A Chance Constraint Model for Multi-Failure Resilience in Communication Networks

C. Helmberg, S. Richter, D. Schupke

Preprint 2015-9

Preprintreihe der Fakultät für Mathematik
ISSN 1614-8835

A Chance Constraint Model for Multi-Failure Resilience in Communication Networks*

Christoph Helmberg, Sebastian Richter†
and Dominic Schupke‡

June 25, 2015

Abstract

For ensuring network survivability in case of single component failures many routing protocols provide a primary and a back up routing path for each origin destination pair. We address the problem of selecting these paths such that in the event of multiple failures, occuring with given probabilities, the total loss in routable demand due to both paths being intersected is small with high probability. We present a chance constraint model and solution approaches based on an explicit integer programming formulation, a robust formulation and a cutting plane approach that yield reasonably good solutions assuming that the failures are caused by at most two elementary events, which may each affect several network components.

Keywords: robust optimization, stochastic programming, network design, network survivability

MSC 2000: 90B18, 90B25, 90C15, 90C35

1 Introduction

Designing and planning networks is an important and complex task and a frequently occuring topic in the optimization community due to its large

*This work was supported by the *Bundesministerium für Bildung und Forschung* under grant 05M10OCB. Responsibility for the content rests with the authors.

†Fakultät für Mathematik, Technische Universität Chemnitz, D-09107 Chemnitz, Germany. {helmberg, sebastian.richter}@mathematik.tu-chemnitz.de

‡Airbus Group Innovations, Munich, Germany.

variety of applications, e.g. in transportation and communication networks. In this, handling uncertainties is a central issue. There are two main sources of uncertainty, one is the development of demand over time, the other is network resilience. Both need to be dealt with in order to cost efficiently provide a quality of service guarantee. The focus here is on network resilience and is motivated by the following observation in operating communication networks today. In routing protocols of communication networks each origin destination pair is assigned a primary and a node disjoint back up routing path, so that single component failures lead to no additional loss. The paths, however, are often selected without further safety considerations. In the case of two failures frequently more than half of the entire demand cannot be routed over any of these paths even though most of the origin destination pairs are still connected in the original graph. The task is thus to find for each origin destination pair a pair of node disjoint paths (or more generally a routing subgraph), so that in the event of multiple disruptions occuring with given probabilities the total loss of demand over all origin destination pairs that are still connected in the network is small with high probability.

While we address this question with respect to a given fixed demand, several basic techniques that are useful in our context have been developed for dealing with demand uncertainty. In this field there is a lot of (ongoing) progress using the concept of Robust Optimization that was established by Soyster [20] in 1973 and was further developed by Ben-Tal and Nemirovski ([2], [3], [4]) and Bertsimas and Sim [6]. For recent work concerning uncertainty in demand see for example [12] or [5]. In the case of wireless networks Koster et al. [9] used a chance-constrained approach to find a minimum cost design of a fixed broadband wireless network such that the capacity is sufficient with a certain probability. The concept of chance constraints has been introduced by Charnes, Cooper and Symonds [8] in 1958 and was extended by Miller and Wagner [15] and Prékopa ([17], [18]). In contrast to Robust Optimization where a solution is called robust if it is feasible for any realization of the uncertain data, in Stochastic Programming a feasible solution has to satisfy the constraints that are influenced by uncertain data with a given probability.

The setting of this paper consists of a telecommunication network with given demands where the components (nodes and edges) of the network may fail due to technical problems or physical influences like intersection of fibers or ducts containing several fibers. Depending on the selected sets of routing paths or the capacities of the connections, these failures can lead to a loss in data (demand). Our goal is to determine a routing (and therefore determine the necessary capacities of the fibers) such that the costs to install the needed capacities are minimized subject to the constraint that the probability of loosing too much data is reasonably small. We present three different approaches to (approximately) solve the associated chance-constrained optimization problem. The first proposes a direct integer programming (IP) formulation of the chance constraint. The second uses a robust approach via a sufficient condition for solutions to satisfy the chance constraint. Finally, we discuss a cutting plane approach that might be better suited for solving the problem exactly if the direct IP formulation is excessive in size.

The paper is organized as follows. Section 2 explains the basic model for the interaction of network, disruptions and routings; this results in a chance constrained optimization problem (2) for selecting routing subgraphs for each origin destination pair. Section 3 provides an illustrative example on how such a disruption model with corresponding probabilities could be set up for real world purposes. Section 4 gives an explicit IP-formulation of the chance constraint based on introducing binary variables for all relevant events. For larger instances this might no longer be feasible, so Section 5 presents a robust counterpart to (2) in the style of Ben-Tal and Nemirovski [1] or Bertsimas and Sim [6], where the chance constraint is replaced by a slightly more conservative convex model, resulting in a problem that is faster to solve in practice. Section 6 is devoted to a further possibility to solve (2) without introducing additional binary variables by enforcing the chance constraint via a cutting plane approach for integer solutions. Some preliminary numerical results discussing the effectiveness of the proposed approaches on some examples of SND-LIB [16] are presented in Section 7.

2 Model Description

A network topology is given by a graph $G = (V, E)$ with node set $V := \{1, \dots, n\}$ (e.g. servers) and edge set $E \subseteq \{\{i, j\} \colon i, j \in V, i \neq j\}$ (e.g. optical fibers). For each ordered pair $(i, j) \in W := \{(i, j) \colon i, j \in V, i \neq j\}$ of nodes the (possibly uncertain) demand to be routed from node i to node j over edges in G is denoted by d_{ij}. For brevity, we will simply write ij instead of (i, j) or also instead of $\{i, j\}$ if there is no danger of confusion.

Failures or disruptions may affect subsets of nodes or edges, *i. e.* any subset of $\mathcal{G} := V \cup E$, and will be modeled as follows. For a given set of elementary disruptions $\overline{R} = \{r_1, \dots, r_{n_{\overline{R}}}\}$ a function $F\colon \overline{R} \to 2^{\mathcal{G}}$ specifies which nodes or edges in $\mathcal{G}$ fail if disruption $r \in \overline{R}$ occurs. These disruptions may be considered to be, *e. g.*, the failure of a single component of $\mathcal{G}$ or the intersection of a single duct (which effects all fibers within it) due to construction works or maybe an earthquake scenario affecting several ducts.

Example 1 *Consider Figure 1. The thin lines represent the fibers/edges*

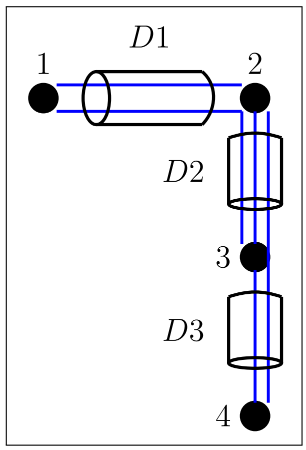


Figure 1: Network section consisting of 4 nodes, 3 ducts and 5 fibers.

which connect the nodes of the network and are embedded in the three ducts $D1$*,* $D2$ *and* $D3$*. If a node* v *is not available due to technical problems then no data can be sent to or from this node. The remaining graph,* i. e.*, the network containing only available components is obtained by deleting the set* $F(v) = \{v, \{v, u\} \in E\}$ *from* G *which comprises the corresponding node as well as all incident edges. A failing fiber* $\{u, v\}$ *only deletes the corresponding edge* $F(\{u, v\}) = \{\{u, v\}\}$*, a disrupted duct* d *removes all fibers/edges that go through it. If for example duct* $D2$ *is disrupted,* $F(D2) = \{\{1, 3\}, \{2, 3\}, \{2, 4\}\}$ *and there is no way to maintain the connection between 2 and 4. If, however, only the fiber* $\{2, 4\}$ *from 2 to 4 is cut, demand between 2 and 4 may still be routed from 2 to 3 to 4 or from 2 to 1 to 3 to 4.*

We assume the elementary disruptions of $\overline{R}$ to be independent, each $r \in \overline{R}$ occuring with probability $\pi_r \in (0, 1)$, so that for each combination of several elementary disruptions $R \in \mathcal{R} := 2^{\overline{R}}$ the probability that exactly the disruptions in R are occuring is $p_R := \prod_{r\in R} \pi_r \prod_{r\in\overline{R}\setminus R}(1-\pi_r)$. This $p \in [0,1]^{\mathcal{R}}$ satisfies $\mathbf{1}^\top p = 1$ and gives rise to our basic probability space $(\mathcal{R}, p)$. In this, each elementary event $R \in \mathcal{R}$ corresponds to the occurrence of the combination of elementary disruptions $R \subseteq \overline{R}$ and this has probability p_R. A general event in the usual probabilistic sense is a subset $\mathcal{S} \subseteq \mathcal{R}$, it takes place with probability $p(\mathcal{S}) = \sum_{R\in\mathcal{S}} p_R$. A disruption combination $R \in \mathcal{R}$ "deletes" all elements $F(R) := \bigcup_{r\in R} F(r)$ from G. The resulting graph is denoted by $G - F(R)$, where the deletion of a node also implies the loss of all its incident edges in G. In the presence of disruptions R the demand still routable in $G - F(R)$ is reduced to demand pairs $W(R) := \{ij \in W \colon i, j \notin F(R), i \text{ and } j \text{ are connected in } G - F(R)\}$ with a remaining routable demand of $d(R) = \sum_{ij\in W(R)} d_{ij}$. Note that determining $W(R)$ simply amounts to finding the connected components of $G - F(R)$.

The presence of elementary disruptions corresponding to single node failures or single edge failures due to technical failures in the corresponding hardware may be thought of as being represented by the requirement $V \cup E = \mathcal{G} \subseteq \overline{R}$. In a resilient routing such "simple disruptions", represented say by $R \in \underline{\mathcal{R}}$ for some appropriate collection $\underline{\mathcal{R}} \subset \mathcal{R}$, should not lead to loss of demand between nodes i and j as long as $ij \in W(R)$. In practice this is achieved for $\underline{\mathcal{R}} = \{\{a\} \colon a \in \mathcal{G} \subseteq \overline{R}\}$ by assigning to each demand pair $(i, j) \in W$ two node disjoint paths from i to j (one working, one as fall back) forming a cycle which connects i and j for any disruption $R \in \underline{\mathcal{R}}$. The task is then to select a suitable cycle for each pair so that as little routing as possible is lost in the entire network even if several disruptions occur. Say, we want to find a design that is able to handle all events in the set $\overline{\mathcal{R}} \subseteq \mathcal{R}$ reasonably well, then taking into account given probabilities p the goal for selecting the cycles may be formulated as a so called chance constraint as follows. The conditional probability that in the presence of exactly one of the disruption combinations $R \in \overline{\mathcal{R}}$ the amount routable along paths in the selected cycles falls below some $\sigma \in (0, 1)$-fraction of the remaining routable demand $d(R)$ should be less than some appropriately chosen $\varepsilon_{\overline{\mathcal{R}}} > 0$.

In order to formalize this in slightly greater generality, collect in $\underline{\mathcal{R}} \subset \mathcal{R}$ all simple disruptions like node and edge failures that should not cause any loss of demand unless they affect the origin or destination node directly. Assume furthermore that for each demand pair $ij \in W$ there is a set of $k_{ij} \in \mathbb{N}$ "routing subgraphs" $\mathcal{C}_{ij} = \{C_{ij}^k \colon k = 1, \ldots, k_{ij}, C_{ij} \subseteq G \text{ connects } i \text{ and } j\}$

(each of them may be thought of as a collection of ij-paths) with the property that for any single disruption combination $R \in \underline{\mathcal{R}}$ with $ij \in W(R)$, each of these subgraphs $C \in \mathcal{C}_{ij}$ still connects i and j by a path in $C - F(R)$. We collect all such index triples in $K = \{(i,j,k)\colon ij \in W, 1 \leq k \leq k_{ij}\}$ and use binary decision variables $x_{ij}^k \in \{0,1\}$ with $\sum_{k=1}^{k_{ij}} x_{ij}^k = 1$, a so called assignment, to select one out of these cycles for demand pair $ij \in W$. For a disruption combination $R \subseteq \overline{R}$ the triples that still allow to route demand are $K(R) := \{(i,j,k) \in K\colon C_{ij}^k - F(R) \text{ connects } i \text{ and } j\}$.

In order to allow explicit computation of the conditional probability that a particular disruption combination R in $\overline{\mathcal{R}}$ occurs if exactly one of them occurs, we will assume throughout that $\overline{\mathcal{R}}$ is rather small, *e. g.* that $\overline{\mathcal{R}}$ consists of all combinations of at most two elementary disruptions. Then the conditional probabilities are easily precomputed by

$$\hat{p}_R := \frac{p_R}{p(\overline{\mathcal{R}})} = \frac{p_R}{\sum_{R' \in \overline{\mathcal{R}}} p_{R'}} \qquad \text{for } R \in \overline{\mathcal{R}}.$$

This yields the vector $\hat{p} \in (0,1)^{\overline{\mathcal{R}}}$ with $\mathbf{1}^T \hat{p} = 1$ describing the required probability distribution for the chance constraint. Likewise, the routable demand $\hat{d}_R := d(R) = \sum_{ij \in W(R)} d_{ij}$ can be precomputed in advance for $R \in \overline{\mathcal{R}}$ and will be assumed to be available in the form of a vector $\hat{d} \in \mathbb{R}_+^{\overline{\mathcal{R}}}$.

For a given assignment $x \in \{0,1\}^K$ of routable graphs to demand pairs the amount still routable in view of disruption combination R is $\sum_{ijk \in K(R)} d_{ij} x_{ij}^k$. The disruption combinations in $\overline{\mathcal{R}}$ leading to a loss of more than a $(1-\sigma)$-fraction of the routable demand for assignment x form the event

$$\overline{\mathcal{R}}(x,\sigma) := \{R \in \overline{\mathcal{R}}\colon \sum_{ijk \in K(R)} d_{ij} x_{ij}^k < \sigma \hat{d}_R\}.$$

The chance constraint requires the probability of this event, *i. e.* the sum of the probabilities of these disruption combinations, to be at most $\epsilon_{\overline{\mathcal{R}}}$

$$\mathbb{P}_{R \in \overline{\mathcal{R}}} \left(\sum_{ijk \in K(R)} d_{ij} x_{ij}^k < \sigma \hat{d}_R \right) = \sum_{R \in \overline{\mathcal{R}}(x,\sigma)} \hat{p}_R \leq \epsilon_{\overline{\mathcal{R}}}. \tag{1}$$

For given $\sigma \in (0,1)$ and $\epsilon_{\overline{\mathcal{R}}} \in (0,1)$ the feasible assignments are thus

$$\mathcal{X}(\sigma, \epsilon_{\overline{\mathcal{R}}}) = \left\{ x \in \{0,1\}^K\colon \sum_{k=1}^{k_{ij}} x_{ij}^k = 1, ij \in W, \sum_{R \in \overline{\mathcal{R}}(x,\sigma)} \hat{p}_R \leq \epsilon_{\overline{\mathcal{R}}} \right\}.$$

For the choice of objective, one might either want to maximize the routable fraction σ, minimize the probability $\varepsilon_{\overline{\mathcal{R}}}$ or minimize the cost associated with installing sufficient capacities. Here, we concentrate on the latter and introduce, for each edge $e \in E$ variables $y_e \in \mathbb{Z}_+$ that give the amount of capacity installed. The constraint ensuring sufficient capacity reads

$$\sum_{e \in E(C_{ij}^k)} d_{ij} x_{ij}^k \leq y_e \qquad \text{for } e \in E.$$

Given cost functions $c_e \colon \mathbb{Z}_+ \to \mathbb{R}$ for $e \in E$ (they will be assumed to be linear), the problem to solve is

$$\begin{array}{rll} \text{minimize} & \sum_{e \in E} c_e(y_e) & \\ \text{subject to} & \sum_{1 \leq k \leq k_{ij}} x_{ij}^k = 1, & ij \in W, \\ & \sum_{e \in E(C_{ij}^k)} d_{ij} x_{ij}^k \leq y_e, & e \in E, \\ & \sum_{R \in \overline{\mathcal{R}}(x,\sigma)} \hat{p}_R \leq \varepsilon_{\overline{\mathcal{R}}}, & \\ & x \in \{0,1\}^K, y \in \mathbb{Z}_+^E. & \end{array} \tag{2}$$

In spite of its rather abstract setting, this is a realistic problem to solve directly for reasonably practical choices of subgraphs indexed by K, disruption scenarios $\underline{\mathcal{R}}, \overline{\mathcal{R}}$ and cost functions c.

3 A Simple Model for Computing Failure Probabilities

The task of this section is to illustrate how a sufficiently simple model of elementary disruptions $r \in \overline{R}$ could be set up together with appropriate probabilities, so that the resulting problem is still of practical relevance for actual networks. As the authors are more than aware that they are no experts in the field of network failures, the model largely relies on the studies in [22] and should not be understood as an attempt to form a highly realistic model with perfectly accurate parameters. The values that are typically used to describe the availability of technical components are MTBF and MTTR representing the mean time before failure and the mean time to repair in hours, respectively. Hence, the mean time between two failures of one component is MTBF + MTTR and the unavailability is given by $\frac{\text{MTTR}}{\text{MTTR+MTBF}}$. Therefore, the probability that a node or an edge is unavailable due to technical failures at some arbitrary time depends on the used components. For $i \in \mathcal{G}$ we have

$$\pi_{r_i} = \frac{\text{MTTR}_i}{\text{MTTR}_i + \text{MTBF}_i}.$$

The intersection of the ducts can be modeled in the same fashion. For this purpose it seems common to use a value CC describing the cable cuts, *i. e.*, the average length in km suffering from one cable-cut a year, see [22]. Defining $l_{\hat{\imath}}$ as the length of duct $\hat{\imath}$, the expected number of cuts in a time interval of length $t > 0$ is $\frac{l_{\hat{\imath}} t}{\text{CC}\cdot 24 \cdot 365}$, hence, $\text{MTTR}_{\hat{\imath}} + \text{MTBF}_{\hat{\imath}} = \text{MTTR}_{\hat{\imath}} + \frac{\text{CC}\cdot 24\cdot 365}{l_{\hat{\imath}}}$ in this case. Thus, if $r_{\hat{\imath}} \in \overline{\mathcal{R}}$ denotes the intersection of a duct (resulting in the intersection of several fibers/edges) the associated probability for this event is

$$\pi_{r_i} = \frac{l_{\hat{\imath}}\text{MTTR}_{\hat{\imath}}}{l_{\hat{\imath}}\text{MTTR}_{\hat{\imath}} + \text{CC} \cdot 24 \cdot 365}$$

assuming that $\text{MTTR}_{\hat{\imath}}$ is given in hours.

4 An explicit IP-Formulation of the Chance Constraint

If the size of $\overline{\mathcal{R}}$ is not too large so that introducing one binary variable $s_R \in \{0, 1\}$ for each $R \in \overline{\mathcal{R}}$ is an acceptable option, it is worth to consider the following explicit integer programming formulation of (2).

For a given assignment x let s_R represent the indicator variable for $R \in \overline{\mathcal{R}}(X, \sigma)$ by requiring the inequality

$$\sum_{ijk \in K(R)} d_{ij} x_{ij}^k + \sigma \hat{d}_R s_R \geq \sigma \hat{d}_R,$$

i. e. if the demand routed falls below the required percentage of routable demand in the presence of disruption combination R, s_R has to be set to one. The chance constraint is satisfied if those forced to one are bounded by

$\sum_{R \in \overline{\mathcal{R}}} \hat{p}_R s_R \leq \varepsilon_{\overline{\mathcal{R}}}$. Thus an explicit IP-formulation of (2) reads

$$\begin{array}{lll} \text{minimize} & \sum_{e \in E} c_e(y_e) & \\ \text{subject to} & \sum_{1 \leq k \leq k_{ij}} x_{ij}^k = 1, & ij \in W, \\ & \sum_{e \in E(C_{ij}^k)} d_{ij} x_{ij}^k \leq y_e, & e \in E, \\ & \sum_{ijk \in K(R)} d_{ij} x_{ij}^k + \sigma \hat{d}_R s_R \geq \sigma \hat{d}_R, & R \in \overline{\mathcal{R}}, \\ & \hat{p}^T s \leq \varepsilon_{\overline{\mathcal{R}}}, & \\ & x \in \{0,1\}^K, s \in \{0,1\}^{\overline{\mathcal{R}}}, y \in \mathbb{Z}_+^E. & \end{array} \tag{3}$$

Because the s_R are used as on/off indicator variables for the constraints, the corresponding linear programming relaxation may be expected to be rather weak and it is not quite clear for what sizes of $\overline{\mathcal{R}}$ this is a realistic option.

5 A Robust Approach to the Chance Constraint

In order to avoid the introduction of additional binary variables for the chance constraint of (2), the approach in this section formulates a sufficient condition for the assignments to satisfy the chance constraint. In general this condition is too strong and also excludes some feasible assignments, but if it still permits feasible solutions it has the advantage that it involves solving one integer program with a reasonable number of binary variables.

The starting point is an attempt to identify in program (2) for given x those events $R \in \overline{\mathcal{R}}$ of joint probability ε that are worst in terms of loss of demand. For this consider the program

$$\begin{array}{rll} \delta(x, \sigma, \epsilon) := & \text{maximize} & \sum_{R \in \overline{\mathcal{R}}} \eta_R (\sigma \hat{d}_R - \sum_{ijk \in K(R)} d_{ij} x_{ij}^k) \\ & \text{subject to} & \sum_{R \in \overline{\mathcal{R}}} \eta_R = \epsilon, \\ & & \eta_R \in [0, \hat{p}_R], \qquad R \in \overline{\mathcal{R}}. \end{array} \tag{4}$$

Theorem 2 *If $\delta(x, \sigma, \epsilon_{\overline{\mathcal{R}}}) \leq 0$ for a given assignment $x \in \{0,1\}^K$, then* (1) *holds.*

Proof. Let x be an assignment that violates (1) and let the elements of $\overline{\mathcal{R}}(x, \sigma) = \{R_1, \ldots, R_m\}$ be numbered so that the values

$$\delta_{R_h} := \sigma \hat{d}_{R_h} - \sum_{ijk \in K(R_h)} d_{ij} x_{ij}^k > 0$$

are sorted nonincreasingly.

$$\delta_{R_1} \geq \cdots \geq \delta_{R_m} > 0.$$

Because x violates (1) there is a smallest index $\bar{m} \leq m$ so that $\sum_{h=1}^{\bar{m}} \hat{p}_{R_h} > \epsilon_{\overline{\mathcal{R}}}$. Put $\eta_{R_h} = \hat{p}_{R_h}$ for $h = 1, \ldots, \bar{m}-1$, $\eta_{R_{\bar{m}}} = \epsilon_{\overline{\mathcal{R}}} - \sum_{h=1}^{\bar{m}-1} \hat{p}_{R_h}$ ($\leq \hat{p}_{R_{\bar{m}}}$ but ≥ 0) and put $\eta_R = 0$ for $R \in \overline{\mathcal{R}} \setminus \{R_1, \ldots, R_{\bar{m}}\}$. Then η is feasible, because

$$\epsilon_{\overline{\mathcal{R}}} = \sum_{h=1}^{\bar{m}} \eta_{R_h} = \sum_{R \in \overline{\mathcal{R}}} \eta_R,$$

and its objective value is positive because

$$0 < \sum_{h=1}^{\bar{m}} \eta_{R_h} \delta_{R_h} = \sum_{R \in \overline{\mathcal{R}}} \eta_R (\sigma \hat{d}_R - \sum_{ijk \in K(R)} d_{ij} x_{ij}^k).$$

■

The conservatism of the condition $\delta(x, \sigma, \epsilon_{\overline{\mathcal{R}}}) \leq 0$ may be explained as follows. Disruption combinations $R \in \overline{\mathcal{R}}(x, \sigma)$ with small probability $\hat{p}_R$ may already contribute a strong positive term if their excess loss $\sigma \hat{d}_R - \sum_{ijk \in K(R)} d_{ij} x_{ij}^k$ is quite large and it is only possible to reach a nonpositive objective if the equation for ϵ forces the compensation of this by including the next worst disruption combinations whose excess loss is already negative. In order to mitigate this effect it is conceivable to require $\delta(x, \sigma, \epsilon_{\overline{\mathcal{R}}})$ to be below some small positive number or to use $\delta(x, \sigma, \epsilon) \leq 0$ for some parameter $\epsilon > \epsilon_{\overline{\mathcal{R}}}$, but then feasibility cannot be guaranteed in general.

In order to arrive at an implementable form, we follow the standard procedure of robust optimization. In particular, any feasible solution to the dual of (4)

$$\begin{array}{rll} \text{minimize} & \varepsilon \mu + \sum_{R \in \overline{\mathcal{R}}} \hat{p}_R \lambda_R & \\ \text{subject to} & \mu + \lambda_R \geq \sigma \hat{d}_R - \sum_{ijk \in K(R)} d_{ij} x_{ij}^k, & R \in \overline{\mathcal{R}}, \\ & \mu \in \mathbb{R}, \lambda_R \geq 0, & R \in \overline{\mathcal{R}}, \end{array}$$

provides an upper bound on $\delta(x, \sigma, \epsilon)$. So requiring that the dual has a feasible solution with nonpositive objective value for the current selection x is a sufficient condition that x satisfies the chance constraint (1). This yields the following program,

$$
\begin{array}{lll}
\text{minimize} & \sum_{e\in E} c_e(y_e) & \\
\text{subject to} & \sum_{1\le k\le k_{ij}} x_{ij}^k = 1, & ij \in W, \\
& \sum_{e\in E(C_{ij}^k)} d_{ij} x_{ij}^k \le y_e, & e \in E, \\
& \hat{p}^T \lambda + \mu \varepsilon_{\overline{\mathcal{R}}} \le 0, & \\
& \lambda_R + \mu \ge \sigma \hat{d}_R - \sum_{ijk\in K(R)} d_{ij} x_{ij}^k, & R \in \overline{\mathcal{R}}, \\
& \lambda \in \mathbb{R}_+^{\overline{\mathcal{R}}}, \mu \in \mathbb{R} & \\
& x \in \{0,1\}^K, y \in \mathbb{Z}_+^E. &
\end{array}
\tag{5}
$$

Depending on the choice of $\overline{\mathcal{R}}$ this may be a rather large integer program, but the number of binary variables has not increased in comparison to (2).

6 A Cutting Plane Approach to Solving the Chance Constrained Problem

A further possibility to handle the chance constraint (1) of problem (2) is to first solve the problem without the chance constraint and then to check the resulting assignment $x \in \{0,1\}^K$ for feasibility. If the assignment x does not satisfy the chance constraint, this assignment is excluded by adding appropriate inequalities as cutting planes and the problem is solved again together with the added constraints. The efficiency of this approach depends on the ability to describe the convex hull of the feasible solutions

$$P = \text{conv } \mathcal{X}(\sigma, \epsilon_{\overline{\mathcal{R}}})$$

and the efficiency of the mixed integer solvers that compute integer solutions for respective partial representations.

Given an assignment $x \in \{0,1\}^K$ it is easy to compute the set $\overline{\mathcal{R}}(x,\sigma)$ of disruption combinations whose loss in routable demand is too high, and by testing $\hat{p}(\overline{\mathcal{R}}(x,\sigma)) := \sum_{R\in\overline{\mathcal{R}}(x,\sigma)} \hat{p}_R \le \epsilon_{\overline{\mathcal{R}}}$ one quickly checks whether $x \in \mathcal{X}(\sigma, \epsilon_{\overline{\mathcal{R}}})$. So the main question is what to do if the probability of the event $\overline{\mathcal{R}}(x,\sigma)$ is too high, *i. e.* $\hat{p}(\overline{\mathcal{R}}(x,\sigma)) > \epsilon_{\overline{\mathcal{R}}}$, and this is what we study next.

For assignment x each disruption combination $R \in \overline{\mathcal{R}}$ has an associated set of demand pairs that cannot be routed within the selected routing subgraphs but would be routable within $G - F(R)$,

$U_R(x) := \{ij \in W(R)\colon x_{ij}^k = 1$ and i and j are not connected in $C_{ij}^k - F(R)\}$.

Thus, $R \in \overline{\mathcal{R}}(x, \sigma)$ is equivalent to the loss $\sum_{ij \in U_R(x)} d_{ij} \geq (1-\sigma)\hat{d}_R$ being too large for the given assignment x under the disruption combination R. A different assignment will be better for R only if this other assignment picks at least one routable subgraph from the set $\{ijk \in K(R)\colon ij \in U_{R(x)}\}$. A change in x is necessary only if the cumulative weight of all events in $\overline{\mathcal{R}}(x, \sigma)$ exceeds $\epsilon_{\overline{\mathcal{R}}}$. This gives rise to the following result.

Theorem 3 *Given events $\mathcal{S} \subseteq \overline{\mathcal{R}}$ with $\hat{p}(\mathcal{S}) > \epsilon_{\overline{\mathcal{R}}}$ and for each disruption combination $R \in \mathcal{S}$ a set of destination pairs $U_R \subseteq W(R)$ with $\hat{d}(U_R) := \sum_{ij \in U_R} d_{ij} \geq (1-\sigma)\hat{d}_R$. Denote by*

$$K(\mathcal{S}, (U_R)_{R \in \mathcal{S}}) := \bigcup_{R \in \mathcal{S}} \{ijk \in K(R)\colon ij \in U_R\},$$

then the inequality

$$\sum_{ijk \in K(\mathcal{S}, (U_R)_{R \in \mathcal{S}})} x_{ij}^k \geq 1 \tag{6}$$

is valid for all $x \in \mathcal{X}(\sigma, \epsilon_{\overline{\mathcal{R}}})$. Furthermore an assignment $x \in \{0,1\}^K$ is in $\mathcal{X}(\sigma, \epsilon_{\overline{\mathcal{R}}})$ if and only if it satisfies all such inequalities.

Proof. For proving that any feasible assignment x satisfies all these inequalities, let $x \in \mathcal{X}(\sigma, \epsilon_{\overline{\mathcal{R}}})$, *i. e.* $\sum_{k=1}^{k_{ij}} x_{ij}^k = 1$, $ij \in W$, $\sum_{R \in \overline{\mathcal{R}}(x,\sigma)} \hat{p}_R \leq \epsilon_{\overline{\mathcal{R}}}$ and assume, for contradiction, that inequality (6) does not hold for some $\mathcal{S}$ and $U_R \subseteq W(R)$, $R \in \mathcal{S}$, fulfilling the conditions above. Then

$$\sum_{ijk \in K(\mathcal{S}, (U_R)_{R \in \mathcal{S}})} x_{ij}^k = 0$$

because $x \in \{0,1\}^K$. Hence, $x_{ij}^k = 0$ for all ijk in $K(\mathcal{S}, (U_R)_{R \in \mathcal{S}})$, *i. e.* for all $ijk \in \bigcup_{R \in \mathcal{S}} \{ijk \in K(R)\colon ij \in U_R\}$. So by definition of $K(R)$ and U_R for each $R \in \mathcal{S}$ all the demand pairs $ij \in U_R$ cannot be routed within the selected subgraphs although they are routable within $G - F(R)$. Therefore, for each $R \in \mathcal{S}$ it holds that $U_R \subseteq U_R(x)$, hence

$$\sum_{ij \in U_R(x)} d_{ij} \geq \sum_{ij \in U_R} d_{ij} = \hat{d}(U_R) \geq (1-\sigma)\hat{d}_R.$$

As mentioned above, this is equivalent to $R \in \overline{\mathcal{R}}(x,\sigma)$, thus $\mathcal{S} \subseteq \overline{\mathcal{R}}(x,\sigma)$ and

$$\sum_{R \in \overline{\mathcal{R}}(x,\sigma)} \hat{p}_R \geq \sum_{R \in \mathcal{S}} \hat{p}_R = \hat{p}(\mathcal{S}) > \epsilon_{\overline{\mathcal{R}}}$$

contradicting the feasibility of x.

Now, let $x \in \{0,1\}^K$, $\sum\limits_{k=1}^{k_{ij}} x_{ij}^k = 1,\ ij \in W$ and assume that x is not a feasible assignment, *i. e.* $\sum_{R \in \overline{\mathcal{R}}(x,\sigma)} \hat{p}_R > \epsilon_{\overline{\mathcal{R}}}$. Then we can choose $\mathcal{S} = \overline{\mathcal{R}}(x,\sigma)$ and $U_R = U_R(x)$ for all $R \in \overline{\mathcal{R}}(x,\sigma)$ satisfying the conditions of the observation. Because $x_{ij}^k = 0$ for all $ijk \in K(R)$ with $ij \in U_R(x)$, we have

$$\sum_{ijk \in K(\overline{\mathcal{R}}(x,\sigma),(U_R(x))_{R \in \overline{\mathcal{R}}(x,\sigma)})} x_{ij}^k \quad \leq \quad \sum_{R \in \overline{\mathcal{R}}(x,\sigma)} \ \sum_{ijk \in K(R)\colon ij \in U_R(x)} x_{ij}^k = \sum_{R \in \overline{\mathcal{R}}(x,\sigma)} 0.$$

■

The second part of this proof already illustrates how to identify violated inequalities and how to reduce their support in order to improve their quality. This is summarized in the next observation.

Observation 4 *Given an assignment $x \in \{0,1\}^K$ with $\hat{p}(\overline{\mathcal{R}}(x,\sigma)) > \epsilon_{\overline{\mathcal{R}}}$, then for $R \in \overline{\mathcal{R}}(x,\sigma)$ the sets $U_R(x) = \{ij \in W(R)\colon x_{ij}^k = 1 \text{ and } ijk \notin K(R)\}$ satisfy $d(U_R(x)) \geq (1-\sigma)\hat{d}_R$. For each $\mathcal{S} \subseteq \overline{\mathcal{R}}(x,\sigma)$ with $\hat{p}(\mathcal{S}) > \epsilon_{\overline{\mathcal{R}}}$ and $U'_R \subseteq U_R(x)$ with $d(U'_R) \geq (1-\sigma)\hat{d}_R$ for $R \in \mathcal{S}$ the inequality*

$$\sum_{ijk \in K(\mathcal{S},(U'_R)_{R \in \mathcal{S}})} x_{ij}^k \geq 1 \tag{7}$$

is valid for $\mathcal{X}(\sigma, \varepsilon_{\overline{\mathcal{R}}})$ but violated by x.

There is an equivalent formulation of the inequalities above that might be better to use in practice because it might be sparser. Indeed, the requirement of the constraint is that for at least one ij with $ij \in U'_R$ for some $R \in \mathcal{S}$ the assignment ijk with $x_{ij}^k = 1$ must change to some $ijk' \in K(R)$. Alternatively, x may not use all those indices simultaneously, that do not satisfy one of these requirements. For this put $U(\mathcal{S}, (U'_R)_{R\in\mathcal{S}}) := \bigcup_{R\in\mathcal{S}} U'_R$, *i. e.* for each $R \in \mathcal{S}$ this set contains a subset of demand pairs that are not connected for this R and jointly exceed the demand loss bound. For each $ij \in U(\mathcal{S}, (U'_R)_{R\in\mathcal{S}})$ let $\bar{K}_{ij}(\mathcal{S}, (U'_R)_{R\in\mathcal{S}}) := \bigcap_{R\in\mathcal{S}\colon ij\in U'_R}\{ijk \in K \setminus K(R)\}$ collect all indices of ij-subgraphs that fail for all $R \in \mathcal{S}$ with $ij \in U'_R$ (it contains ijk with $x_{ij}^k = 1$). Because $\hat{p}(\mathcal{S}) > \epsilon_{\overline{\mathcal{R}}}$, for at least one of these demand pairs ij a subgraph with index outside $\bar{K}(\mathcal{S}, (U'_R)_{R\in\mathcal{S}}) := \bigcup_{ij\in U(\mathcal{S},(U'_R)_{R\in\mathcal{S}})} \bar{K}_{ij}(\mathcal{S}, (U'_R)_{R\in\mathcal{S}})$ has to be picked. This gives rise to the following reformulation of (7)

$$\sum_{ijk\in\bar{K}(\mathcal{S},(U'_R)_{R\in\mathcal{S}})} x_{ij}^k \leq |U(\mathcal{S}, (U'_R)_{R\in\mathcal{S}})| - 1. \tag{8}$$

The equivalence may be proved by direct computation using the assignment constraints $\sum_{k=1}^{k_{ij}} x_{ij}^k = 1$ for $ij \in U(\mathcal{S}, (U'_R)_{R\in\mathcal{S}})$. Likely this constraint has significantly fewer nonzero coefficients than (7).

7 Numerical results

In this section we present first comparisons of the performance of formulation (3), the robust problem (5) and a proof-of-concept implementation of the cutting plane approach. Before embarking on this we need to explain how we provide the necessary data comprising the networks, the ducts and the routing subgraphs C_{ij}^k. As usual, the latter will be cycles containing i and j so that in the case of a single simple disruption like a node or edge failure, *i. e.* $\underline{\mathcal{R}} = \{\{a\}\colon a \in \mathcal{G} \subseteq \overline{R}\}$, there is no loss of demand unless the disruption affects the origin and destination node directly. Therefore we only consider two-connected networks (any two nodes are connected by two node disjoint paths).

7.1 Providing data

As described in Example 1, we assume that the network is given by nodes, fibers/edges and ducts that contain these fibers. A disruption of a duct with corresponding failure probability entails a failure of all fibers that go through it. Because we had no access to network data that includes the specification of ducts, we generated it on the basis of instances from SNDlib [16] in the following way. For a graph $G = (V, E)$ with node set $V = \{1, \ldots, n\}$ and fiber/edge set E representing a given network, we first decide which subset of the given edges will also be considered as ducts. These ducts give rise a coarser spanning network $G_D = (V, E_D) \subseteq G$; once G_D is found, the original fibers are routed through these ducts.

In an attempt to respect the actual geometry of the duct network as well as two-connectedness we start constructing G_D by solving the traveling salesman problem for the complete graph on V with edge lengths corresponding to the Euclidean distances between the nodes. If the TSP solution uses connections that do not exist in G, we compute shortest paths between the affected nodes using only the given connections in E and collect all these edges in E_D, so G_D is not necessarily two-connected. If G_D contains a bridge $e_D \in E_D$ giving rise to two connected components V_1 and V_2 in $G_D - e_D$, every demand between V_1 and V_2 is routed over this duct. While we do not exclude this possibility in the end, it could lead to a network architecture where the failure of one or two ducts leads to a major loss of demand in the whole network. Therefore we add a small number of additional edges from $E \backslash E_D$ in order to provide a certain connectedness in G_D and to keep the distance between any two nodes in the graph metric (number of edges) adequately small if this is possible.

For this, we sort the connections in $E \backslash E_D$ by their fiber length starting with the smallest. Then we check for every connection $e = uv \in E \backslash E_D$ if there is a cycle that contains u and v whose number of edges is at most $n/10$ and only uses edges contained in E_D. If this is the case, we check the next edge in $E \backslash E_D$. If we cannot find a cycle at all, we simply add the edge e to E_D. In the case that we find a cycle whose number of edges is bigger than $n/10$ we take a look at the two u-v paths that form the cycle. If one of these two paths has less than $n/5$ edges we check the next edge, otherwise we add the edge e to E_D. This is motivated by the consideration that in practice one might not want to build a duct if it generates a short cycle, nor to build one if there is no significant improvement. Furthermore, this should help to provide reasonable dependencies between the failure scenarios of the edges to test our model.

Once G_D is constructed, all fibers/edges in E are embedded into the ducts by using shortest paths along the network G_D. The set of elementary disruptions is now formed by $\overline{R} = V \dot{\cup} E \dot{\cup} E_D$ and the failures by $F(v) = \{v, \{u, v\} \in E\}$ for $v \in V$, $F(e) = \{e\}$ for $e \in E$, and $F(d) = \{e \in$

E: e is embedded in $d\}$ for $d \in E_D$. The considered disruption combinations are all pairs of elementary disruptions, $\overline{\mathcal{R}} = \{\{r_1, r_2\} \subseteq \overline{R}\colon r_1 \neq r_2\}$, and the failure probabilities are computed as described in sections 2 and 3.

7.2 Providing the set of routing subgraphs

For every node pair we determine the set of routing subgraphs by solving a sequence of constrained Min-Cost Flow Problems in a directed graph. In order to ensure the generation of two node disjoint paths we employ the standard technique of doubling every node v of the network and classify it into a transmitter v_t and a receiver v_r and add a single arc going from v_r to v_t. Then for every edge $e = \{u, v\}$ of the undirected graph we add two arcs (u_t, v_r) and (v_t, u_r). Let A be the node-arc-incidence matrix of this directed graph, ξ a binary decision variable where each entry represents an arc used by the solution and γ a vector that contains the weights of the arcs. The first cycle is obtained by setting every entry of γ to one and solving

$$\begin{array}{ll} \text{minimize} & \gamma^T \xi \\ \text{subject to} & A\xi = b, \\ & \xi \in \{0, 1\}, \end{array}$$

where b are the balances indicating the source and destination node, *i. e.* if i is the source and j the destination, we put $b_i = -2$, $b_j = 2$ and $b_k = 0$ for $k \in V \backslash \{i, j\}$. By construction, this yields two edge and node disjoint i-j paths in the original undirected graph. In the next steps we exclude previous cycles by adding new constraints. Let $\bar{C}$ be the set of cycles (given by their edges) found so far, then for every $\bar{c} \in \bar{C}$ the constraint $\bar{c}^T \xi \leq \mathbf{1}^T \bar{c} - 1$ excludes this cycle. Furthermore, we update the cost vector γ in each step to increase the costs of edges that are already included in these cycles by setting $\gamma_e = 2^{N_e}$ where N_e counts the number of cycles the edge e appears in (for this specific node pair).

7.3 Separation heuristic

The cutting plane approach is based on formulation (8) and implemented in C++ using Gurobi [11] as IP-solver. The initial problem is set up without the chance constraint. During the branch-and-cut search a callback routine asks to check for every new solution x whether it satisfies the chance constraint. This is done by direct computation as described in Section 6. If it violates the chance constraint, x could be cut off by (8) for $\mathcal{S} = \overline{\mathcal{R}}(x, \sigma)$ and $U'_R = U_R(x)$, $R \in \mathcal{S}$, but in general this is rather inefficient. Instead we try to find, for each x, several stronger inequalities (8) without too much overlap in their supports by the following heuristic.

We use the reported solution x to construct a bipartite graph $\tilde{G}_x = (\tilde{V}_x, \tilde{E}_x)$ with $\tilde{V}_x$ of the form $\mathcal{R}_x \cup K_x$ where the set $\mathcal{R}_x = \overline{\mathcal{R}}(x, \sigma)$ contains all disruptions that lead to a loss of more than a $(1-\sigma)$-fraction of the routable demand for the current assignment and $K_x := \{ijk \colon x_{ij}^k = 1\}$ contains the indices of the chosen routing subgraphs. The edge set $\tilde{E}_x$ is generated by adding an edge $\{R, ijk\}$ if the disruption R leads to a complete breakdown of the cycle C_{ij}^k.

Let N_R denote the neighborhood of a disruption R in the graph $\tilde{G}_x$, for $\mathcal{J} \subseteq K_x$ define $\mathcal{R}_x(\mathcal{J}) := \{R \in \mathcal{R}_x \colon \sum_{ijk \in N_R \cap \mathcal{J}} \hat{d}_{ij} \geq (1-\sigma)\hat{d}_R\}$. Then the pair $(K_x, \mathcal{F})$ with $\mathcal{F} := \{\mathcal{J} \subseteq K_x \colon \sum_{R \in \mathcal{R}_x(\mathcal{J})} \hat{p}_R \leq \varepsilon_{\overline{\mathcal{R}}}\}$ is an independence system where the independent sets are subsets of K_x not yet violating the chance constraint. It is well known that for $\mathcal{J} \subseteq K_x$ the rank inequality $\sum_{ijk \in \mathcal{J}} x_{ijk} \leq r(\mathcal{J})$ where r denotes the rank function, is a valid inequality and if it is facet defining, then $\mathcal{J}$ is non-separable and closed ([13]). In general $(K_x, \mathcal{F})$ is not a matroid, so this is only a necessary condition. While we do not know how to efficiently test a given set for being non-separable or closed, we are able to find some circuits in dependent sets. This helps, because the rank inequalities of the circuits ensure feasibility for integer variables. In general the problem of enumerating all circuits of an independence system is NP-complete ([21]), so we resort to a heuristic to find small circuits.

We collect the cycles that will be used for the new constraint in a set $\mathcal{J}$ and the disruptions in a set $\mathcal{S}$, both are initialized to the empty set. Because disruptions that lead to a high loss in demand could have small probability or vice versa, we sort the disruptions of $\mathcal{R}_x$ in a descending order with respect to the product of the loss in demand and the probability of the disruption, *i. e.* with respect to the expected loss. Let R_1 be the first disruption in this ordering, N_{R_1} its neighborhood in $\tilde{G}_x$, set $\mathcal{S} \leftarrow \{R_1\}$, $\mathcal{J} \leftarrow N_{R_1}$. Assuming $\hat{p}(\mathcal{S}) \leq \epsilon_{\overline{\mathcal{R}}}$, the next step is to find disruptions in $\mathcal{R}_x$ such that the cardinality of the intersection of their neighborhood with $\mathcal{J}$ is maximal. Let R_2 be a disruption in $\mathcal{R}_x$ that satisfies this condition (if it is not unique, we pick one with maximum expected loss) and N_{R_2} its neighborhood

in $\tilde{G}_x$. Subsequently, we calculate the loss in demand for $\mathcal{J} \cap N_{R_2}$. If it already satisfies $\sum_{ijk \in \mathcal{J} \cap N_{R_2}} x_{ij}^k d_{ij} \geq (1-\sigma)\hat{d}_{R_2}$, *i. e.* $R_2 \in \mathcal{R}_x(\mathcal{J})$, we put $\mathcal{S} \leftarrow \mathcal{S} \cup \{R_2\}$ and go on with the next disruption in the mentioned order. If it is too small, we add the neighbors, sorted by their demand in descending order, in $N_{R_2} \backslash \mathcal{J}$ to $\mathcal{J}$ until the loss in demand is big enough. As mentioned above, we check $\hat{p}(\mathcal{S})$ in each step and repeat this procedure until the resulting probability is greater than $\epsilon_{\overline{\mathcal{R}}}$ such that as soon as this algorithm terminates it returns a subset $\mathcal{J}$ with the desired property.
Now we use the set $\mathcal{J}$ to find circuits of the mentioned independence system as follows. We initialize $\mathcal{C} \leftarrow \mathcal{J}$ and check for each element $ijk \in \mathcal{J}$ if $\mathcal{C} \backslash \{ijk\}$ is still a dependent set of $(K_x, \mathcal{F})$. If this is the case, we set $\mathcal{C} \leftarrow \mathcal{C} \backslash \{ijk\}$ and continue with the next element in $\mathcal{J}$. Eventually $\mathcal{C}$ is a circuit of $(K_x, \mathcal{F})$. If $\mathcal{C} = \mathcal{J}$ (i.e. $\mathcal{J}$ is a circuit), we only use $\mathcal{J}$ to add a new rank inequality $\sum_{ijk \in \mathcal{J}} x_{ij}^k \leq |\mathcal{J}| - 1$. In the other case we look for other circuits $\mathcal{C}$ by first trying to exclude elements that are already contained in some circuit that was found before until all elements in $\mathcal{J}$ are covered by the circuits. Furthermore, we stop this loop if we end up with a circuit that we found before. The rank inequalities of all these circuits are then added to the problem description.

7.4 Results

We give numerical results for several instances taken from SNDlib [16] to compare the explicit formulation (3) to the robust (5) and the cutting plane approach. The biggest test instance was *nobel-eu–D-B-E-N-C-A-N-S* with 28 nodes and 41 links, see Figure 2, the others are *norway–D-B-E-N-C-A-N-N* (27 nodes, 51 links), *janos-us–D-D-L-N-C-A-N-N* (26 nodes, 84 links), *newyork–D-B-E-N-C-A-N-N* (16 nodes, 49 links) and *atlanta–D-B-M-N-C-A-N-S* (15 nodes, 22 links). All other test instances of SNDlib in the range of 30 nodes were not two-connected, so we could not apply our approach there. The costs are calculated by multiplying the edge length and the respective computed capacity for every edge and summing all these values. By calling CPLEX 12.6 [10] using YALMIP [23] in MATLAB [14] we were able to generate integral solutions of (3) and (5) for $\sigma = 0.9$ (in our experiments we found only very few solutions for greater σ) and different choices of ϵ. All computation times refer to a QUAD-core processor INTEL-Core-I7-4770 (4x 3400MHz, 8 MB cache) machine with 32 GB RAM operating under openSUSE Linux 13.1.

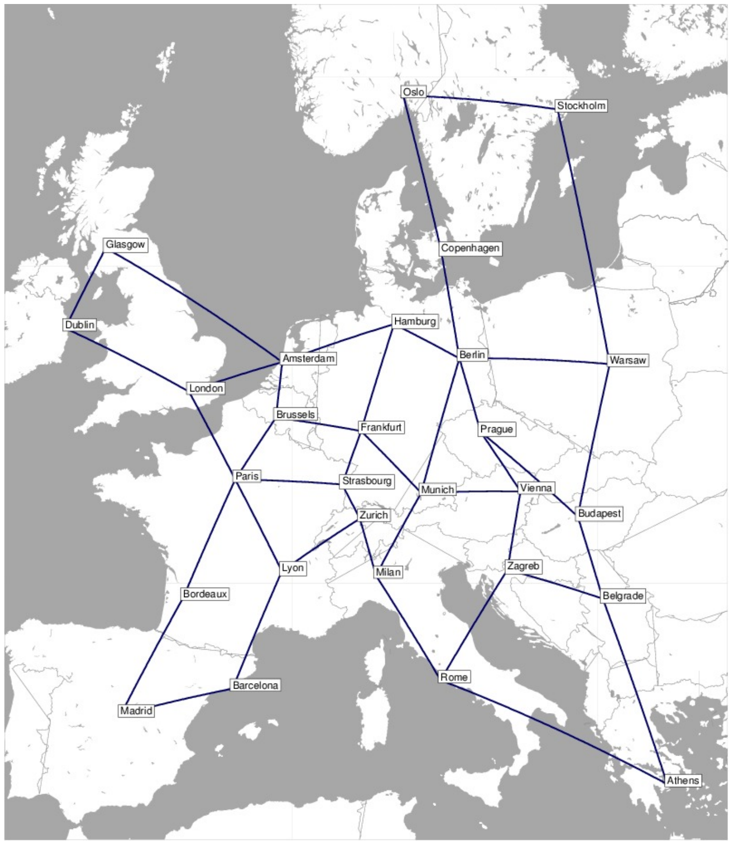


Figure 2: nobel-eu–D-B-E-N-C-A-N-S [16]

The columns of Table 1 give the name of the network, the value of ϵ, the optimal solution and computation time in seconds for the explicit formulation (3) and for the robust approach (5) and the gap (robust−IP)/IP in percent. CPLEX solves these instances of the IP-formulation in surprisingly reasonable time, but in most cases the robust approach yields high quality solutions in much shorter time. One problem that stands out is Norway. In the Norway network there are several significant nodes whose breakdown affects a large number of the cycles between the nodes. Furthermore, by our approach to select the ducts in this network, some of the ducts contain a lot of fibers. So there are many disruption combinations that lead to a loss of several cycles. In consequence, we only got a solution starting at $\epsilon = 0.3$ (this need not be the minimum feasible value for $\sigma = 0.9$). For this network the robust condition is too strong and (5) has no feasible solutions.

network	ϵ	explicit IP-formulation solution (time)	robust approach solution (time)	gap to minimum (in %)
nobel	0.01	70824.04 (4648.01s)	71581.06 (16.43s)	1.07
	0.05	70717.19 (4672.43s)	70717.19 (13.39s)	0
norway	0.3	1584567.76 (4548.24s)	– (15s)	–
	0.4	1569664.04 (4576.39s)	– (16s)	–
janos-us	0.01	4819928.28 (14033.9s)	4836246.72 (62.02s)	0.34
	0.05	4783310.85 (13258.14s)	4786321.66 (232.25s)	0.06
newyork	0.01	1750727.25 (1609.98s)	1781783.15 (8.64s)	1.77
	0.05	1690383.39 (2457.82s)	1752799.22 (8.98s)	3.69
atlanta	0.01	102169654.67 (424.58s)	102169654.67 (1.32s)	0
	0.05	102169654.67 (437.65s)	102169654.67 (2.3s)	0

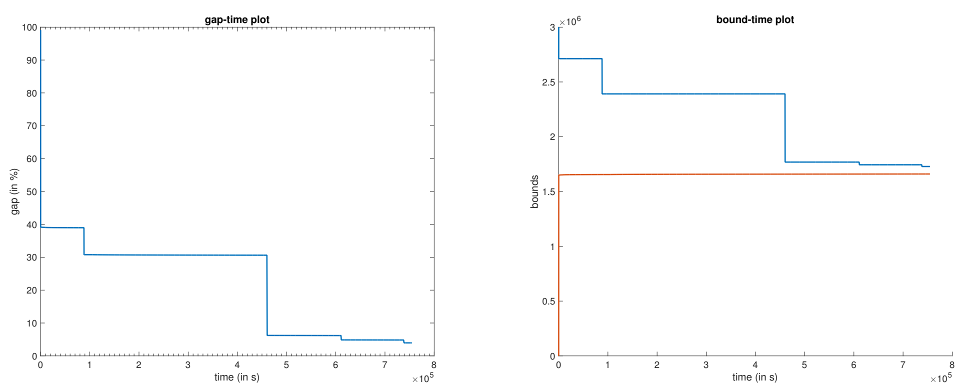
Table 1: Comparison of the cutting plane and the robust approach

network	ϵ	cutting plane approach lower bound	cutting plane approach best solution (time)	explicit IP-formulation solution (time)
nobel	0.01	70826.59	70835.49 (10563s)	70824.04 (4648.01s)
	0.05	70717.20	70717.20 (15.38s)	70717.19 (4672.43s)
norway	0.3	1569665.12	– (1120609s)	1584567.76 (4548.24s)
	0.4	1569665.12	1569665.12 (14s)	1569664.04 (4576.39s)
janos-us	0.01	4783309.73	– (1114676s)	4819928.28 (14033.9s)
	0.05	4783309.73	4783309.73 (32s)	4783310.85 (13258.14s)
newyork	0.01	1569665.12	1999198.56 (1109434s)	1750727.25 (1609.98s)
	0.05	1670334.77	1728348.59 (1098388s)	1690383.39 (2457.82s)
atlanta	0.01	102169663.61	102169663.61 (1s)	102169654.67 (424.58s)
	0.05	102169663.61	102169663.61 (1s)	102169654.67 (437.65s)

Table 2: Comparison of the cutting plane approach and the IP-formulation

For comparing the IP-formulation to the cutting plane approach Table 2 displays the lower bound and the best solution generated by the cutting plane approach (a proof-of-concept implementation in C++ using Gurobi [11] as IP-solver) together with their computation times and repeats the IP-solution (3). The slight deviations in the optimal values are due to the different environments C++ and MATLAB; indeed, evaluating the solutions in the same environment resulted in identical values. While the cutting approach produces good solutions quickly on some instances, it is also significantly slower on several others. These ambiguous results indicate that there is hope to outperform the explicit IP-formulation so as to allow for significantly large sizes of $\overline{\mathcal{R}}$, but a lot more work has to be invested into the polyhedral properties of the feasible set and the separation routines.

We complete this section with a plot of the development of the gap (in %, Figure 3(a)) and of the upper and lower bounds (Figure 3(b)) depending on the time (seconds) of the cutting plane approach for the New York network with $\epsilon = 0.05$ and $\sigma = 0.9$. This illustrates that the lower bound is of reasonable quality while finding good feasible solutions takes time.



(a) gap-time plot

(b) bound-time plot

Figure 3: time plots

8 Conclusion

Chance constraints offer a practically viable approach to multi-failure resilience. Realistic stochastic disruption models for single and multiple failures can be set up algorithmically. With respect to these predetermined probabilities the proposed chance constraint formulation selects among several given possibilities for each demand pair simultaneously a primary and a back up routing path so that probability is small to loose more than a given fraction of the routable demand. The model may be formulated explicitly as a mixed integer programming problem and is solvable exactly for backbone networks with up to 30 nodes within a few hours. For larger networks the robust variant (5) is likely the better choice because in most test cases it produced high quality solutions in much shorter time. The cutting plane approach needs more work to be competitive but it may offer a useful alternative when the number of considered disruption events is excessively large.

The approach may be extended in several directions. Some possibilities are including demand uncertainties, generating routing subgraphs dynamically, limiting network modifications, or using more elaborate failure models. Similar ideas may also be applicable in logistics or production. In mathematical terms it would be desirable to gain a better understanding of the convex hull of the feasible routing subgraph selections induced by the chance constraint. Indeed, there is a natural extension of Theorem 3 to general probability measures, so the cutting plane approach is not limited to discrete settings.

References

[1] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization.* Princeton University Press, 2009.

[2] A. Ben-Tal and A. Nemirovski, *Robust solutions of uncertain linear programs.* Operations Research Letters 25(1999)1-14.

[3] A. Ben-Tal and A. Nemirovski, *Robust solutions of linear programming problems contaminated with uncertain data.* Mathematical Programming 88(2000)411-424.

[4] A. Ben-Tal and A. Nemirovski, *Robust optimization - methodology and application.* Mathematical Programming 92(2002)453-480.

[5] D. Bertsimas and M. Sim, *Robust discrete optimization and network flows.* Mathematical Programming 98(2003)49-71.

[6] D. Bertsimas and M. Sim, *The Price of Robustness*. Operations Research 52(2004)35-53.

[7] A. Bley, F. D'Andreagiovanni, and D. Hanemann, *Robustness in Communication Networks: Scenarios and Mathematical Approaches*. Proc. of the ITG Symposium on Photonic Networks 2011, 1-8, VDE Verlag, Berlin (2011).

[8] A. Charnes, W.W. Cooper, and G.H. Symonds, *Cost Horizons and Certainty Equivalents: An Approach to Stochastic Programming of Heating Oil*. Management Science 4(1958)235-263.

[9] G. Claßen, D. Coudert, A.M.C.A Koster, and N. Nepomuceno. *A Chance-Constrained Model & Cutting Planes for Fixed Broadband Wireless Networks*. Network Optimization, 5th International Conference, INOC 2011, Hamburg, Germany, June 2011, Proceedings, 37-42, Springer (2011).

[10] CPLEX Optimizer, version 12.6.1. URL (June 25, 2015): `http://www-01.ibm.com/support/knowledgecenter/SSSA5P_12.6.1/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html`

[11] Gurobi Optimization, version 6.0.4. URL (June 25, 2015): `http://www.gurobi.com/documentation/6.0/refman/index.html`

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

[13] M. Laurent, *A generalization of antiwebs to independence systems and their canonical facets*. Mathematical Programming 45(1989)97-108.

[14] MATLAB, R2015a. URL (June 25, 2015): `http://www.mathworks.com`

[15] L.B. Miller and H. Wagner, *Chance-Constrained Programming with Joint Constraints*. Operations Research 13(1965)930-945.

[16] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, *SNDlib 1.0–Survivable Network Design Library*. Network Optimization, 3rd International Conference, INOC 2007, Spa, Belgium, April 2007, Proceedings. URL (June 25, 2015): `http://sndlib.zib.de`

[17] A. Prékopa, *On Probabilistic Constrained Programming.* Mathematical Programming Study 28(1970)113-138.

[18] A. Prékopa, *Contributions to the theory of Stochastic Programming.* Mathematical Programming 4(1973)202-221.

[19] A. Prékopa, *Stochastic Programming.* Kluwer, 1995.

[20] A.L. Soyster, *Convex programming with set-inclusive constraints and applications to inexact linear programming.* Operations Research 21(1973)1154-1157.

[21] F. Stork and M. Uetz, *On the generation of circuits and minimal forbidden sets.* Mathematical Programming 102(2005)185-203.

[22] S. Verbrugge, D. Colle, P. Demeester, R. Huelsermann, and M.Jaeger, *General Availability Model for Multilayer Transport Networks.* Proc. DRCN 2005, Lacco Ameno, Italy, Oct. 16-19, 2005

[23] YALMIP, Release 20150204. URL (June 25, 2015): http://users.isy.liu.se/johanl/yalmip