

 Open access • Journal Article • DOI:10.1002/NET.21807

## **Solving the bifurcated and nonbifurcated robust network loading problem with k-adaptive routing** — [Source link](#)

Marco Antonio Silva, Marco Antonio Silva, Michael Poss, Nelson Maculan

**Institutions:** Federal University of Rio de Janeiro, University of Avignon, University of Montpellier

**Published on:** 01 Jul 2018 - Networks (John Wiley & Sons, Ltd)

**Topics:** Routing (electronic design automation) and Iterative method

Related papers:

- [k-Adaptive Routing for the Robust Network Loading Problem](#)
- [Affine recourse for the robust network design problem: Between static and dynamic routing](#)
- [An optimal control approach to dynamic routing in networks](#)
- [A Topology Control Algorithm for Sensor Networks Based on Robust Optimization](#)
- [Application of aggregation strategies in the solution of the optimal routing problem in data networks](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/solving-the-bifurcated-and-nonbifurcated-robust-network-35kmcjv6>



**HAL**  
open science

# Solving the bifurcated and nonbifurcated robust network loading problem with k-adaptive routing

Marco Silva, Michael Poss, Nelson Maculan

► **To cite this version:**

Marco Silva, Michael Poss, Nelson Maculan. Solving the bifurcated and nonbifurcated robust network loading problem with k-adaptive routing. *Networks*, Wiley, 2018, Special Issue: Special Issue “New Advances and Applications in Deterministic and Stochastic Network Optimization”, 72 (1), pp.151-170. 10.1002/net.21807 . hal-01537872

**HAL Id: hal-01537872**

**<https://hal.archives-ouvertes.fr/hal-01537872>**

Submitted on 13 Jun 2017

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

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

# Solving the bifurcated and nonbifurcated robust network loading problem with $k$ -adaptive routing

Marco Silva<sup>a,c,1</sup>, Michael Poss<sup>b,2</sup>, Nelson Maculan<sup>c,3</sup>

<sup>a</sup> *LIA, Université d'Avignon et des Pays des Vaucluse, Avignon, France*

<sup>b</sup> *UMR CNRS 5506 LIRMM, Université de Montpellier, Montpellier, France*

<sup>c</sup> *PESC, UFRJ, Rio de Janeiro, Brazil*

---

## Abstract

We experiment an alternative routing scheme for the Robust Network Loading problem with demand uncertainty. Named  $k$ -adaptive, it is based on the fact that the decision-maker chooses  $k$  second-stage solutions and then commits to one of them only after realization of the uncertainty. This routing scheme, with its corresponding  $k$ -partition of the uncertainty set, is dynamically defined under an iterative method to sequentially improve the solution. The method has an inherent characteristic of multiplying the number of variables and constraints after each iteration, so that additional measures are introduced in the solution strategy in order to control time performance. We compare our  $k$ -adaptive results with the ones obtained through other routing schemes and also verify the effectiveness of the methods utilized using several realistic instances from SNDlib.

*Keywords:* Robust Linear Optimization,  $k$ -Adaptability, Network Loading Problem

---

<sup>1</sup> Email: marcosil@cos.ufrj.br

<sup>2</sup> Email: michael.poss@lirmm.fr

<sup>3</sup> Email: maculan@cos.ufrj.br

# 1 Introduction

This study is about a robust linear optimization approach for the network loading problem under demand uncertainty (*RNL*). Given a graph  $G(V, E)$  and a set of node-to-node uncertain demands as commodities origin-destination flows, we want to define minimum cost integer capacity installations for the edges (investment decisions), such that all commodities can be routed simultaneously on the network (routing decisions defining a routing scheme).

Robust network design problems, in general, have been widely studied motivated by the fact that demands in modern applications vary and are hard to forecast. The standard robust approach encompasses solutions that are feasible for any realization of uncertainty, so that a decision taken hedges against the worst contingency that may arise. This leads to a conservative solution approach.

A robust approach to the network design problem with demand uncertainty can be tracked back to the work of [19]. They introduce a polyhedral uncertainty set for demands and a min-max-min approach to the problem using dualizations to derive an equivalent linear problem formulation. In [22] the authors analyse for the first time many network design problems from a robustness perspective and introduce the term robust network design. They assert that multi commodity network design problems exhibit the important property of having multiple, and significantly different, solutions within a few percentage points from the optimal. This feature makes them amenable to algorithmic developments that can find common near optimal (robust) solutions for a variety of future operating scenarios. The uncertainty in the problems they consider are in the costs of the instances and are modeled as discrete scenarios.

Modeling uncertainty as discrete scenarios is limiting (see [15]) and since then many studies have been made on how to model uncertainty and the effect this has on the formulation, solution, and complexity of algorithms developed. Modeling uncertainty has a objective of reflecting real data behavior, but can affect conservatism of the solution.

Conservatism is also related to how the demands are routed through the network, meaning how commodities flows are carried from source to destination using network resources. The problem can be seen as a two stage robust optimization problem where the recourse function consists in choosing the routing for each demand vector. In fact, robust network design problems have a intrinsic separation between first stage decisions and second stage decisions: investment decisions must be made before we observe the results of

the demand uncertainty, while routing decisions have to route whatever demand occurred. One can decide which route to take based on actual demands, leading to less conservative solutions.

Routes can also have restrictions inherent to the application being modeled. Possible restrictions are if demands can be split between different paths (bifurcated) or not (non-bifurcated) or if there is a number maximum of edges to be used for each path, among others.

In this study we experiment and compare different modeling strategies to treat conservatism of robust solutions for *RNL*. Our focus is on *RNL* with polyedral uncertainty sets and routings based on the  $k$ -partitioning of the uncertainty, named  $k$ -adaptive.

The purpose of this paper, which is an extended version of the short paper [40] is four-fold:

- We present an extensive survey of the literature on robust network design, focusing on the uncertainty set and routing used.
- We show the application of iterative  $k$ -adaptive partitioning algorithm defined in [12] to *RNL*. As a side effect we apply decomposition techniques where we verify performance of newly CPLEX 12.7 Benders Decomposition functions.
- For nonbifurcated flows, we show the cost reductions provided by  $k$ -adaptive routing scheme over static. This is the first attempt to improve over static solutions as the integrality of the second stage variables prevents us from enumerating the extreme points of the uncertainty set (see [37]), using classical decomposition algorithms (see [27], [3]) or using affine decision rules (see [33], [34], [37]).
- For bifurcated flows, we compare the solution times and costs of  $k$ -adaptive routing scheme with those of static, volume and 2-partitioning routings.

**Outline** We first review in sections 2 and 3 the literature dealing with polyhedral and ellipsoidal uncertainty sets and routing schemes, applied to robust network design problems. Next, in Section 4, we formally present the problem that we will focus on our experiment. Section 5 details key concepts used in our routing scheme, that dynamically partitions the uncertainty set. In Section 6 we explain the solution strategy, with emphasis on techniques to reduce branching and decomposition techniques developed to cope with large instances. In Section 7 we present algorithm implementation details and results. Finally, concluding remarks follow in Section 8.

## 2 Uncertainty set

As stated in [9], the question of specifying an uncertainty set is mainly a modeling issue that should be resolved on the basis of application-driven considerations. Whatever the model utilized, special care should be taken to correctly express correlations between parameters that compose uncertainty and its consequence on problem's complexity. For the robust network design problem most of the literature reviewed models uncertainty by polyhedral or ellipsoidal sets.

The polyhedral uncertainty set was generalized in [7] for robust network design problems. They assume that demands satisfy linear inequalities, are bounded and argue that the polytope so defined,  $D$ , has to be sufficiently large to allow routes to be flexible, but not excessively large to avoid wasting network resources. No need exists to make any probabilistic assumptions about the demands. Each demand vector is defined in its components by its node origin and destination ( $d_{ij}$ ,  $i, j \in V$ ). They define a model in which  $A$  is a real-valued matrix and  $b$  is a real-valued vector and :

$$D = \{d \in \mathbb{R}_+^{|V| \times |V-1|} : Ad \leq b\}$$

Prior to that, another approach in the same direction can be found in the work of [18] for ATM's communications network and [17] for VPN's communications networks. They introduced a particular polyhedral model for uncertainty, based on the definition of demand upper bounds, named Hose model. The main driver was the ease and flexibility to model uncertainty with not much data information. In the Hose model two upper bound demand parameters are defined for each node:

$$\begin{aligned} d_i^{out} &: \text{ outgoing demand from node } i \in V \text{ to all other nodes} \\ d_i^{in} &: \text{ ingoing demand to node } i \in V \text{ from all other nodes} \end{aligned}$$

The polyhedral uncertainty polytope that results is defined by the following inequalities:

$$\begin{aligned} \sum_{j \in V \setminus \{i\}} d_{ij} &\leq d_i^{out} \\ \sum_{j \in V \setminus \{i\}} d_{ji} &\leq d_i^{in} \end{aligned}$$

The evolution of robust optimization as a discipline (see [9]) further leads

to studying different forms of uncertainty sets. The effect of uncertainty applied to data of previous deterministic problems on algorithm's complexity was studied.

Keeping our focus on polyhedral uncertainty sets, it was verified that robust linear problems defined under these uncertainty sets can be reformulated as linear problems, maintaining the complexity of the original deterministic problem. It was also exercised how the design of the uncertainty sets can influence the conservatism of the robust solution.

In that matter, a breakthrough that influenced future work on network design was in [13]. They recognize that a robust approach can be too conservative and a polyhedral uncertainty set is proposed in which one can flexibly adjust the level of conservatism of the robust solutions in terms of bounds of constraint violations. They make a link with the stochastic optimization approach to handle uncertainty, where uncertainty is defined by their probability properties. They protect against violation of constraint  $i$  of optimization problem deterministically, when only a pre-specified number  $\Gamma_i$  of the coefficients changes and where each coefficient is defined within a band of variation. In our demand uncertainty context this band defines different maximum value  $d_{ij}^{max}$  and minimum value  $d_{ij}^{min}$ . Named gamma-model or budgeted uncertainty, the polyhedral uncertainty polytope that results is defined by the following inequalities:

$$\begin{aligned} \sum_{i,j \in V} \frac{d_{ij} - d_{ij}^{min}}{d_{ij}^{max} - d_{ij}^{min}} &\leq \Gamma_i \\ d_{ij} &\geq d_{ij}^{min}, \quad i, j \in V \\ d_{ij} &\leq d_{ij}^{max}, \quad i, j \in V \end{aligned}$$

While the Hose model approach is useful when very limited information on demand is available, the result may be an overly conservative network design, as observed in [1]. Hence, when at least some degree of basic information is available on node-to-node demands and if it conforms as model to real data, the gamma-model may be an approach to avoid overly conservative network design.

Further studies on polyhedral uncertainty sets have identified other opportunities to reduce conservatism. In [14] they verify that the single deviation band for each coefficient proposed in [13] may be too limitative in practice, so that getting a higher resolution by partitioning the band into multiple sub-bands seems advisable. Here we consider demands as commodities ( $q \in Q$ ) origin-destination flows, expressed by their components

( $d^q$ ), where each commodity ( $q \in Q$ ) is defined through its nodes origin and destination. Following authors definition, each demand component is equal to the summation of a nominal value  $\bar{d}_q$  and a deviation lying in the range  $[d_q^{K^-}, d_q^{K^+}]$ , where  $d_q^{K^-}, d_q^{K^+} \in \mathbb{R}$  represent the maximum negative and positive deviations from  $\bar{d}_q$ . They model by partitioning the single deviation band  $[d_q^{K^-}, d_q^{K^+}]$  into sub bands defined on the basis of deviation values  $-\text{inf} < d_q^{K^-} < \dots < d_q^{-1} < d_q^0 < d_q^1 < \dots < d_q^{K^+} < +\text{inf}$ , so that we have:

- set of positive deviations bands:  $k \in \{1, \dots, K^+\}, (d_{ij}^{k-1}, d_{ij}^k]$
- set of negative deviations bands:  $k \in \{K^- + 1, \dots, 0\}, (d_{ij}^{k-1}, d_{ij}^k]$
- a deviation band:  $k = K^-$ , that corresponds to a single value  $d_{ij}^{K^-}$

This definition of sub bands further facilitates the reformulation proposed by the authors. They also define a maximum ( $u_k \in \mathbb{Z}$ ) and minimum value ( $l_k \in \mathbb{Z}$ ) of number of coefficients that can change per sub band:

$$\begin{aligned} 0 &\leq l_k \leq u_k \leq |Q| \\ u_0 &= |Q| \\ \sum_{k \in K} l_k &\leq |Q| \end{aligned}$$

They use dominance rules and dual reformulation to come up with an equivalent compact linear formulation. The performance of the new approach is tested on instances of a wireless robust network design problem.

In [35], the author introduces the concept of decision dependent uncertainty as an extension of the work presented in [13]. He identifies that the uncertainty set proposed in [13] suffers from a practical drawback since they are independent from the value of decision variables, so that some decision vectors, for instance in combinatorial problems with few non-zero components, are more protected than others. He proposes a new model where the uncertain parameters belong to the image of multi-functions of the problem decision variables. It can provide the same probabilistic guarantee as the uncertainty set defined in [13] while being less conservative. The feasibility set of the resulting optimization problem is in general non-convex so that a mixed-integer programming reformulation for the problem is proposed for uncertainty sets with affine dependency on decision variables. The study is developed for binary variables but is extended to non-binary variables and to more general multi functions involving uncertainty set defined by conic constraints that are



affine in the problem variables. For affine dependency on decision variables ( $x$ ) it defines a polyhedral uncertainty set, where  $B$  is a real valued decision coefficient matrix, that can be expressed as:

$$D(x) = \{d \in \mathbb{R}_+^{|V| \times |V-1|} : Ad \leq b + Bx\}$$

The model has been revived more recently in [31].

Ellipsoidal uncertainty sets were introduced in [10]. The motivation was to make a link with stochastic optimization, where uncertain data is associated with some probability measure. The correspondent robust optimization problem would pick as the uncertainty set some subset of the support of the probability measure. The authors present the ellipsoidal uncertainty set as a choice for many probability distributions. A ellipsoidal uncertainty set, in our demand uncertainty context, is given by:

$$D = \{d \in \mathbb{R}_+^{|V| \times |V-1|} : d^t Ad \leq b\}$$

In [4] and [5] the authors adopt ellipsoidal uncertainty sets to solve robust network design problems. Compared to stochastic optimization, it has better computational tractability. It also allows for the incorporation of statistical data based on past observation of traffic as well as probabilistic assumptions, especially correlation data. This is an advantage over polyhedral uncertainty sets. On the other hand, robust counterparts of ellipsoidal uncertainty sets linear problems turn into conic-quadratic optimization problems. The authors devise a modeling framework that uses a compact and tight approximation of ellipsoidal uncertainty sets by polyhedra. In this case, the robust counterpart remains linear. In [30] the authors also exercise ellipsoidal uncertainty sets to solve robust network design problems, under transportation cost and demand uncertainty. They solve the LP relaxation of the quadratic robust counterpart problem and provide computational experiments comparing results of different polyhedral and ellipsoidal uncertainty sets.

### 3 Routings

From now on we consider an undirected graph  $G(V, E)$ . Different routing schemes have been proposed in the literature with the objective of mitigating conservatism of standard robust approach where one solution only must satisfy all realizations of uncertainty.

Here we take the approach of [36] and define each routing scheme from a robust optimization perspective. Therein, a formal definition of routing is

made. A routing is a function,  $f : D \subset \mathbb{R}^{|Q|} \rightarrow \mathbb{R}^{|E| \times |Q|}$  that associates each demand vector, expressed by its components ( $d^q$ ) as commodities ( $q \in Q$ ) with nodes origin ( $s(q)$ ) and destination ( $t(q)$ ), with a multi-commodity flow, defined for each edge and commodity. This function satisfies the flow conservation constraint at each node of the network, where  $\delta(v) = \{ij \in E \mid i = v \text{ or } j = v\}$  and:

$$\sum_{j \in \delta(i)} f_{ij}^q(d) - f_{ji}^q(d) = \begin{cases} d^q & \text{if } i = s(q) \\ -d^q & \text{if } i = t(q), \text{ for each } i \in V \\ 0 & \text{else} \end{cases} \quad (1)$$

$$f_{ij}^q(d) \geq 0 \quad (2)$$

**Dynamic routing** One routing scheme is the set of all functions that satisfy (1) and (2) :

$$\mathbb{F} \equiv \{f : D \rightarrow \mathbb{R}^{|E| \times |Q|} \mid f \text{ satisfies (1) and (2)}\}$$

Named dynamic routing, it permits full flexibility in re-routing according to demand changes and, in consequence, is potentially less costly. It has been shown by [29] that the robust network design with dynamic routing is NP-hard for polyhedral uncertainty. The author of [27] develops a branch-and-cut algorithm to handle splittable dynamic routing *RNL* based on the use of metric inequalities. She performs a polyhedral analysis of the resulting capacity formulation and define a subproblem to separate metric inequalities that based on a bilevel formulation that she is able to reformulate as one level. She develops heuristic approaches as initial solutions and provides results in terms of lower and upper bounds of solutions obtained within a time limit.

**Static routing** An opposite approach to dynamic routing has been introduced by [7] for the robust network design problem. Designated static routing, meaning that for every commodity the same paths are used with the same splitting independently of the realization of demand. It can be formulated as a framework that considers a restriction on the second stage recourse. Each function component  $f^q : D$  to  $\mathbb{R}_+^{|E|}$  is forced to be a linear function of  $d^q$  :

$$f_{ij}^q(d) := y_{ij}^q d^q, ij \in E, q \in Q, d \in D \quad (3)$$

The flow  $y$  is called a routing template since it decides, for every commodity, which paths are used to route the demand and what is the percentage

splitting among these paths. Combining equations (1) and (3), the routing template  $y$  satisfies:

$$\sum_{j \in \delta(i)} y_{ij}^q - y_{ji}^q = \begin{cases} 1 & \text{if } i = s(q) \\ -1 & \text{if } i = t(q), \text{ for each } i \in V \\ 0 & \text{else} \end{cases} \quad (4)$$

We define formally the set of all routing templates:

$$\mathbb{Y} \equiv \{y \in \mathbb{R}_+^{|\mathcal{E}| \times |\mathcal{Q}|} \mid y \text{ satisfies (4)}\} \quad (5)$$

With that, we can define the set of all static routings:

$$\mathbb{F}^{stat} \equiv \{f : D \rightarrow \mathbb{R}^{|\mathcal{E}| \times |\mathcal{Q}|} \mid \exists y \in \mathbb{Y} : f \text{ satisfies (3)}\}$$

The resulting optimization problem is polynomially solvable. Many approaches have been presented to solve static routing network design problems, experimenting the effect of different formulations, uncertainty sets or solution methods through robust reformulation, decomposition and/or valid inequalities (as examples, see [1],[2], [23],[24], [21])

As presented, static and dynamic are two extremes routing schemes. In [20], they show that the solution for static routing *RNL* (splittable or unsplittable, under polyhedral uncertainty set) may be a factor of  $\Omega(\log |V|)$  more than the cost required when using dynamic routing. So, there is a need to define approximate solutions to the dynamic problem, that are tractable and not as conservative as the static solution. Recently many progress has been made in defining routing schemes in between static and dynamic, imposing a restriction on how the flow can be tuned to demand.

**Affine routing** Defined in [9] as the simplest restriction of this type (two stage), affine routing requires of the recourse variables to be affine functions of the data. The main motivation behind this restriction is that it results in a computationally tractable robust reformulation. The Affinely Adjustable Robust Counterpart problem (AARC), as named by the authors, is then shown to be, in certain important cases, equivalent to a tractable optimization problem, and in other cases, having a tight approximation which is tractable. The affine approximation does not support, however, second stage integer variables, or unsplittable flows for *RNL*. Affine routing was first addressed for the robust network design problem by [33]. They did that by restricting  $f^q$  to

be an affine function of all components of  $d$  giving

$$\mathbb{F}^{aff} \equiv \{f : D \rightarrow \mathbb{R}^{|E| \times |Q|} \mid \exists f_0 \in \mathbb{R}^{|Q|}, y \in \mathbb{R}^{|E| \times |Q| \times |Q|} : \\ f_{ij}^q(d) = f_{ij}^{0q} + \sum_{h \in Q} y_{ij}^{qh} d^h, \quad ij \in E, q \in Q, d \in D, \text{ and } f \text{ satisfies (1) and (2)}\}$$

In [37] the authors perform a theoretical and numerical study of the affine routing robust network design problem and describe particular conditions when affine, static or dynamic routing are equivalent. They also observe that the concept of dominance between demand vectors cannot be applied to affine routing, so that all demand vectors have to be considered, which in turn has implications to how one defines the uncertainty set. They experiment algorithms based on robust dual reformulation and conclude that the affine principle can be used to approximate free recourse using tractable robust counterparts. The formulations however tend to be very large such that they become hard to solve for large instances. In this context the authors suggest that it might be wise to restrict the number of commodities in the affine recourse or apply decomposition methods.

**Volume routing** Another concept for routing is derived from the idea of sharing two routes for each commodity in a way that it depends on actual volumes of demands. This type of routing, called volume routing, was originally introduced in [42]. Formally, the authors use the following set of routings, according to thresholds  $h^q$  for each  $q \in Q$ :

$$\mathbb{F}^V \equiv \{f : D \rightarrow \mathbb{R}^{|E| \times |Q|} \mid \exists y^1, y^2 \in \mathbb{Y}, h \in \mathbb{R}_+^{|Q|} : \\ f_{ij}^q(d) = y_{ij}^{1q} \min(d^q, h^q) + y_{ij}^{2q} \max(d^q - h^q, 0) \quad ij \in E, q \in Q, d \in D\}$$

They prove that the robust network design problem associated with  $\mathbb{F}^V$  is a NP-hard optimization problem. Hence, they introduce simpler frameworks described below. Defining  $d_{min}^q = \min_{d \in D} d^q$  and  $d_{max}^q = \max_{d \in D} d^q$ , the set of routings becomes one of the following:

$$\mathbb{F}^{VS} \equiv \{f : D \rightarrow \mathbb{R}^{|E| \times |Q|} \mid \exists y^1, y^2 \in \mathbb{Y} : \\ f_{ij}^q(d) = y_{ij}^{1q} d_{min}^q + y_{ij}^{2q} (d^q - d_{min}^q), \quad ij \in E, q \in Q, d \in D\} \\ \mathbb{F}^{VG} \equiv \{f : D \rightarrow \mathbb{R}^{|E| \times |Q|} \mid \exists y^1, y^2 \in \mathbb{Y} : \\ f_{ij}^q(d) = y_{ij}^{1q} d_{min}^q \frac{d_{max}^q - d^q}{d_{max}^q - d_{min}^q} + y_{ij}^{2q} d_{max}^q \frac{d^q - d_{min}^q}{d_{max}^q - d_{min}^q}, \\ ij \in E, q \in Q, d \in D\}$$

which are both well-defined whenever  $d_{min}^q < d_{max}^q$  for each  $q \in Q$ . When  $d_{min}^q = d_{max}^q$  for some  $q \in Q$ , the  $q$ -th component of  $f \in \mathbb{F}^{VG}$  is defined by  $f^q(d) = y^{1q}d^q$ . By definition  $\mathbb{F}^{VS}$  and  $\mathbb{F}^{VG}$  are special cases of affine routing.

A generic volume routing is defined in [28], where each commodity flow is based on two components: a flow that varies with demand and is a set of a arc-paths from origin to destination plus a fixed circulation (see [37]). It also restricts the number of commodities in the affine recourse and leads to smaller formulations.

$$\mathbb{F}^{VA} \equiv \{f : D \rightarrow \mathbb{R}^{|E| \times |Q|} \mid \exists f_0 \in \mathbb{R}^{|Q|}, y \in \mathbb{Y} : \\ f_{ij}^q(d) = f_{ij}^{0q} + y_{ij}^q d^q, \quad ij \in E, q \in Q, d \in D\}$$

In [36] the author compares different alternatives to  $opt(\mathbb{F})$ , meaning a comparison of conservatism between different routings solutions for the same problem. They show that  $opt(\mathbb{F}^{aff}) \leq opt(\mathbb{F}^{VG}) \leq opt(\mathbb{F}^{VS})$ , and  $opt(\mathbb{F}^V) \leq opt(\mathbb{F}^{VS})$ . They also show that it is not possible, in general, to order  $opt(\mathbb{F}^V)$  and  $opt(\mathbb{F}^{aff})$ . Also one can check that  $opt(\mathbb{F}^{aff}) \leq opt(\mathbb{F}^{VA}) \leq opt(\mathbb{F}^{VG})$  using the same arguments of [36].

**Multipolar routing** More recently, in [8], an alternative routing scheme is proposed with the objective of maintaining tractability of the reformulated optimization problem to be solved. It assumes a polyhedral uncertainty set and fixed recourse. Called multipolar routing, the authors approximate the uncertainty set by a polytope defined by its extreme routings, named poles. Each pole  $k \in \mathbb{K}$  is a special demand vector. The poles are given before hand and are defined in such a way as:

$$\exists \lambda \mid \sum_{k \in \mathbb{K}} \lambda_d^k k_{ij} = d_{ij}; \quad \sum_{k \in \mathbb{K}} \lambda_d^k k_{ij} \leq 1; \quad \lambda_d^k \geq 0; \quad \forall d \in D, \forall i, j \in V \quad (6)$$

Each pole  $k$  is defined as having a route associated with it, so that we have:

$$f_{ij}^q(k) := y_{ij}^{qk} k^q, \quad ij \in E, q \in Q, k \in \mathbb{K} \quad (7)$$

Moreover, the route for each realization of the demand vector will be a convex combination of the routes defined for each pole. Expression (6) states also

how each demand should be routed and we define:

$$f_{ij}^q(d) = \sum_{k \in \mathbb{K}} \lambda_d^k k_{ij} f_{ij}^q(k), \quad ij \in E, q \in Q, k \in \mathbb{K} \quad (8)$$

$$= \sum_{k \in \mathbb{K}} \lambda_d^k k_{ij} y_{ij}^{qk} k^q, \quad ij \in E, q \in Q, k \in \mathbb{K} \quad (9)$$

We define formally the the set of all multipolar routings as

$$\mathbb{F}^{MP} \equiv \{f : D \rightarrow \mathbb{R}^{|E| \times |Q|} \mid \exists \lambda \in \mathbb{R}^{\mathbb{K}}, y \in \mathbb{R}^{|\mathbb{K}| \times |E| \times |Q|} : f \text{ satisfies (1), (2), (9)}\} \quad (10)$$

The adaptability (and conservatism) of the solution will depend on how poles are defined. The authors show that multipolar routing encompasses static and dynamic routings.

**Routing by Partition of Uncertainty set.** One recurring idea for another approximate routing is the partitioning of the uncertainty set into subsets and considering a static routing for each subset. It has been studied under the general framework of multi-stage robust optimization in [11]. Therein, it is called finite adaptability, where the decision-maker chooses  $k$  second-stage static solutions, and then commits to one of them only after seeing the realization of the uncertainty. The decision-maker defines a cover of the uncertainty set with  $k$  subsets (possibly non-disjoint). It has the advantage of being able to naturally accommodate discrete second stage decision variables. We will call this routing scheme  $k$ -adaptive ( $\mathbb{F}^k$ ).

The inequalities  $opt(\mathbb{F}) \leq opt(\mathbb{F}^k) \leq opt(\mathbb{F}^{stat})$  hold in general. The authors provide an important geometric interpretation of the gap between static and dynamic routing solutions. They show that the gap is related to the fact that the static solution is not able to correlate uncertainty components between different constraints. For polyhedral uncertainty sets they show that this gap can be expressed in terms of matrices that belong to the uncertainty set. The authors also affirm that their  $k$ -adaptability proposal is not comparable to affine adaptability: in some cases affine adaptability fails where  $k$ -adaptability succeeds, and vice versa, so that we cannot order conservatism between the two approaches. The quality of solutions obtained with a  $k$ -adaptive approach depends on how the partition is built. Therefore there is a natural trade-off between the number of sets of a partition (and the solution time) and how “close” the solution is to the dynamic solution. Hence, a goal when using  $k$ -adaptability is to identify a partitioning scheme that is near the efficient frontier of this trade-off. In [11], the authors show that even

finding an optimal 2 partitioning is NP-hard and reformulate it as a bilinear optimization problem. Being NP-Hard, many heuristics have been proposed on how to split the uncertainty set.

In [6], the author introduces the idea of partitioning the uncertainty set with two (or more) subsets using hyperplanes and they propose to use a static routing template for each subset. This yields the following set of routings:

$$\mathbb{F}^2 \equiv \{f : D \rightarrow \mathbb{R}^{|E| \times |Q|} \mid \exists y^1, y^2 \in \mathbb{Y} \text{ and } \alpha \in \mathbb{R}^q, \beta \in \mathbb{R} : \\ f_{ij}^q = \begin{cases} y_{ij}^{1q} d^q & d \in D \cap \{d, \alpha d \leq \beta\} \\ y_{ij}^{2q} d^q & d \in D \cap \{d, \alpha d \geq \beta\} \end{cases}, \quad ij \in E, q \in Q, d \in D\}$$

The definition above implies that both routing templates  $y^1$  and  $y^2$  must be able to route demand vectors that lie in the hyperplane  $\alpha d = \beta$  without exceeding the capacity. They prove that the problem is NP hard in general and describe simplification schemes, where the direction  $\alpha$  is given and solution for each partition can be different or have to be identical.

The authors of [41] develop an algorithm based on a sequence of binary searches to solve a 2-partition robust linear problem with given hyperplane direction  $\alpha$  and identical solutions. They define a routine that can answer the question: Is there a feasible solution with smaller or equal cost than a given value? They perform a binary search using this routine and given upper and lower bounds on the cost function. The routine is also based on binary searches using given  $\beta_{max}$  and  $\beta_{min}$  as bounds to constantly shrink the interval of possible positions of the hyperplane.

Also using the strategy of partitioning of the uncertainty set, [39] introduces the idea of using conjointly two routing templates. Formally, the author proposes to use two routing templates  $y^1$  and  $y^2$  such that each  $d \in D$  can be served either by  $y^1$  or by  $y^2$  (or both). This yields the following set of routings:

$$\mathbb{F}^2 \equiv \{f : D \rightarrow \mathbb{R}^{|E| \times |Q|} \mid \exists y^1, y^2 \in \mathbb{Y} \text{ and } D^1, D^2 \subseteq D, D = D^1 \cup D^2 : \\ f_{ij}^q(d) = \begin{cases} y_{ij}^{1q} d^q & d \in D^1 \\ y_{ij}^{2q} d^q & d \in D^2 \end{cases} \quad ij \in E, q \in Q, d \in D\}$$

## 4 Problem definition

*RNL* are robust network design problems with integer capacities decision variables, reflecting practical cases where a number of different facilities type

can be installed at each edge. We focus on a simplification of *RNL* where each commodity  $q$  can be routed along a predetermined set of paths  $P(q)$ . In many practical applications there are limits on the number of different configurations of paths that can be implemented, so that we exercise the flexibility of routing schemes to mitigate the static solution conservatism even with this restriction. It has been used in several papers, see for instance [33] and [34]. These paths are represented by  $\delta_{qp}^{ij}$  that is equal to 1 if edge  $ij$  is contained in path  $p \in P(q)$ , for some  $q \in Q$  and is equal to 0 otherwise. They were predetermined as shortest paths weighted by edges costs for each commodity.

We work with two cases: one in which flows are unsplittable, or nonbifurcated, and must use a single path, and another one in which flows are splittable, or bifurcated, and can be fractionally split among several paths. The cost for routing flows is zero.

Each commodity  $q \in Q$  is associated with the uncertain demand  $d^q$ , within a given polyhedral uncertainty set  $D$  defined by  $m$  inequalities and given by its matrices  $A^{m \times |Q|}d \leq b^m$ .

The formulation contains integer investment decision variables  $x$ , where  $x_{ij}$  equals the planned installed capacity for edge  $ij$ , considering only one type of facility at a unit cost of  $c_{ij}$ . It contains continuous (for bifurcated flows) or binary (for nonbifurcated flows) routing decision variables  $y_{qp}$ , where  $y_{qp}$  equals the fraction of commodity  $q \in Q$  assigned to path  $p \in P(q)$ .

Given a partition  $D_1 \cup \dots \cup D_K$  of the uncertainty set  $D$ , the  $k$ -adaptive routing scheme restricts the routing functions to piece-wise constant functions defined as  $y_{qp}(d) = y_{qp}^k \Leftrightarrow d \in D_k$  where  $k \in \{1, \dots, K\}$ . The formulation follows, where  $Y \equiv \{0, 1\}$  for nonbifurcated flows and  $Y \equiv [0, 1] \subset \mathbb{R}$  for bifurcated flows and the partition can be optimized along with the other op-



timization variables:

$$\begin{aligned}
(kRNL) \quad & \min \quad \sum_{ij \in E} c_{ij} x_{ij} \\
& \text{s.t.} \quad \sum_{q \in Q} \sum_{p \in P(q)} d^q \delta_{qp}^{ij} y_{qp}^k \leq x_{ij} \quad \forall ij \in E, \forall d \in D_k, \\
& \quad \quad \quad k \in \{1, \dots, K\} \quad (11) \\
& \quad \quad \quad \sum_{p \in P(q)} y_{qp}^k = 1 \quad \forall q \in Q, k \in \{1, \dots, K\} \\
& \quad \quad \quad y_{qp}^k \in Y, x_{ij} \geq 0, x_{ij} \in \mathbb{Z} \quad \forall q \in Q, \forall p \in P(q), \\
& \quad \quad \quad \forall ij \in E, \\
& \quad \quad \quad k \in \{1, \dots, K\}
\end{aligned}$$

As mentioned before, choosing the optimal partition makes the problem extremely difficult so that we focus in the heuristic solution algorithm proposed by [12].

## 5 Iterative Nested Partitioning

In [12] and [38] the authors use the concept of dominance and independently identify that there is a set of active uncertain parameters  $\hat{d}$ . The active uncertain parameters are the values of the uncertain parameters that correspond to the constraints with minimum slack and that are the constraints that restricts the objective function. Applied to *kRNL*, the only constraints that involve uncertainty are constraints (11). We can define the set  $\hat{D}$ , of the active uncertain parameters of  $D$  (or equivalently for any subset  $D_k$ ) as follows. Given a solution  $(\tilde{x}_{ij}, \tilde{y}_{qp})$  we define:

$$\hat{D} = \{\hat{d} \mid \hat{d} = \arg \min_{d \in D} (\tilde{x}_{ij} - \sum_{q \in Q} \sum_{p \in P(q)} d^q \delta_{qp}^{ij} \tilde{y}_{qp}), \forall ij \in E\}, \quad (12)$$

where only one nominal element will be selected for each constraint involved in (12). If we were to remove all possible constraints that correspond to uncertain parameters that are not active at optimality, we are left with a reduced set of constraints that is sufficient to constrain the solution to its value. These remaining constraints, and the values of the active uncertain parameters that give rise to them, offer an important insight into how the uncertainty set structure relates to the optimal solution of the RO problem.

It shows that we must partition the uncertainty set in such a way as to guarantee that the uncertain parameters for the active constraints do not all lie in one set of the partition.

This insight can help, for instance, to improve results and performance of the 2 partitioning heuristics presented in Section 3. For the algorithms developed in [6] for routing scheme  $\mathbb{F}^{2|}$  it helps to reduce the limits of the binary search and can be used to make sure that the defined hyperplane does not include all active uncertain parameters in one partition.

The authors of [12] use the active uncertain parameters to construct a particular partition. Applied to  $kRNL$ , their approach imposes that each indexed element selected  $\hat{d}_k, k \in \{1, \dots, |E|\}$  of  $\hat{D}$  belongs to a single subset of the new partition, using Voronoi diagrams. In other words, given a set of  $|E|$  active uncertain parameters, the Voronoi diagram associated with these parameters defines a partition of  $D$  with one subset defined for each  $\hat{d}_k \in \hat{D}$  such that the Euclidean distance between  $\hat{d}_k$  and any  $d$  in this subset is less than or equal to the distance of  $d$  to any other given parameter  $\hat{d}_j, j \neq k$ :

$$\begin{aligned} D_k = D(\hat{d}_k) &= \{d \mid \|\hat{d}_k - d\|_2 \leq \|\hat{d}_j - d\|_2, \quad \forall \hat{d}_j, j \neq k\} \cap D \\ &= \{d \mid (\hat{d}_j - \hat{d}_k)d \leq \frac{1}{2}(\hat{d}_j - \hat{d}_k)(\hat{d}_j + \hat{d}_k), \quad \forall \hat{d}_j, j \neq k\} \cap D \end{aligned} \tag{13}$$

They further create the concept of nested partitions, where partitions can be created after previous partitions, so as to create a partition tree of active uncertain parameters. In other words one uses Voronoi Diagrams to partition subsets of the previous level of the tree. The nodes of the tree,  $T$ , corresponds to the set of all active uncertain parameters. Each level of the tree defines a sequence of subsets that define a partition of  $D$  at that level. To define these subsets under the concept of nested partitioning we need first to define some sets:

- $Leaves(T)$  is the set of leaves of the tree  $T$ . This is the set of all active uncertain parameters ( $\hat{d}_k$ ) at the last level of the tree. As mentioned above, each one is uniquely associated with a subset of a partition being considered for the  $kRNL$  problem. We index each set as  $k \in \{1, \dots, K\}$  as pointed out in our formulation of  $kRNL$ .
- $Children(\hat{d}_k)$  is the set of children of  $\hat{d}_k$  in the tree  $T$ . This is the set of the active uncertain parameters associated with the partition of one subset.
- $Parent(\hat{d}_k)$  is the parent of  $\hat{d}_k$  in the tree  $T$ . This is a one element set defined by the active uncertain parameter that originated one subset at

previous level of the tree.

- $Siblings(\hat{d}_k) = Children(Parent(\hat{d}_k))$

Hence a subset of the  $k$ -adaptive partition of the uncertainty set is obtained through a sequence of partitioning of many levels. Each subset is defined by the following inequalities:

$$\begin{aligned}
 D_k = D(\hat{d}_k) &= \{d \mid \|\hat{d}_k - d\|_2 \leq \|\hat{d}_j - d\|_2, \quad \forall \hat{d}_j \in Siblings(\hat{d}_k)\} \\
 &\cap \{d \mid \|Parent(\hat{d}_k) - d\|_2 \leq \|\hat{d}_j - d\|_2, \quad \forall \hat{d}_j \in Siblings(Parent(\hat{d}_k))\} \\
 &\dots \\
 &\cap D
 \end{aligned} \tag{14}$$

An iterative method is built around the partitioning scheme. It starts by solving a static-policy version of the adaptive optimization problem to determine a set of active uncertain parameters. They are used, through Voronoi Diagrams, to construct a finitely-adaptive version of the problem, and solve it. This in turn produces a new set of active uncertain parameters which we can then use to partition further, ideally improving on the previous solution at each iteration.

Partitions are created inside the partitions from the previous iteration, under the tree hierarchy concept presented above. This is important for our approach since optimal solutions for a partition can be used as initial feasible solutions (upper bounds) for the future nested partitions originated from it and because at each iteration will have a solution that is as good or better than the previous iteration (objective values using the nested approach do decrease monotonically through iterations).

The pseudo code in Algorithm 1 reflects these main steps and defines the algorithm implemented in our experiment for  $k$ -adaptive routing.

## 6 Algorithm improvements

We describe below improvements of Algorithm 1. The first two were already suggested in [12], while the decomposition algorithm and valid inequalities are specific to  $kRNL$  studied herein.

**Nested partial partitioning** In our problem, there are  $|E|$  capacity constraints leading to a total of as much of  $|E|$  active uncertain parameters and, so,  $|E|$  subsets associated to the repartition of each subset  $D_k$  of the previous partition. Using this approach without modifications, at each iteration  $n$  of

---

**Algorithm 1** Iterative  $k$ -Adaptive routing

---

Initialize ▷ No partition at first round  
**repeat**  
    Solve Equivalent Static *RNL* Problem  
    Calculate Bound  
    **if** Bound gap  $\geq$  Threshold **then**  
        Calculate Active Uncertain Parameters  
        Formulate Nested Partitioning  
    **end if**  
**until** Termination Criteria reached

---

our method, there will be as much as  $|E|^{(n-1)}$  subsets of a partition. This will possibly impact time performance due to the increase in the number of variables and constraints.

In order to mitigate the effect above, the authors in [12] suggest two practical approaches as trade-offs between computation time and quality of solution. One is to only take a subset of active samples from generated constraints, the ones with lowest slack, since constraints with greater slack are less likely to constrain the solution even with additional iterations. The other is based on the fact that the value of the objective is given by the worst result if we consider each element ( $D_k$ ) of a partition. Only the elements ( $D_k$ ) of the partition with this worst objective value restrict the overall objective and are called active subsets of a partition. If a subset is not active, further partitioning it is less likely to improve the objective. Applying these approaches will reduce the number of subsets of the partition at the next iteration, improving computational efficiency. The overall number of subsets defined at each iteration will then be algorithmic dependent.

To facilitate the formulations derived in further sections, we will represent inequalities of polyhedral uncertainty set  $D_k$ , defined in Section 5, by matrix notation  $B_k d \geq C_k$ , except for inequalities defining the original uncertainty set  $D$ .

For this, using equation (14) as reference, we define  $R$ , with  $r \in R$ , as an index set for  $H \equiv \{Siblings(\hat{d}_k) \cup Siblings(Parent(\hat{d}_k)) \cup \dots\}$  and function  $Siblings^{-1}(H_r), \forall H_r \in H$ , that returns the vector originating the sequence of siblings. For instance, if  $H_r \in Siblings(Parent(\hat{d}_k))$ , then  $Siblings^{-1}(H_r) = Parent(\hat{d}_k)$ .

So,  $B_k \in \mathbb{R}^{|R| \times |Q|}$  and vector  $C_k \in \mathbb{R}^{|R|}$  where, with further development of equation (14), we have:

$$\begin{aligned}
B_k(r, q) &= 2((Siblings^{-1}(H_r))^q - H_r^q) \\
C_k(r) &= \sum_{q \in Q} ((Siblings^{-1}(H_r))^q)^2 - (H_r^q)^2
\end{aligned}$$

**Bounds** Given a lower bound on the dynamic solution, we can use it to estimate the gap between our  $k$ -adaptive solution and the dynamic solution. The set of all active uncertain parameters generated through different iterations presented in Section 5,  $T$ , is used to obtain this lower bound. That is, after the solution of the partitioned problem at iteration  $n$ , we can augment the set and use them to construct and solve a deterministic problem whose objective provides the lower bound. A useful property of this bound is that as we obtain more samples the lower bound improves monotonically, so that at each iteration we shrink the bound gap from above and below. The lower bound formulation is given by:

$$\begin{aligned}
\min \quad & \sum_{ij \in E} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{q \in Q} \sum_{p \in P(q)} \hat{d}^q \delta_{qp}^{ij} y_{qp}^{\hat{d}} \leq x_{ij} \quad \forall ij \in E, \forall \hat{d} \in T \\
& \sum_{p \in P(q)} y_{qp}^{\hat{d}} = 1 \quad \forall q \in Q, \forall \hat{d} \in T \\
& y_{qp}^{\hat{d}} \geq 0, x_{ij} \geq 0 \quad \forall q \in Q, \forall p \in P(q), \forall ij \in E \\
& \quad \quad \quad \forall \hat{d} \in T
\end{aligned}$$

**Benders decomposition** As already identified, iterative partitioning suffers from an inherent characteristic of multiplying the number of variables and constraints after each iteration. This affects time performance of problem resolution, so that decomposition techniques are introduced in order to minimize the impact of this characteristic. For the unbifurcated case, we have implemented a Dantzig-Wolfe decomposition approach, but that leads to poor improvement in performance so that we do not present it here. For the bifurcated case, we take advantage of the natural staircase block structure of the  $kRNL$  formulation, where each block is related to a subset of the partition, to implement a Benders based decomposition. We use a branch-and-cut approach and break the problem into one that designs edge capacities (master problem defining variables  $x_{ij}, ij \in E$ ) and a sequence of other problems that checks feasibility of the designed edge capacities with respect to the uncertain demand requirements.

Specifically, given an integer solution  $\tilde{x}_{ij}$  for the master problem, typically obtained at an integer node of the branch-and-bound tree of this master problem, we solve a feasibility subproblem for each block related to a subset of the partition. Notice that each subproblem is the following robust linear program:

$$\begin{aligned}
\min \quad & \sum_{ij \in E} s_{ij} \\
\text{s.t.} \quad & \sum_{q \in Q} \sum_{p \in P(q)} d^q y_{qp}^k \delta_{ij}^p - s_{ij} \leq \tilde{x}_{ij} \quad \forall ij \in E, \forall d \in D_k \\
& \sum_{p \in P(q)} y_{qp}^k = 1 \quad \forall q \in Q \\
& s_{ij} \geq 0, y_{qp}^k \geq 0, x_{ij} \geq 0 \quad \forall q \in Q, \forall p \in P(q), \forall ij \in E,
\end{aligned}$$

where variables  $s_{ij}$  are additional slack variables. This way, it is sufficient to consider the demand uncertainty issues only when solving the much smaller subproblems. Each subproblem is defined by a subset  $D_k$  of  $D$ .  $D$  is defined by matrix notation  $A^{m \times |Q|} d \leq b^m$ . Each subset  $D_k$  is a polyhedral uncertainty set defined by an active uncertain parameter. Since each subproblem will be solved many times it will be wise to reformulate it using robust deterministic reformulation, so that we leverage this structure and not have to solve subproblems from scratch at each iteration. Using classical dualization techniques (e.g. [9]) leads to the following reformulation:

$$\begin{aligned}
\min \quad & \sum_{ij \in E} s_{ij} \\
\text{s.t.} \quad & \sum_{q \in Q} \sum_{p \in P(q)} (\bar{d}^q \delta_{qp}^{ij} y_{qp}^k) + \sum_{w \in \{1, \dots, m\}} b^w p_{ij}^w + \\
& \quad + \sum_{r \in R} C_k(r) u_{ij}^r \leq \tilde{x}_{ij} + s_{ij} \quad \forall ij \in E \quad (15) \\
& \sum_{p \in P(q)} y_{qp}^k = 1 \quad \forall q \in Q \quad (16) \\
& \sum_{w \in \{1, \dots, m\}} A^T(q, w) p_{ij}^w + \sum_{r \in R} B_k(r, q) u_{ij}^r \\
& \quad \geq \sum_{p \in P(q)} \hat{d}^q \delta_{qp}^{ij} y_{qp}^k \quad \forall ij \in E, \forall q \in Q \\
& p_{ij}^w \geq 0, u_{ij}^r \geq 0, y_{qp}^k \geq 0, s_{ij} \geq 0 \quad \forall q \in Q, \forall p \in P(q), \\
& \quad \forall ij \in E, r \in R, \\
& \quad w \in \{1, \dots, m\}
\end{aligned}$$

Let  $\pi_{ij} \leq 0, \forall ij \in E$  and  $\lambda_q, \forall q \in Q$  be the dual variables of constraints (15) and (16), respectively, and let  $\tilde{\pi}_{ij}$  and  $\tilde{\lambda}_q$  denote the optimal dual solution. By linear programming strong duality the solution for the subproblem above can be given by  $\sum_{ij \in E} \tilde{x}_{ij} \tilde{\pi}_{ij} + \sum_{q \in Q} \tilde{\lambda}_q$ . Moreover, this solution is equal to 0 if and only the feasible set is nonempty. Hence, if  $\sum_{ij \in E} \tilde{x}_{ij} \tilde{\pi}_{ij} + \sum_{q \in Q} \tilde{\lambda}_q \geq 0$  we add a strengthened Benders cut to the master problem, where  $m = \min_{ij \in E} \tilde{\pi}_{ij}$ :

$$\sum_{ij \in E} \lceil \frac{-\tilde{\pi}_{ij}}{m} \rceil x_{ij} \geq \lceil \frac{\sum_{q \in Q} \tilde{\lambda}_q}{m} \rceil,$$

If the master solution  $\tilde{x}$  is not violated by the strengthened Benders cut we add the non-rounded cut instead.

We take advantage of non zero values of the slack variables ( $s_{ij}$ ) and implement a heuristic solution for the master problem. For each subproblem, non zero value slack variables correspond to the capacity value missing to transform the master problem solution into a feasible one. For each solution  $\tilde{x}_{ij}$  we select the maximum non zero slack value  $s_{ij}$  provided by the different subproblems. These non zero values of slack variables complement master problem solutions and, rounded up, are used as upper bounds for the master problem.

**Valid Inequalities** The Benders decomposition approach normally has bad performance since it has a poor convergence. Other than that the master problem is an integer problem and may be computationally expensive. It is critical to reduce the number of branchings. This can be done, additionally, by the introduction of valid inequalities to the master problem. Many classes of valid inequalities have been derived by exploiting the integer property of design variables (see [21], [26], [28]). The cut-set inequalities are often used to tighten the integrality gap in the linear programming based branch-and-cut algorithms. Despite their simplicity, it has turned out that the cut-set inequalities are quite strong in terms of reducing the linear programming relaxation gap.

Consider a partition of the node set  $V$  given by sets  $S_1$  and  $\bar{S}_1$ , and let  $E(S_1, \bar{S}_1)$  and  $Q(S_1, \bar{S}_1)$  be the set of edges and commodities with extremities in different sets of the partition. The cut-set inequality associated with the partition states that the amount of capacity installed on edges in  $E(S_1, \bar{S}_1)$  should be not less than the rounded up sum of the demands of commodities in  $Q(S_1, \bar{S}_1)$ . We have our cut-set inequality given by:

$$\sum_{ij \in E(S_1, \bar{S}_1)} x_{ij} \geq \lceil \max_{d \in D} \sum_{q \in Q(S_1, \bar{S}_1)} d^q \rceil$$

Since  $x$  is a first stage decision variable cut-set inequalities have to hold for the original uncertainty set.

We separate robust cut-set inequalities using two approaches: the master problem starts with cuts generated for each node of the network and at each integer solution, during branching, we separate cuts heuristically by generating a two subset random partition of the nodes and then performing a local search picking up one node and moving it to the other subset until there is no more improvement in the violation. If no violated inequality is found we repeat the procedure up to a maximum of 20 iterations.

## 7 Implementation and Results

**Instances** Six network instances available from SNDlib [32] were utilized with the characteristics given in Table 1. We have predetermined a maximum of 5 paths for each commodity (if they exist). Each commodity  $q \in Q$  is associated with the uncertain demand  $d^q$ , within a given uncertainty set modeled as in [13]. The demand for commodity  $q \in Q$  varies around a given nominal demand  $\bar{d}^q \geq 0$  with a maximal possible deviation of  $\hat{d}^q \geq 0$ , that is,  $d^q(\xi) = \bar{d}^q + \xi^q \hat{d}^q$  for  $\xi \in \Xi$  where  $\Xi = \{\xi \mid \sum_{q \in Q} \xi^q \leq \Gamma, 0 \leq \xi^q \leq 1\}$ . Two  $\Gamma$ 's for the uncertainty



set are defined for each instance, based on the probabilistic bound of 25% and 1% (see [13],[28]). For the abilene instance a test case with all possible  $\Gamma$ s was also utilized.

**Routing schemes** We experiment many of the different routing schemes presented in this study. We test routing schemes with bifurcated (BF) and nonbifurcated (NBF) flows according to configuration Table 2. Static routing scheme solution is provided as a result of first iteration of our nested iterative  $k$ -adaptive static algorithm here presented. The volume routing scheme was implemented according to routing  $\mathbb{F}^{VS}$ . The 2 partitioning was implemented according to routing  $\mathbb{F}^2$ . Full partitioning (FP) means  $k$ -adaptive routing scheme going through our nested iterative partitioning where at the end of each iteration we add active uncertain parameters for all capacity constraints (11) of our formulation. Partial partitioning (PP) means nested iterative  $k$ -adaptive routing scheme where (i) we only add active uncertain parameters referent to the set of the partition that is restricting the objective value and (ii) we restrict to active uncertain parameters referent to the first half of constraints of this set with minimum slack. We do this as a trade off between time performance and quality of the solution. FP and PP were implemented using equivalent static solutions at each iteration, as presented in this study, but also incorporating an extension where we adopt volume routing solutions for each subset of the partition.

**Algorithms Specification** Algorithms were coded in Julia [25] using JuMP, JuMPeR and BlockDecomposition packages and Cplex 12.7. All algorithms were run in an Intel CORE i7 CPU 3770 machine. A limit of 7200 seconds of computing time was given for each instance and iteration.

Algorithms were defined according to second column of Table 3. Algorithms were adapted to each routing scheme, and can run with or without adding cut-set inequalities, or also adding cut-set inequalities only at root node of our branch-and-cut procedure. Cut-set inequalities were implemented using JuMP lazy callback functions. Root node only cut-set inequalities were introduced to CPLEX BENDERS Algorithm because Cplex 12.7 does not support callback functions for its Benders Decomposition.

Compact formulation means using classical robust dual reformulation to solve the original formulation of the problem (for Static and Volume routing schemes), or the  $k$ -adaptive formulation (for Partial Partitioning or Full Partitioning routing schemes). This is done using JuMP.

For CPLEX BENDERS, we have utilized the new automatic Benders de-

composition functionality of Cplex 12.7 to solve a compact formulation, using BlockDecomposition and JuMP.

Adapted Benders Decomposition implements the Benders method presented in this study. Benders primal subproblems were solved through robust dual reformulation using JuMP lazy callback functions and with dual simplex method to leverage the fact that only the right hand side ( $\tilde{x}_{ij}$ ) of constraints (15) change between two calls of the subproblems.

The 2-partition was implemented replicating the algorithm presented in [41], extended to integer capacities. Here problems were solved using a cutting plane approach provided by JuMPeR.

As mentioned above, both Partial Partitioning and Full Partitioning routing schemes were implemented using the overall nested iterative  $k$ -adaptive algorithm presented in this study. At each iteration we solve Static or Volume routing  $k$ -adaptive problems using algorithms indicated in Table 3. We have applied decomposition algorithms only for Static Partial Partitioning routing as an experiment to verify their time performance compared to Compact reformulations.

**Results** Figures 1 and 2 present solution improvement over static routing provided by different bifurcated and nonbifurcated routing schemes with 2 different  $\Gamma$ s. For Figure 1 there are a number of cases that do not achieve an optimal solution under the time limit and they appear as blank bars. For Figure 2 instances missing correspond to cases where there was no improvement in solution or to cases that did not achieve an optimal solution under the time limit. Solution improvement is given as percentage by the formula  $100\% * \frac{(Static\ Solution - Solution)}{Static\ Solution}$ . The bars referenced as Bound correspond to the solution of the Bound procedure at the last iteration of our nested iterative  $k$ -adaptive algorithm and provides a reference of how much space for reducing conservatism is still available.

Figure 3 presents solution improvement in percentage over static routing provided by different bifurcated routing schemes, for instance abilene and  $0 \leq \Gamma \leq |Q|$ . Routings  $\mathbb{F}^k VIPP$  and  $\mathbb{F}^k VIFP$  are not presented since they did not improve over  $\mathbb{F}^{VS}$  in this case.

Table 7 summarizes analysis we make of Figures 1 and 2 regarding the performance of each routing scheme in improving static result. Table 7 and further analysis of conservatism from Figures 1 and 2 and Figure 3 indicates:

- For our SNDlib instances, there is a short gap between static and dynamic solutions. Nevertheless, the routing schemes implemented were able to re-

duce this gap in many cases.

- The restriction on number of paths imposes less flexibility in reducing conservatism. This is even worse for non bifurcated flows where more restrictions are imposed on the permitted flows. This impacts the solutions associated with non bifurcated routings where improvement was even more reduced.
- Cost reduction is higher for lower  $\Gamma$ s. From theory we know already that increasing  $\Gamma$  leads to more conservatism, but we see also that all routing schemes have more difficulty to provide better solutions for higher  $\Gamma$ s.
- The analysis of routes based on partitioning shows that  $\mathbb{F}^{2l}$  provided less solution improvement. In part this was due to time limit and because the partition hyperplane direction was arbitrarily chosen. In that sense,  $\mathbb{F}^k PP$  and  $\mathbb{F}^k FP$  had more success, for Static and especially for Volume solutions. In particular,  $\mathbb{F}^k PP$  for Static and Volume solutions, was able to give results comparable to  $\mathbb{F}^k FP$  and in some instances even better.
- Volume solutions, both for  $\mathbb{F}^{VS}$  and  $\mathbb{F}^k$ , were able to give best improvements, but it cannot handle non bifurcated flows.  $\mathbb{F}^k VI$  was able to further improve  $\mathbb{F}^{VS}$ .
- The fact that we have limit the number of iterations of  $\mathbb{F}^k PP$  and  $\mathbb{F}^k FP$  routing schemes has an effect on solution improvement. More iterations would, in general, improve solution at a cost of time performance. In general, for both BF and NBF solutions,  $k$ -adaptive routing scheme was able to improve over static.
- For non bifurcated flows many instances were not solved within time limit. Moreover, solution improvement was not as large as in the bifurcated case. This suggests we need to consider more path flow options for the non bifurcated case, and also develop more efficient decomposition algorithms.

Table 5 presents total time for different algorithms running for bifurcated routing schemes. We have limited our algorithms to 2 iterations for full partitioning and 3 iterations for partial partitioning. There is a number of instances that do not achieve an optimal solution under the time limit, marked as  $M$ . Table 6 presents a comparison between the two different decomposition algorithms we have used, CPLEX BENDERS and Adapted Benders Decomposition. They were run for static bifurcated partial partition routings. Here we present total time (TT) and also number of Benders cuts (BC) and cut-set cuts (CSC) introduced by each algorithm. We duplicate total time (TT) columns for compact formulation with cut-set and without cut-set inequalities added

to facilitate comparison between decomposition and non decomposition algorithms. Figure 4 compares algorithms for static bifurcated partial partition routing through a performance profile ([16]). Table 8 summarizes the analysis we make of Tables 5 and 6 regarding time performance of each algorithm and routing scheme. We remark that cut-set inequalities are randomly generated and that can influence results of comparison. We are interesting in verifying effectiveness of introducing cut-set inequalities to our formulations, the use of decomposition algorithms, the new Benders functionality of CPLEX 12.7, the use of partial partitioning and volume versus static partitioning.

Table 8 and further analysis of Tables 5, 6 and Figure 4 indicates:

- Robust cut-sets inequalities were key to reduce branching and provide better time performance.
- In general, Decomposition provided better time performance when compared to non Decomposition algorithms.
- Figure 4 shows that robust cut-sets inequalities were also key to improve performance of Decomposition algorithms. It shows also that our Adapted Benders Decomposition implementation was in general better in performance than CPLEX 12.7 Benders Decomposition, even though we did not implement features as parallel processing of subproblems, as an example. This is due, in part, to the fact that we have previous knowledge of the problem. For instance, we make use of intermediary heuristics (upper bounds) for the master problem. Moreover, CPLEX 12.7 Benders Decomposition does not allow lazy constraints and we were not able to introduce robust cut-sets inequalities during branching.
- For Adapted Benders Decomposition the introduction of robust cut-sets inequalities resulted in a significant reduction in the number of Benders cut, that are in general more costly to separate (reduction in 7 of 12 instances).
- Although  $\mathbb{F}^k PP$  runs more iterations than  $\mathbb{F}^k FP$ , it runs in less time. In that sense, and the fact that it delivers comparable solutions,  $\mathbb{F}^k PP$  presented itself as a good option to  $\mathbb{F}^k FP$ .
- The analyses of routes based on partitioning shows that  $\mathbb{F}^{21}$  provided worst time performance, due to the associated non polynomial binary search algorithm. Based on that and on solution improvement,  $\mathbb{F}^k PP$  and  $\mathbb{F}^k FP$  turned out to be preferred partitioning based routing schemes.
- As identified above, many non bifurcated flows instances were not resolved within time limit, suggesting the need to develop more efficient decomposition algorithms.

Instance	V	E	Q	$\Gamma^{0.25}$	$\Gamma^{0.01}$
abilene	12	15	65	6	19
polska	12	18	66	6	19
pdh	11	34	24	4	12
di-yuan	11	42	22	4	11
nobel-us	14	21	91	7	23
atlanta	15	22	105	7	24

Table 1  
Instances Profile

Code	Routing Scheme
$\mathbb{F}^{stat}BF$	Static BF
$\mathbb{F}^{stat}NBF$	Static NBF
$\mathbb{F}^{VS}BF$	Volume BF
$\mathbb{F}^{2l}BF$	2 Partitioning BF
$\mathbb{F}^kStPPBF$	Static Partial Partitioning BF
$\mathbb{F}^kStPPNBF$	Static Partial Partitioning NBF
$\mathbb{F}^kVlPPBF$	Volume Partial Partitioning BF
$\mathbb{F}^kStFPBF$	Static Full Partitioning BF
$\mathbb{F}^kStFPNBF$	Static Full Partitioning NBF
$\mathbb{F}^kVlFPBF$	Volume Full Partitioning BF

Table 2  
Routing Schemes

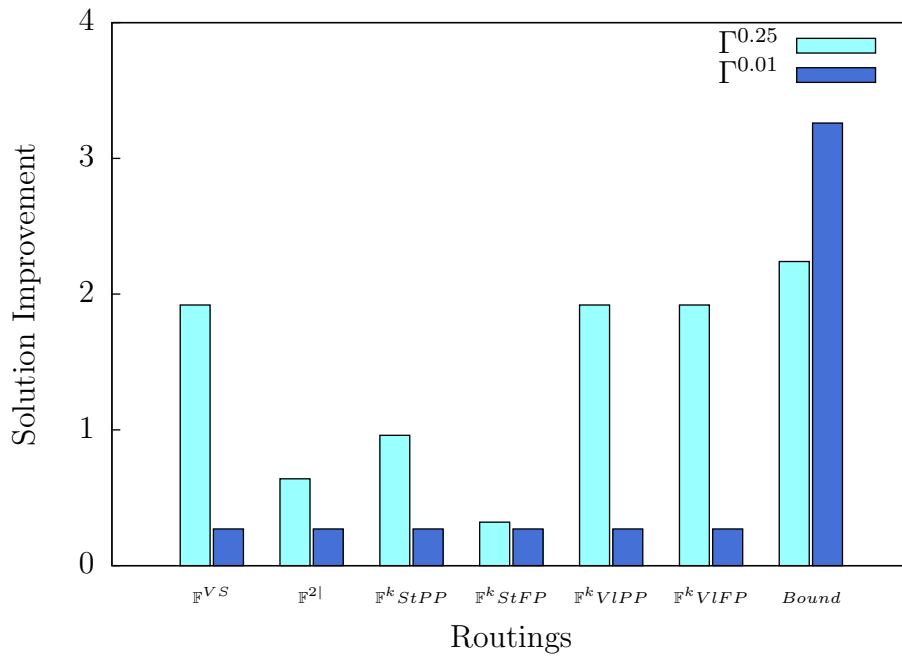
- In general, for our instances,  $\mathbb{F}^kVl$  has worse time performance than  $\mathbb{F}^kSt$ , but provides better bifurcated solutions as we have seen.

Code	Algorithm	$\mathbb{F}^{stat}BF$ $\mathbb{F}^k StPPBF$	$\mathbb{F}^{VS}BF$ $\mathbb{F}^k VIPPBF$ $\mathbb{F}^k VIFPBF$ $\mathbb{F}^k StFPBF$	$\mathbb{F}^{2 }BF$
C	Compact reformulation	✓	✓	-
C+C	Compact reformulation + cut-set	✓	✓	-
DCP	CPLEX BENDERS	✓	-	-
DCP+RC	CPLEX BENDERS + root cut-set	✓	-	-
DBD	Adapted Benders Decomposition	✓	-	-
DBD+C	Adapted Benders Decomposition + cut-set	✓	-	-
TP	Two partition	-	-	✓

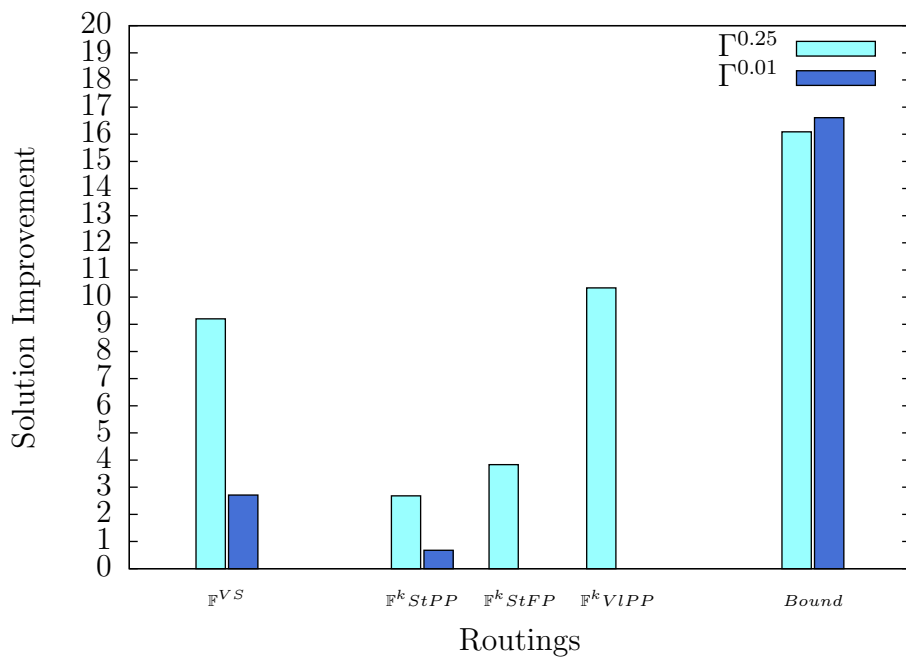
Table 3  
BF Algorithms implemented

Code	Algorithm	$\mathbb{F}^{stat}NBF$	$\mathbb{F}^k PPBF$	$\mathbb{F}^k StFPBF$
C	Compact reformulation	✓	✓	✓
C+C	Compact reformulation + cut-set	✓	✓	✓

Table 4  
NBF Algorithms implemented

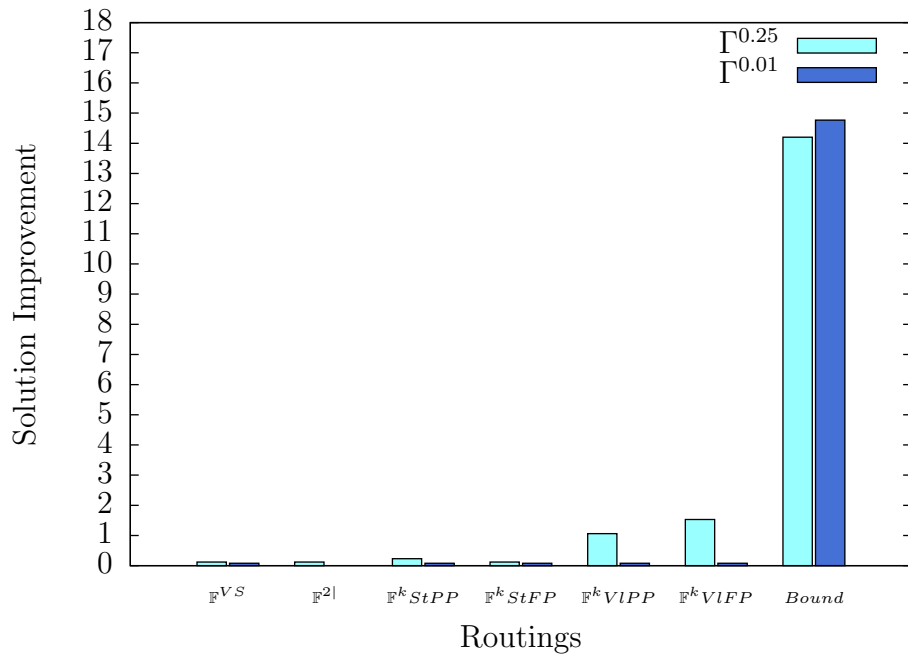


(a) abilene

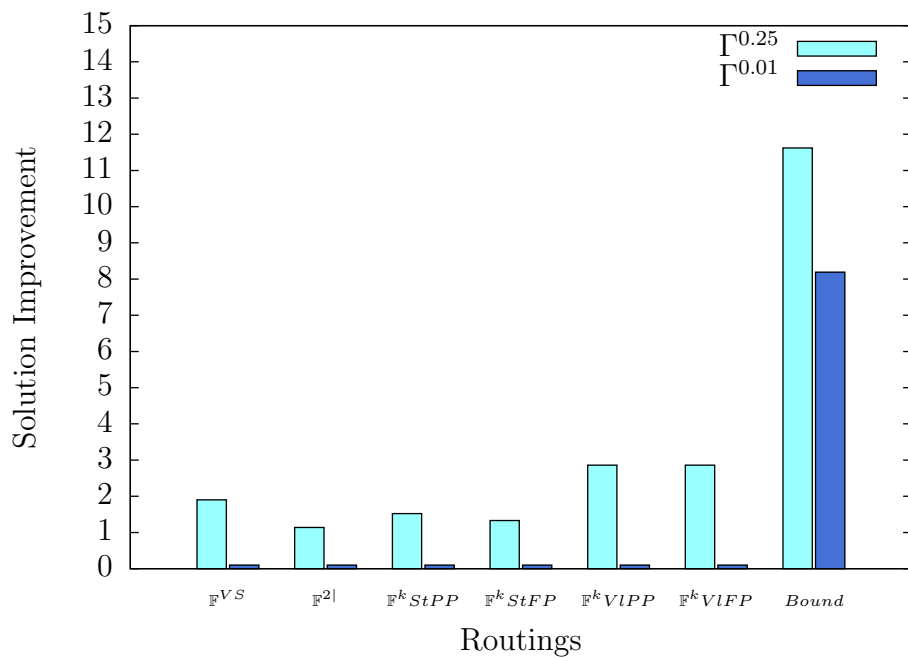


(b) polska

Fig. 1. Part 1: Percentage of Solution Improvement over Static BF Routing



(c) pdh



(d) di-yuan

Fig. 1. Part 2: Percentage of Solution Improvement over Static BF Routing



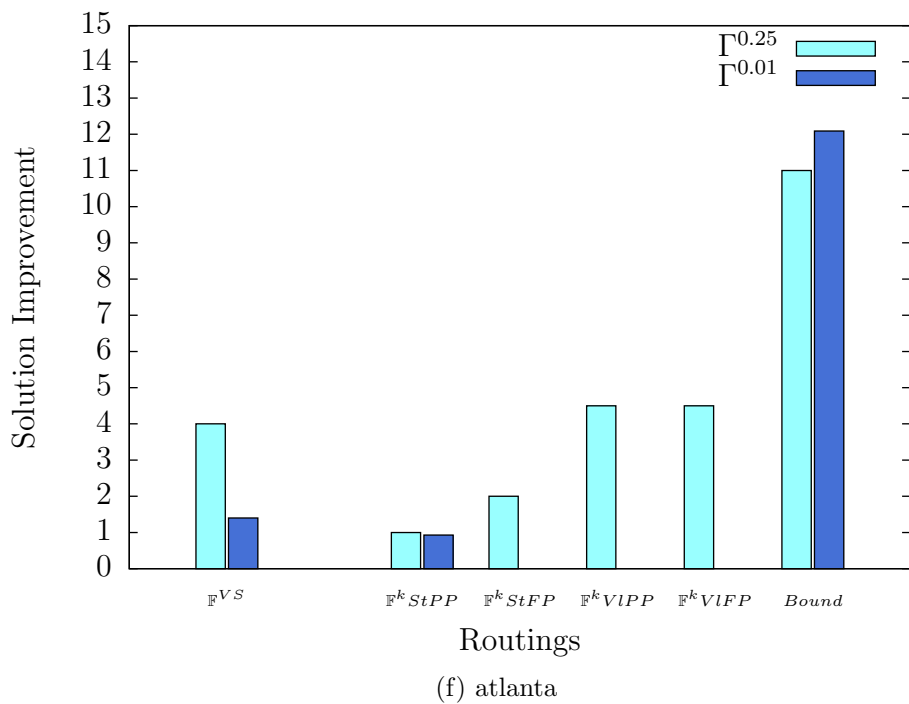
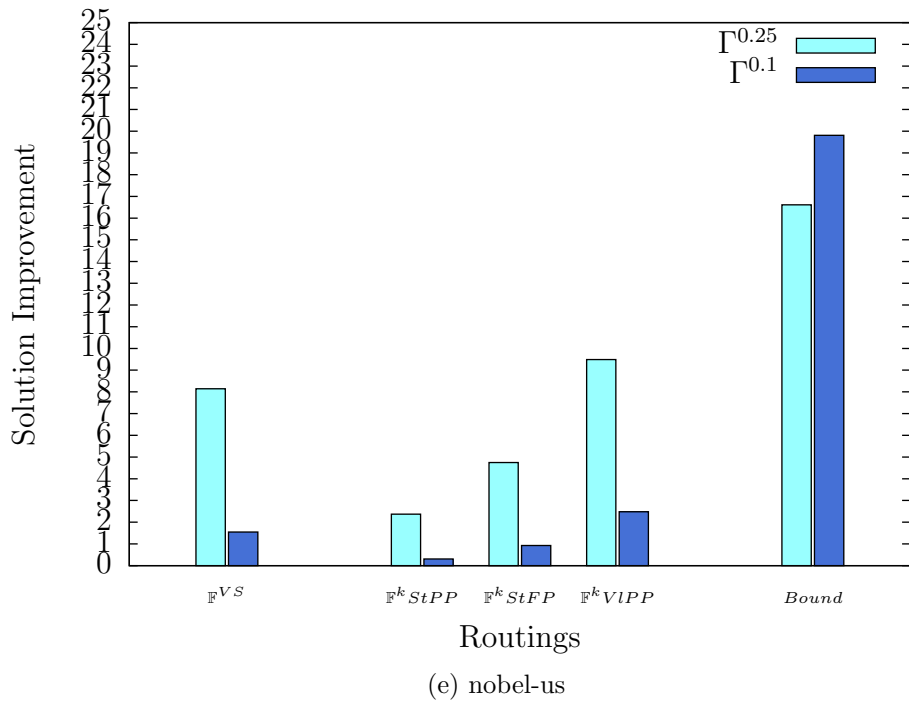
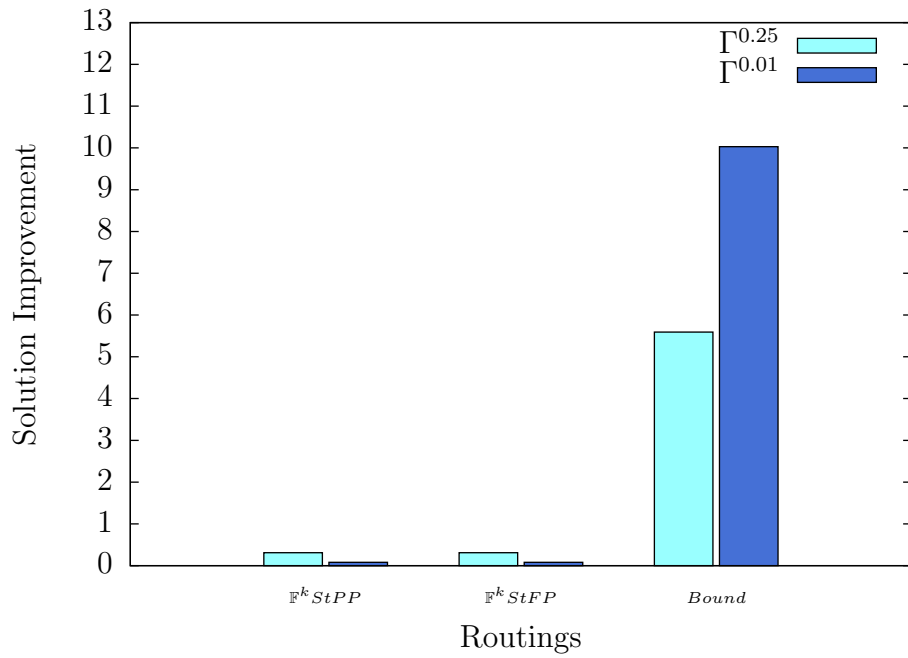
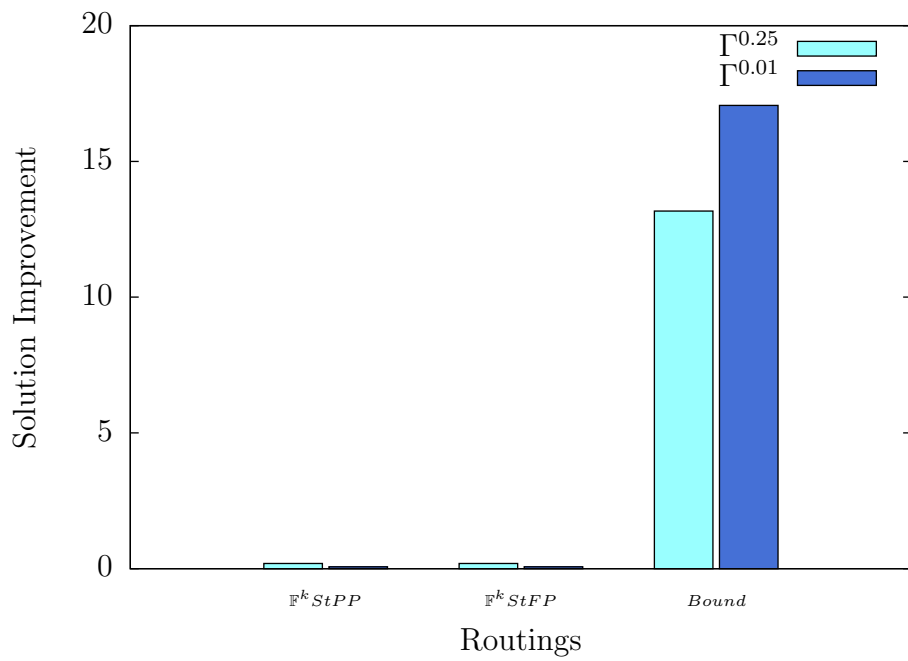


Fig. 1. Part 3: Percentage of Solution Improvement over Static BF Routing

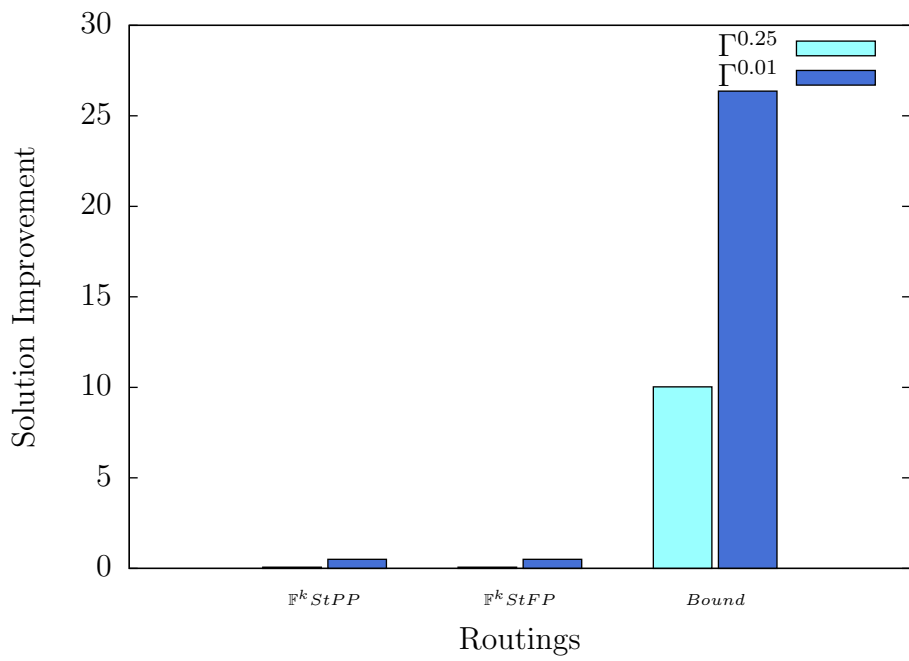


(a) abilene



(b) di-yuan

Fig. 2. Part 1: Percentage of Solution Improvement over Static NBF Routing



(c) atlanta

Fig. 2. Part 2: Percentage of Solution Improvement over Static NBF Routing

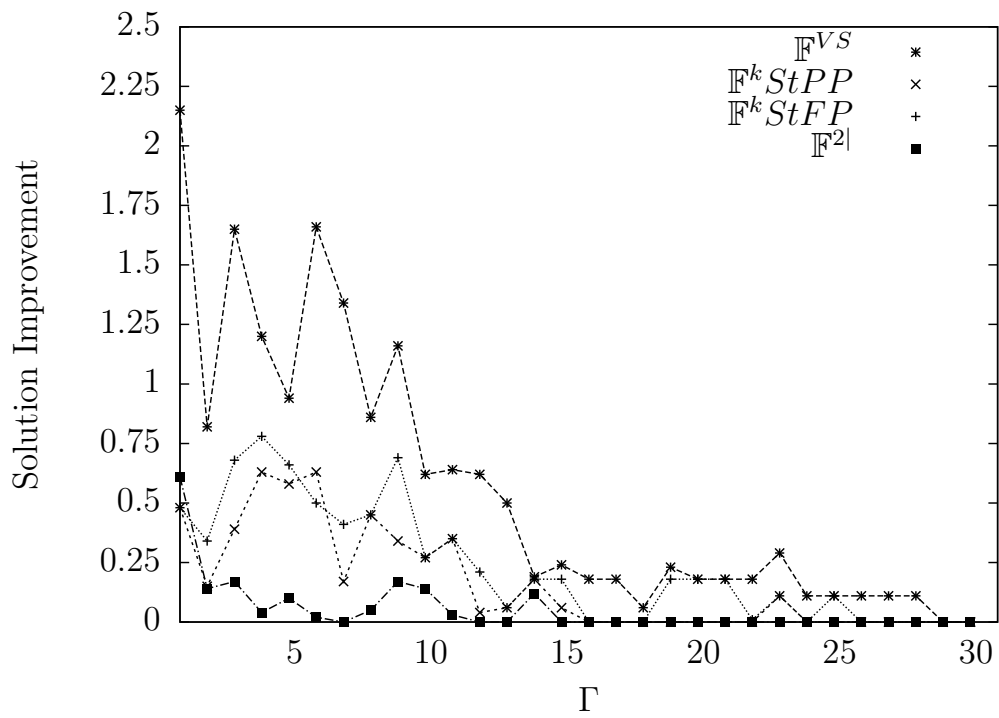


Fig. 3. Percentage of Solution Improvement over Static BF Routing for Instance abilene and different  $\Gamma$ s

		Routing Scheme (all BF) and Algorithm Code								
		$\mathbb{F}^{stat}$	$\mathbb{F}^{VS}$	$\mathbb{F}^k VIPP$		$\mathbb{F}^k VIFP$	$\mathbb{F}^{2l}$	$\mathbb{F}^k StPP$		$\mathbb{F}^k StFP$
Instance	$\Gamma$	C+C	C	C	C+C	C+C	C	C	C+C	C+C
abilene	6	2.20	2.52	3.15	2.14	1.62	71.44	11.03	2.84	4.29
	19	0.55	1.55	543.30	241.79	473.81	3676.84	352.42	53.25	162.87
polska	6	2.52	8.72	4448.67	2761.89	M	M	919.74	409.78	1341.70
	19	5.82	7.77	M	M	M	M	M	M	M
pdh	4	5.24	11.89	4728.78	3815.63	292.49	6534.29	M	5572.47	17.88
	12	11.55	11.15	M	1742.02	1379.20	M	M	3310.10	4336.65
di-yuan	4	0.87	2.42	1648.51	280.80	572.36	6943.07	807.35	152.53	157.43
	11	0.09	0.24	69.53	5.82	1.70	7123.09	101.17	9.86	3.81
nobel-us	7	2.72	1.22	7406.52	2252.20	M	M	1006.32	455.20	425.03
	23	23.09	6.72	M	8838.02	M	M	M	4916.07	5213.05
atlanta	7	1.85	3.88	1688.62	1531.52	M	M	1625.36	1196.10	2034.70
	24	8.11	14.95	M	M	M	M	M	2979.89	M

Table 5  
Bifurcated Flows Total Time Performance (s)

		Algorithms for $\mathbb{F}^k StPPBF$													
		C	C+C	DCP			DCP+RC			DBD			DBD+C		
Instance	$\Gamma$	TT	TT	TT	BC	CSC	TT	BC	CSC	TT	BC	CSC	TT	BC	CSC
abilene	6	11.03	2.84	49.38	69	0	199.46	673	24	12.52	549	0	5.40	255	54
	19	352.42	53.25	410.09	140	0	M	M	M	27.72	758	0	10.49	327	47
polska	6	919.74	409.78	461.35	1124	0	261.54	2791	24	626.19	4435	0	1176.53	5536	53
	19	M	M	1597.32	501	0	976.20	918	24	586.96	2476	0	509.37	1551	63
pdh	4	M	5572.47	2184.11	3294	0	339.46	1571	22	9576.76	9168	0	3933.03	6337	57
	12	M	3310.10	265.93	1122	0	101.34	686	22	209.26	5582	0	92.50	2243	57
di-yuan	4	807.35	152.53	125.65	78	0	52.86	169	22	175.77	3740	0	203.15	2487	74
	11	101.17	9.86	66.36	18	0	20.83	14	22	67.87	2934	0	22.53	1936	66
nobel-us	7	1006.32	455.20	1496.64	4982	0	389.74	2210	28	3285.77	8534	0	1978.23	6997	58
	23	M	4916.07	5911.77	3710	0	1795.92	2805	28	2804.29	6384	0	562.95	3442	56
atlanta	7	1625.36	1196.10	665.17	439	0	309.20	584	30	303.93	3307	0	196.25	2260	30
	24	M	2979.89	3654.80	296	0	2052.01	168	30	381.03	3359	0	314.24	2957	28

Table 6

Algorithms Time Performance and Number of Cuts for Partial Partitioning and Bifurcated Static Routings

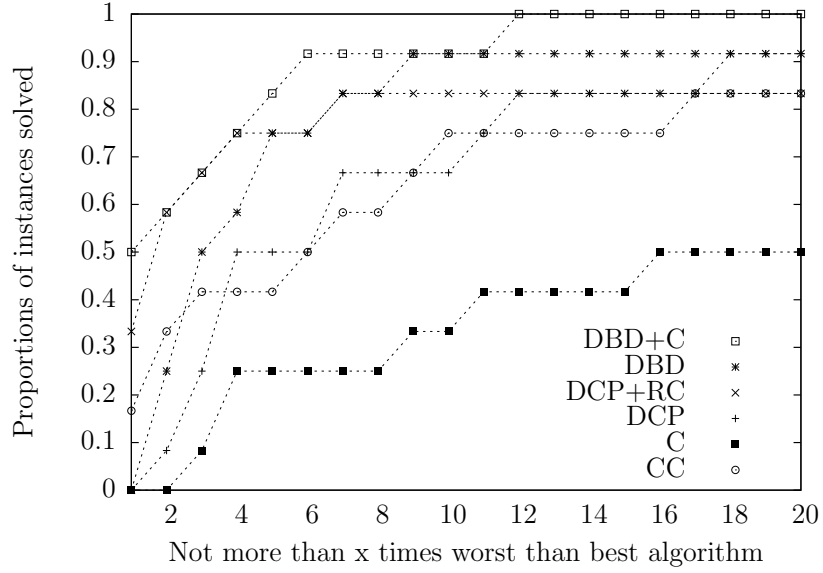


Fig. 4. Performance Profile of Decomposition Algorithms

Routing		Number of Cases of Solution Improvement comparing to Static		
Split	Scheme	$\Gamma^{0.25}$	$\Gamma^{0.01}$	Total
BF	$\mathbb{F}^{VS}$	6 of 6	4 of 6	10 of 12
	$\mathbb{F}^{2 }$	3 of 6	1 of 6	4 of 12
	$\mathbb{F}^k StPP$	6 of 6	4 of 6	10 of 12
	$\mathbb{F}^k StFP$	6 of 6	3 of 6	9 of 12
	$\mathbb{F}^k VI PP$	6 of 6	4 of 6	9 of 12
	$\mathbb{F}^k VI FP$	6 of 6	4 of 6	4 of 12
NBF	$\mathbb{F}^k StPP$	3 of 6	0 of 6	3 of 12
	$\mathbb{F}^k StFP$	3 of 6	0 of 6	3 of 12

Table 7

Summary for Solution Improvement comparison

Comparison	Cases compared*	Number of better Time Performance
Cut-set versus Non cut-set	$\mathbb{F}^k StPP C+C \leq \mathbb{F}^k StPP C$	12 of 12
	$\mathbb{F}^k StPP DBD+C \leq \mathbb{F}^k StPP DBD$	10 of 12
	$\mathbb{F}^k StPP DCP+RC \leq \mathbb{F}^k StPP DCP$	10 of 12
	$\mathbb{F}^k VIPP C+C \leq \mathbb{F}^k VIPP C$	12 of 12
Decomposition versus Non Decomposition	$\mathbb{F}^k StPP DBD+C \leq \mathbb{F}^k StPP C+C$	7 of 12
	$\mathbb{F}^k StPP DBD \leq \mathbb{F}^k StPP C$	10 of 12
	$\mathbb{F}^k StPP DCP+RC \leq \mathbb{F}^k StPP C+C$	9 of 12
	$\mathbb{F}^k StPP DCP \leq \mathbb{F}^k StPP C$	9 of 12
Adapted Benders Decomposition versus CPLEX BENDERS	$\mathbb{F}^k StPP DBD+C \leq \mathbb{F}^k StPP DCP$	8 of 12
	$\mathbb{F}^k StPP DBD+C \leq \mathbb{F}^k StPP DCP+RC$	6 of 12
	$\mathbb{F}^k StPP DBD \leq \mathbb{F}^k StPP DCP$	7 of 12
Partial versus Full Partitioning	$\mathbb{F}^k StPP C+C \leq \mathbb{F}^k StFP C +C$	9 of 12
	$\mathbb{F}^k VIPP C+C \leq \mathbb{F}^k VIFP C +C$	8 of 12
Static Partitioning versus Volume Partitioning	$\mathbb{F}^k StFP C+C \leq \mathbb{F}^k VIFP C+C$	9 of 12
	$\mathbb{F}^k StPP C+C \leq \mathbb{F}^k VIPP C +C$	8 of 12

\*  $\leq$  indicates better than or equal

Table 8  
Summary for Bifurcated Flows Time Performance comparison



## 8 Conclusions

We have performed a numerical experiment in a search to provide efficient routings and less conservative solutions to *RNL*. The  $k$ -adaptive routing scheme was able to provide good solutions and time performance when compared to other partitioning algorithms. In particular, the extension of  $k$ -adaptive volume routing scheme was the routing to provide better solutions among the routings experimented. The method utilized does suffer of dimensionality issues so that special techniques to control time performance are fundamental. In fact, the  $k$ -adaptive partial partitioning can provide good results, when compared to full partitioning and have better time performance. Our results also showed that Benders decomposition can be efficient to speed up instances.

## References

- [1] Aysegül Altın, Edoardo Amaldi, Pietro Belotti, and Mustafa Ç. Pinar. Provisioning virtual private networks under traffic uncertainty. *Networks*, 49(1):100–115, 2007.
- [2] Hande Yaman Mustafa Ç. Pinar Aysegül Altın. The Robust Network Loading Problem Under Hose Demand Uncertainty: Formulation, Polyhedral Analysis, and Computations. *INFORMS Journal on Computing*, 23(1):75–89, 2011.
- [3] Josette Ayoub and Michael Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Comput. Manag. Science*, 13(2):219–239, 2016.
- [4] Andreas Börmann, Andreas Heide, Alexander Martin, Sebastian Pokutta, and Christoph Thurner. Polyhedral approximation of ellipsoidal uncertainty sets via extended formulations: a computational case study. *Comput. Manag. Science*, 13(2):151–193, 2016.
- [5] Andreas Börmann, Andreas Heide, Alexander Martin, Sebastian Pokutta, and Christoph Thurner. Erratum to: Polyhedral approximation of ellipsoidal uncertainty sets via extended formulations: a computational case study. *Comput. Manag. Science*, 14(2):293–296, 2017.
- [6] Walid Ben-Ameur. Between fully dynamic routing and robust stable routing. In *6th International Workshop on Design and Reliable Communication Networks, DRCN*, 2007.

- [7] Walid Ben-Ameur and Hervé Kerivin. Routing of Uncertain Traffic Demands. *Optimization and Engineering*, 6(3):283–313, 2005.
- [8] Walid Ben-Ameur and Mateusz Zotkiewicz. Multipolar routing: where dynamic and static routing meet. *Electronic Notes in Discrete Mathematics*, 41:61–68, 2013.
- [9] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, October 2009.
- [10] Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Math. Oper. Res.*, 23(4):769–805, 1998.
- [11] D. Bertsimas and C. Caramanis. Finite Adaptability in Multistage Linear Optimization. *IEEE Transactions on Automatic Control*, 55(12):2751–2766, Dec 2010.
- [12] Dimitris Bertsimas and Iain Dunning. Multistage Robust Mixed-Integer Optimization with Adaptive Partitions. *Oper Res*, 64(4):980–998, 2016.
- [13] Dimitris Bertsimas and Melvyn Sim. The Price of Robustness. *Oper. Res.*, 52(1):35–53, January 2004.
- [14] Christina Büsing and Fabio D’Andreagiovanni. A new theoretical framework for robust optimization under multi-band uncertainty. In *Operations Research Proceedings, German Operations Research Society (GOR)*, pages 115–121, 2012.
- [15] Chandra Chekuri. Routing and Network Design with Robustness to Changing or Uncertain Traffic Demands. *SIGACT News*, 38(3):106–129, September 2007.
- [16] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *CoRR*, cs.MS/0102001, 2001.
- [17] Nick G. Duffield, Pawan Goyal, Albert G. Greenberg, Partho Pratim Mishra, K. K. Ramakrishnan, and Jacobus E. van der Merwe. A Flexible Model for Resource Management in Virtual Private Networks. In *SIGCOMM*, pages 95–108, 1999.
- [18] J. Andrew Fingerhut, Subhash Suri, and Jonathan S. Turner. Designing Least-Cost Nonblocking Broadband Networks. *J. Algorithms*, 24(2):287–309, 1997.
- [19] M Lucertini Giuseppe Paletta. Network Design with Multiple Demand: A New Approach. In *North-Holland Mathematics Studies*, 1986.
- [20] Navin Goyal, Neil Olver, and F. Bruce Shepherd. *Dynamic vs. Oblivious Routing in Network Design*, pages 277–288. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

- [21] Arie M. C. A. Koster, Manuel Kutschka, and Christian Raack. Robust network design: Formulations, valid inequalities, and computations. *Networks*, 61(2):128–149, 2013.
- [22] P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications (Nonconvex Optimization and Its Applications)*. Springer, 1st edition, November 1997.
- [23] Chungmok Lee, Kyungsik Lee, Kyungchul Park, and Sungsoo Park. Technical note - branch-and-price-and-cut approach to the robust network design problem without flow bifurcations. *Operations Research*, 60(3):604–610, 2012.
- [24] Chungmok Lee, Kyungsik Lee, and Sungsoo Park. Benders decomposition approach for the robust network design problem with flow bifurcations. *Networks*, 62(1):1–16, 2013.
- [25] Miles Lubin and Iain Dunning. Computing in Operations Research using Julia. *CoRR*, abs/1312.1431, 2013.
- [26] Sara Mattia. The cut condition for robust network design. *Electronic Notes in Discrete Mathematics*, 41:303–310, 2013.
- [27] Sara Mattia. The robust network loading problem with dynamic routing. *Comp. Opt. and Appl.*, 54(3):619–643, 2013.
- [28] Sara Mattia and Michael Poss. A comparison of different routing schemes for the robust network loading problem: polyhedral results and computation. *Submitted, Available at [www.optimization-online.org](http://www.optimization-online.org)*, 2014.
- [29] Michel Minoux. Robust network optimization under polyhedral demand uncertainty is NP-hard. *Discrete Applied Mathematics*, 158(5):597–603, 2010.
- [30] S. Mudchanatongsuk, F. Ordóñez, and J. Liu. Robust solutions for network design under transportation cost and demand uncertainty. *Journal of the Operational Research Society*, 59(5):652–662, 2008.
- [31] O. Nohadani and K. Sharma. Optimization under Decision-Dependent Uncertainty. *ArXiv e-prints*, November 2016.
- [32] Sebastian Orlowski, Roland Wessäly, Michal Pióro, and Artur Tomaszewski. SNDlib 1.0 - Survivable Network Design Library. *Networks*, 55(3):276–286, 2010.
- [33] A. Ouorou and J.-P. Vial. A model for robust capacity planning for telecommunications networks under demand uncertainty. *Proc 6th Int Workshop Design Reliable Commun Networks*, pages 1–4, 2007.

- [34] Adam Ouorou. Tractable approximations to a robust capacity assignment model in telecommunications under demand uncertainty. *Computers & OR*, 40(1):318–327, 2013.
- [35] Michael Poss. Robust combinatorial optimization with variable budgeted uncertainty. *4OR*, 11(1):75–92, 2013.
- [36] Michael Poss. A comparison of routing sets for robust network design. *Optimization Letters*, 8(5):1619–1635, 2014.
- [37] Michael Poss and Christian Raack. Affine recourse for the robust network design problem: Between static and dynamic routing. *Networks*, 61(2):180–198, 2013.
- [38] Krzysztof Postek and Dick den Hertog. Multistage Adjustable Robust Mixed-Integer Optimization via Iterative Splitting of the Uncertainty Set. *INFORMS Journal on Computing*, 28(3):553–574, 2016.
- [39] Maria Grazia Scutellà. On improving optimal oblivious routing. *Oper. Res. Lett.*, 37(3):197–200, 2009.
- [40] Marco Silva, Michael Poss, and Nelson Maculan. k-Adaptive Routing for the Robust Network Loading Problem. In *accepted INOC2017*, 2017.
- [41] Mateusz Zotkiewicz and Walid Ben-Ameur. More Adaptive Robust Stable Routing. In *Proceedings of the Global Communications Conference, 2009. GLOBECOM 2009, Honolulu, Hawaii, USA, 30 November - 4 December 2009*, pages 1–6, 2009.
- [42] Mateusz Zotkiewicz and Walid Ben-Ameur. Volume-oriented routing and its modifications. *Telecommunication Systems*, 52(2):935–945, 2013.