100% robust FSO network can be very costly. A natural remedium for this issue (considered for example in [14]) would be to use a terrestrial fiber technology for some selected (bottleneck) links instead of the wireless FSO technology since the fiber links are not sensitive to weather conditions and thus can be assumed to be never degraded in our model (failure ratio always equal to 0). For example, assuming that the Paris1-Paris2 link is realized on a fiber (and the rest 20 links are FSO links), the optimum solution of P(S) requires 56 modules instead of 145, that is, as much as 91 modules less than required in the pure FSO network.

8. Concluding remarks and future work

In the paper we have presented a robust optimization approach to modular dimensioning of FSO networks taking into account degradation of link capacities due to weather conditions. Central to the approach is a cut-generation algorithm (Algorithm 2) for minimizing the cost of links for a network robust to all link failure states (where different link losses different portions of their nominal capacity) described by a given K-set (uncertainty polytope). In order to optimize a realistic network, we first analyze the weather data and produce a representative set of its link failure states (called the reference failure set). Then we dimension the network (using Algorithm 2) assuming uncertainty K-sets for several values of parameter K and check to what extent the states in reference failure set are covered by the K-set based solution. Using a realistic FSO network instance (based on the Paris metropolitan area data) we have shown that our approach is effective.

As suggested in the paper, the following extensions/enhancements of the presented approach are worth considering.

- Extending the K-sets by the multi failure rate case (see Section 6.3) to improve the coverage of the real states.
- Enhancing the optimization procedure by using valid inequalities (see Section 6.2) to speed up the algorithm convergence.

- Developing a heuristic algorithm for (suboptimal) rounding off the continuous link capacity solution to a modular solution. This will speed up the optimization process as Algorithm 2 will be applied only to the linear relaxation of the main problem (through Phase 1 of Algorithm 2).
- Adding fiber (non-failing) links to the optimization model, and analyzing the increase in cost efficiency of such obtained hybrid FSO/fiber networks (an initial study along these lines is presented in [33]).
- Applying the proposed approach to a realistic protection mechanism called Flow Thinning described in [13].

These improvements will be subject of our research in the near future (some of them are already being elaborated).

Acknowledgement: The work of the authors from Université de technologie de Compiègne was carried out in the framework of the Labex MS2T funded by the French Government through the program "Investments for the future" managed by the National Agency for Research [Reference ANR-11-IDEX-0004-02], and was partially financed by the PGMO Project - Programme Gaspard Monge. M. Pióro, I. Kalesnikau and A. Tomaszewski were supported by the National Science Center (Poland) [Grant 2015/17/B/ST7/03910 "Logical tunnel capacity control – a traffic routing and protection strategy for communication networks with variable link capacity"]. Research of M. Poss was supported by the ANR project ROBUST [ANR-16-CE40-0018].

References

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

[2] A. Sen, S. Murthy, S. Banerjee, A new paradigm for the design of fault-tolerant networks, in: Proc. of the 15th International Conference on High-Performance Switching and Routing (HPSR09), 2009, pp. 1–7.

[3] J. Rak, Measures of region failure survivability for wireless mesh networks, Wireless Networks 21 (2015) 673684.

[4] J. Rak, A new approach to design of weather disruption-tolerant wireless mesh networks, Telecommunication Systems 61 (2016) 311–323.

[5] F. Gjata, Measuring the impact of weather factors to FSO transmission, Intership Report, Information and Systems Technologies, Universit de Technologie de Compigne.

[6] M. Pióro, D. Medhi, Routing, Flow, and Capacity Design in Commu- nication and Computer Networks, Morgan Kaufmann, 2004.

[7] T. Bauschert, C. Büsing, F. D'Andreagiovanni, A. Koster, M. Kutschka, U. Steglich, Network planning under demand uncertainty with robust optimization, IEEE Commununications Magazine 52 (2014) 178–185. doi:10.1109/MCOM.2014.6736760.

[8] F. D'Andreagiovanni, J. Krolikowski, J. Pulaj, A fast hybrid primal heuristic for multiband robust capacitated network design with multiple time periods, Appl. Soft Comput. 26 (2015) 497–507. doi:10.1016/j.asoc.2014.10.016.

[9] A. Koster, M. Kutschka, C. Raack, Robust Network Design: Formulations, Valid Inequalities, and Computations, Networks 61 (2) (2013) 128–149.

[10] F. D'Andreagiovanni, Revisiting wireless network jamming by SIR-based considerations and multiband robust optimization, Optim. Lett. 9 (2015) 1495–1510. doi:10.1007/s11590-014-0839-2.

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

[12] G. Classen, D. Coudert, A. Koster, N. Nepomuceno, A chance-constrained model and cutting planes for fixed broadband wireless networks, in: Proc. INOC 2011, 2011.

[13] M. Pióro, Y. Fouquet, D. Nace, M. Poss, Optimizing flow thinning protection in multicommodity networks with variable link capacity, Operations Research 64 (2) (2016) 273–289.

[14] Y. Li, N. Pappas, V. Angelakis, M. Pióro, D. Yuan, Optimization of free space optical wireless network for cellular backhauling, IEEE JSAC 33 (9) (2015) 1841–1854.

[15] Y. Fouquet, D. Nace, M. Pióro, M. Poss, An optimization framework for traffic restoration in optical wireless networks with partial link failures, Optical Switching and Networking 23, Part 2 (2017) 108–117.

[16] FSona optical wireless. http://www.fsona.com/product.php.

[17] LightePointe AireLink 80 10 Gbps 70/80GHz Radios, https://www.azenn.com/img/cms/Promotions/Lightpointe/Lancement-10Gbps/AireLink_80_10Gig.pdf.

[18] A. Svensson, An introduction to adaptive qam modulation schemes for known and predicted channels, Proceedings of the IEEE 95 (12) (2007) 2322–2336. doi:10.1109/JPROC.2007.904442.

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

[20] J. Priyanka, S. Bhuperdra, C. Rashmi, Survey on performance of free space optical communication links under various field parameters, IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) 9 (2) (2014) 71–75.

[21] J. Benders, Partitioning procedures for solving mixed variable programming problems, Numerische Mathematik 4 (1962) 238–252.

[22] J. Ayoub, M. Poss, Decomposition for adjustable robust linear optimization subject to uncertainty polytope, Computational Management Science 2 (2013) 219–239.

[23] B. Korte, J. Vygen, Combinatorial Optimization: Theory and Algorithms, Springer-Verlag, 2012.

[24] C. Chekuri, F. Shepherd, G. Oriolo, M. Scutellá, Hardness of robust network design, Networks 50 (1) (2007) 50–54.

[25] S. Khot, On the unique games conjecture (invited survey), in: Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, USA, June 9-12, 2010, 2010, pp. 99–121.

[26] G. Nemhauser, L. Wolsey, Integer and Combinatorial Optimization, J. Wiley, 1999.

[27] D. Nace, M. Pióro, A. Tomaszewski, M. Żotkiewicz, Complexity of a classical flow restoration problem, Networks 62 (2) (2013) 149–160.

[28] S. Mattia, M. Poss, A comparison of different routing schemes for the robust network loading problem: polyhedral results and computation, Computational Optimization and Applications (in press).

[29] M. Fischetti, I. Ljubic, M. Sinnl, Redesigning Benders Decomposition for Large-Scale Facility Location, Management Science 63 (7) (2017) 2146–2162. doi:10.1287/mnsc.2016.2461.
URL https://doi.org/10.1287/mnsc.2016.2461

[30] S. Orlowski, M. Pióro, A. Tomaszewski, R. Wessäly, SNDlib 1.0 – Survivable Netwrok Design Library, Networks 55 (3) (2010) 276–285.

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

[32] M. Poss, C. Raack, Affine recourse for the robust network design problem: Between static and dynamic routing, Networks 61 (2) (2013) 180–198. doi:10.1002/net.21482.
URL https://doi.org/10.1002/net.21482

[33] M. Shehaj, D. Nace, I. Kalesnikau, M. Pióro, Dimensioning of hybrid FSO/fiber networks, in: Proc. BalkanCom 2018, Podgorica, Montenegro, 2018.

[34] Chiffres clefs de la Famille 2016, http://www.udaf42.org/sites/default/files/Chiffres_cles_2016.pdf.

[35] Key numbers of Ile-de-France region, http://www.cci-paris-idf.fr/etudes/organisation/crocis/chiffres-cles/chiffres-cles-region-ile-de-france-crocis.

Appendix A. Paris Metropolitan Area network – data used in the study

The Paris Metropolitan Area network (PMAN) example was prepared on the basis of publicly available data. The description of the PMAN is summarized below.

cities	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
(1) Paris1	–	10.0	8.8	10.0	10.2	8.2	9.0	12.4	13.2	16.0	11.6	18.9
(2) Paris2	10.0	–	8.8	15.0	15.2	13.2	16.3	22.4	8.2	8.0	6.6	9.0
(3) Saint Denis	8.8	8.8	–	8.1	7.7	8.7	12.1	22.4	16.7	18.3	10.2	10.7
(4) Argenteuil	10.0	15.0	8.1	–	2.8	5.9	6.9	18.4	20.7	23.6	17.0	18.5
(5) Colombes	10.2	15.2	7.7	2.8	–	3.6	4.8	16.2	18.1	21.2	15.2	18.0
(6) Courbevoie	8.2	13.2	8.7	5.9	3.6	–	3.6	13.9	15.6	19.0	14.1	18.3
(7) Nanterre	9.0	16.3	12.1	6.9	4.8	3.6	–	11.5	17.8	21.6	17.5	21.9
(8) Versailles	12.4	22.5	22.4	18.4	16.2	13.9	11.5	–	19.5	23.9	23.9	30.9
(9) Vitry-Sur-Seine	13.2	8.2	16.7	20.7	18.1	15.6	17.9	19.5	–	4.4	9.1	18.1
(10) Creteil	16.0	8.0	18.3	23.6	21.2	19.0	21.6	23.9	4.4	–	8.9	17.2
(11) Montreuil	11.6	6.6	10.2	17.0	15.2	14.1	17.5	23.9	9.1	8.9	–	9.1
(12) Aulnay-Sous-Bois	18.9	9.0	10.4	18.5	18.0	18.3	21.9	30.9	18.1	17.2	9.1	–

Table A.4: Distances between cities [km].

Table A.4 gives the line-of-sight distance (measured on the map) between all the pairs of the 12 cities composing PMAN. Clearly, the matrix is symmetric and its entries imply the

set of 21 potential (undirected) links of PAM that is depicted in Figure 3: a link between a pair of cities can be provided if the distance between its end nodes does not exceed 15 km.

cities Ile-de-France	population 2016	number of families	DL traffic per family [Mbps]	UL traffic per family [Mbps]	aggregated DL traffic [Gbps]	aggregated UL traffic [Gbps]
Paris1	1 099 921	274 980.25	28.71	21.65	7893.31	5954.70
Paris2	1 099 921	274 980.25	28.71	21.65	7893.31	5954.70
Saint-Denis	116 302	29 075.50	19.14	10.23	556.42	297.36
Argenteuil	107 408	26 852.00	11.71	4.75	314.46	127.55
Colombes	88 257	22 064.25	30.60	25.04	675.14	522.58
Courbevoie	88 776	22 194.00	30.60	25.04	879.11	555.83
Nanterre	92 607	23 151.75	30.60	25.04	708.42	579.81
Versailles	83 879	20 974.25	30.50	13.16	639.78	275.98
Vitry-Sur-Seine	92 791	23 197.75	32.50	13.16	753.53	305.24
Creteil	90 282	22 570.50	32.50	13.16	733.16	296.98
Montreuil	104 963	26 240.75	19.14	10.23	502.17	268.36
Aulnay-Sous-Bois	82 513	20 628.25	19.14	10.23	394.76	210.97

Table A.5: Demographic data and traffic estimates.

The next table (Table A.5) contains the data necessary to prepare the traffic matrix for PMAN. The second column gives the population in the areas served by each node (*http://population.city/france*). Assuming that the average family consists of 4 persons (actually, according to [34], the real figure is 3.85), we calculated the number of families in each area (column 3). Next, we took the estimations of the average rate of the traffic downloaded (DL, column 4) and uploaded (UL, column 4) from/to the Internet per family in each area (using the data available at [35]), and finally computed the aggregated DL (column 6) and UL (column 7) traffic rate from/to the Internet per the whole areas. In the calculations we considered only the residential traffic, as for instance VoIP and peer-to-peer traffic, assuming that (i) the considered traffic is equal to 10% of the whole generated traffic, and (ii) only 10% of population is served by this network.

Using the data from Table A.5, we calculated the (directed) traffic matrix $T = [T(i,j)]_{i,j=1}^{12}$ for PMAN applying the following formula for all directed pairs (i,j), $i,j = 1,2,\ldots,12$, $i \neq j$:

$$T(i,j) = UL(i) \cdot \frac{P(j)}{\sum_{k=1}^{12} P(k) - P(i)} + DL(j) \cdot \frac{P(i)}{\sum_{k=1}^{12} P(k) - P(j)}. \tag{A.1}$$

In the formula, $P(k)$, $DL(i)$ and $UL(i)$ denote the entries in columns 2, 6 and 7, respectively, of Table A.5. The formula is based on the assumption that the (directed) traffic demand from city i to city j is estimated as the sum of a fraction of the upload (UL) traffic generated at city i (this fraction is proportional to the share of the population of the area j in the total population of PMAN outside the area of i), and a fraction of the download (DL) traffic received at city j (this fraction is proportional to the share of the population of the area of i in the total population of PMAN outside the area of j).

cities	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
(1) Paris1	–	74.38	5.40	4.26	4.99	5.02	5.24	4.74	5.41	5.26	4.87	3.82
(2) Paris2	74.38	–	5.40	4.26	4.99	5.02	5.24	4.74	5.41	5.26	4.87	3.82
(3) Saint-Denis	5.56	5.56	–	0.23	0.34	0.35	0.36	0.33	0.38	0.37	0.29	0.23
(4) Argenteuil	4.60	4.60	0.25	–	0.27	0.28	0.29	0.26	0.30	0.30	0.22	0.01
(5) Colombes	5.39	5.39	0.37	0.29	–	0.36	0.37	0.34	0.39	0.37	0.34	0.26
(6) Courbevoie	5.42	5.42	0.37	0.29	0.36	–	0.37	0.34	0.39	0.38	0.34	0.26
(7) Nanterre	5.66	5.66	0.39	0.30	0.37	0.37	–	0.34	0.40	0.39	0.35	0.28
(8) Versailles	4.22	4.22	0.26	0.18	0.26	0.27	0.28	–	0.29	0.28	0.23	0.18
(9) Vitry-Sur-Seine	4.68	0.13	0.29	0.20	0.29	0.29	0.31	0.28	–	0.31	0.26	0.20
(10) Creteil	4.55	4.55	0.28	0.20	0.28	0.29	0.30	0.27	0.31	–	0.25	0.20
(11) Montreuil	5.02	5.02	0.30	0.20	0.31	0.31	0.33	0.29	0.34	0.33	–	0.21
(12) Aulnay-Sous-Bois	3.94	3.94	0.23	0.16	0.24	0.24	0.26	0.23	0.27	0.26	0.21	–

Table A.6: Traffic matrix [Gbps].