A comparison of different routing schemes for the robust network loading problem: polyhedral results and computation — Source link

Sara Mattia, Michael Poss

Institutions: University of Montpellier

Published on: 01 Apr 2018 - Computational Optimization and Applications (Springer US)

Topics: Static routing and Routing (electronic design automation)

Related papers:

- The robust network loading problem with dynamic routing

- Dynamic vs. Oblivious Routing in Network Design

- A General Approach to Dynamic Packet Routing with Bounded Buffers (Extended abstract)

- Parity Balancing Path Flow Decomposition and Routing

- A general approach to dynamic packet routing with bounded buffers

A comparison of different routing schemes for the robust network loading problem: polyhedral results and computation

Sara Mattia, Michael Poss

A comparison of different routing schemes for the robust network loading problem: polyhedral results and computation

Sara Mattia*   Michael Poss†

Abstract

We consider the capacity formulation of the Robust Network Loading Problem. The aim of the paper is to study what happens from the theoretical and from the computational point of view when the routing policy (or scheme) changes. The theoretical results consider static, volume, affine and dynamic routing, along with splittable and unsplittable flows. Our polyhedral study provides evidence that some well-known valid inequalities (the robust cutset inequalities) are facets for all the considered routing/flows policies under the same assumptions. We also introduce a new class of valid inequalities, the robust 3-partition inequalities, showing that, instead, they are facets in some settings, but not in others. A branch-and-cut algorithm is also proposed and tested. The computational experiments refer to the problem with splittable flows and the budgeted uncertainty set. We report results on several instances coming from real-life networks, also including historical traffic data, as well as on randomly generated instances. Our results show that the problem with static and volume routing can be solved quite efficiently in practice and that, in many cases, volume routing is cheaper than static routing, thus possibly representing the best compromise between cost and computing time. Moreover, unlikely from what one may expect, the problem with dynamic routing is easier to solve than the one with affine routing, which is hardly tractable, even using decomposition methods.

Keywords: robust network loading, budgeted uncertainty, Benders decomposition, static routing, volume routing, affine routing, dynamic routing.

1 Introduction

Given an undirected graph and a set of point-to-point commodities with known demands (traffic matrix), the objective of the network design problem is to find the cheapest capacity installation on the edges of the graph such that the resulting network supports the routing of the commodities. The problem has numerous applications in telecommunications, transportation, and energy management, among many others. Accordingly, a large number of variations can be defined, which restrict for instance, the type of flows admissible on the edges or the type of capacities that can be installed on the edges. A large variety of technical constraints can also be considered. They include ensuring a given level of survivability in case of link failures or limiting the length of the paths used. Different flows policies may be used: the demands may be restricted to be routed on single paths (*unsplittable flows*) or the flows may be unrestricted (*splittable flows*) [9]. Herein, we focus on the so-called Network Loading Problem, where capacities can be installed by integer multiples. An important aspect of network design problems is related to the knowledge of the demands. In a large number of applications, these demands are not available at the time we decide on the capacity installation. Here we refer to the Network Loading Problem under uncertain demands as the Robust Network Loading Problem. A way to deal with demand uncertainty is to rely on demand forecasts, based, for instance, on population statistics [18] or on traffic measurements [48]. If these statistical studies are accurate enough, one can come up with a stochastic model that considers demands as known random

---

*S. Mattia. Istituto di Analisi dei Sistemi ed Informatica. Consiglio Nazionale delle Ricerche. Via dei Taurini 19, 00185 Roma (Italy). Email: sara.mattia@iasi.cnr.it.

†M. Poss. UMR CNRS 5506 LIRMM. Université de Montpellier. Rue Ada 161, 34095 Montpellier cedex 5, (France). Email: michael.poss@lirmm.fr. Corresponding author.

variables, typically leading to two-stage stochastic programs. See [5] and the references therein for additional details. Unfortunately, it is very difficult in practice to come up with an accurate description of these random variables. An alternative approach is to define sets of admissible values of random variables compatible with the available data, falling into the framework of distributionally robust optimization [46]. In this work, we assume that the uncertain demands are described through an uncertainty set [11], falling into the framework of robust optimization [13]. Hence, the problem turns to designing a network able to route each traffic matrix in the uncertainty set. Although conservative, this approach has been used extensively in recent years to model demand uncertainty in telecommunications and transportation networks [4, 8, 26, 33, 34, 37, 38, 44]. A popular choice to model the forecast demands is to use the budgeted uncertainty set [15]. The latter supposes that the demands fluctuate between their nominal values and given peak values and that at most Γ of them reach their peak values simultaneously. The model is motivated by probabilistic guarantees [15] and has been used in numerous papers on robust network design problems [8, 34, 44].

The introduction of uncertainty in demands raises the question of how to adapt the flows to different realizations of the demand. This concept is often referred to as *routing* in the literature on network optimization. Different routing policies (or schemes) have been studied in the past, each with its own flexibility and computational issues [4, 33, 44]. At one extreme, we find *static routing*, which imposes that the fractional splitting of the commodities among a fixed set of paths stays constant for all realizations of the demands. In the other extreme, *dynamic routing* allows the flows to be changed completely any time that the demands change. Both approaches have advantages and drawbacks. Dynamic routing is more flexible, but the corresponding problem is difficult to solve. In addition, dynamic routing can be difficult to implement in practice because the routing depends on the current status of all the demands in the network, thus, hardening decentralization. On the other hand, static problems are usually computationally more tractable and easier to implement in decentralized environments, but the corresponding solutions may be too conservative. Therefore, in the last years, several intermediate routing schemes have been proposed. They include *affine* and *volume routing*. The aim was to obtain more flexibility than static routing, solving a problem that is theoretically easier than the dynamic one. Affine routing [41] restricts the flows to be affine functions of the demands, as it applies to network optimization what has long been known as affine decision rules in adjustable robust optimization [14]. Affine routing has been used in several papers on robust network design, see for instance [8, 28, 40, 44], including a variation of the problem where it is the capacity, rather than the demand, which is uncertain [42]. Volume routing [12] is a special case of affine routing, where the set of paths used for each commodity can be adjusted according to the current value of the demand for that commodity. In addition to its numerical tractability, volume routing is easier to implement in a decentralized environment than affine and dynamic routing, since the routing for a commodity only depends on the demand for that commodity. Other intermediate routings have also been proposed in the literature, such as those based on dynamic partitions of the uncertainty set [10, 45]. However, they lead to optimization problems that are even harder to solve than the problem with dynamic routing [43]. For this reason, we do not consider them in what follows.

The computational tractability of the Robust Network Loading Problem has been studied under static, affine and dynamic routing. The problem is $\mathcal{NP}$-hard independently of the routing scheme, as it includes the problem without uncertainty, and then the Steiner tree problem, as a special case. However, even for the splittable case, when the integrality restrictions on the capacity variables are relaxed, the complexity of the problem does depend on the routing. For affine, volume and static routing there exists a compact formulation, that is, a formulation with a polynomial number of variables and constraints, and, hence, the problem can be solved in polynomial time [3, 12, 44]. In contrast, the problem with dynamic routing was proved to be $\mathcal{NP}$-hard, both in general and for the budgeted uncertainty set [20, 25, 36]. Therefore, no compact formulation exists, unless $\mathcal{P} = \mathcal{NP}$. As a consequence, for static/affine/volume routing separation is polynomial, whereas for dynamic routing it is $\mathcal{NP}$-hard [33]. This affects the computational performance of the problem with integer capacities. For splittable flows and static routing, in [4] the authors propose a polyhedral investigation and numerical results for the problem assuming that the demand uncertainty polytope is the Hose model [22, 23]. Similar numerical studies have been carried out in [26] for the problem with

2

budgeted uncertainty. In [27] a Benders decomposition approach for the splittable Robust Network Loading Problem with static routing and budgeted uncertainty is derived and studied. In [33] the author studies the problem with dynamic routing and splittable flows under the Hose model, proposing a branch-and-cut procedure related to bilevel optimization. A special case of that problem is studied in [19], where the authors focus on single-commodity problems, which allows them to provide more efficient inequalities and separation routines. Finally, in [37] the problem with splittable flows, affine routing and polyhedral or ellipsoidal uncertainty sets is studied. The authors further consider a version of the problem with restrictions on the set of feasible paths and propose column generation algorithms.

In this paper we study the capacity formulation of the Robust Network Loading Problem, that is, a formulation including only design variables. The scope of the paper is to investigate what happens both theoretically and computationally when the routing policy changes. To the best of our knowledge, this is the first time that such a comprehensive investigation is carried out. From the theoretical point of view, a first contribution of the paper is to derive a capacity formulation for the problem with volume and affine routing using a Benders decomposition approach. For dynamic routing, the exponential number of variables and constraints in the formulation prevents us from using the flow formulation as a starting point for deriving a capacity formulation, and hence, from applying the classical Benders reformulation. Instead, we must draw from the more advanced tools proposed in the recent years for adjustable robust optimization, see [7, 33, 47]. We refer to the formulations including only design variables as Benders formulations in the rest of the paper, despite the differences in the techniques used to obtain them. We provide polyhedral results characterizing the convex-hull of integer feasible solutions of the problem for all the considered routing schemes under splittable and unsplittable flows, for a general uncertainty set. We establish relations among the polyhedra corresponding to the considered routing schemes, giving conditions for an inequality that is valid or facet under a given routing/flows policy pair to be valid or facet for another routing/flows pair. Following the comments in [33], we formally prove that the well-known robust cutset inequalities [26, 33, 34] are facets under the same assumptions in all the considered settings. Actually, this is, so far, the only class of inequalities having such a property, but for the non-negativity constraints. Indeed, we present a new class of valid inequalities for the Robust Network Loading Problem, the robust 3-partition inequalities, and show that they are facet-defining for the problem with dynamic routing and splittable flows, whereas they are not facets, under the same assumptions, for the other routing schemes or for unsplittable flows. They provide the first example of such a behavior. From a computational perspective, we investigate what is the effect of the different routing schemes on costs and computing times, when the splittable problem is solved using the budgeted uncertainty set, the Benders formulations and the cuts investigated in the theoretical part. We provide exact and heuristic separation routines for the robust cutset inequalities, a heuristic approach for finding a violated robust 3-partition inequality, an exact approach for the Benders cuts and a primal heuristic. We compare the Benders formulations using many different cutting plane approaches. We also investigate the compact flow formulation for volume, affine and static routing. Although compact, the corresponding problem may be time consuming for some of the considered routing schemes. However, we show that its performance can be significantly improved by separating the above mentioned inequalities in the branch-and-cut tree. We report computational experiments on real-life instances, including instances based on historical traffic data [39], as well as on randomly generated ones. We prove that the problem with static and volume routing can be solved quite efficiently in practice on the considered instances. Moreover, unlikely from what one may expect, the problem with dynamic routing is easier to solve than the one with affine routing. It turns out that all the considered routing schemes (but possibly affine routing) are able to solve real-life problems. We also show that volume routing yields cost reductions over static routing in half of the instances, while not requiring more computational time. This is important because many papers use static routing, whereas volume routing is not so popular.

The paper is structured as follows. Sections 2 and Section 3 present, respectively, the so-called flow and Benders formulations of the Robust Network Loading Problem for each of the aforementioned routing schemes. In Section 4 we give polyhedral results. In Section 5 we present implementation

details and the test-bed. In Sections 6 we discuss computational experiments. The paper is concluded in Section 7. When presenting models and computational results (Sections 2, 3, 5, 6), we restrict to splittable flows and to the budgeted uncertainty set, whereas the theoretical results (Section 4) are more general and they also hold for unsplittable flows and are independent of the uncertainty set.

2 Flow formulations

Let $G(V, E)$ be an undirected graph without loops and parallel edges, let K be the set of point-to-point commodities to be routed on the network and assume that all the demands belonging to a given uncertainty set $\mathcal{U} \subset \mathbb{R}_+^{|K|}$ must be served. Each commodity $k \in K$ is defined by its endnodes s^k and t^k and its demand value d^k for any $\mathbf{d} \in \mathcal{U}$. In presenting the models, we suppose that the flows are splittable and that the uncertainty set has a special structure, often used in the literature [15]: each demand value d^k varies between its nominal value $\bar{d}^k$ and its peak value $\bar{d}^k + \hat{d}^k$ and the number of deviations from the nominal value is bounded by integer Γ. It corresponds to the extended formulation below and we denote it by $\mathcal{U}_b$.

$$\mathcal{U}_b \equiv \left\{ \mathbf{d} \in \mathbb{R}_+^{|K|} \mid \exists \boldsymbol{\delta} \in [0, 1]^{|K|} : d^k = \bar{d}^k + \delta^k \hat{d}^k, k \in K, \sum_{k \in K} \delta^k \leq \Gamma \right\} \tag{1}$$

In what follows we use $\mathcal{U}$ when the formulation or the result refers to a general uncertainty set, whereas we use $\mathcal{U}_b$ when we specifically refer to the budgeted uncertainty set. We do not make any special assumption on $\mathcal{U}$, but that it is bounded (otherwise the resulting problem is unbounded) and non-empty (otherwise the resulting problem is trivial). By non-empty we also mean that $\mathcal{U}$ is different from the singleton $\{\mathbf{0}\}$, as the resulting problem admits the trivial solution $\mathbf{x} = \mathbf{0}$, as well. We associate to E the set of directed arcs A: for each $e = \{i, j\} \in E$, we create two directed arcs (i, j) and (j, i). The unitary cost of installing capacity on edge $e \in E$ is given by c_e. The Robust Network Loading Problem studied herein aims at installing the cheapest capacities $\mathbf{x}$ on the edges of the graph, such that all realizations of the demand vectors $\mathbf{d} \in \mathcal{U}$ can be routed on the resulting network.

2.1 Dynamic routing

The problem with dynamic routing can be formulated mathematically as follows. The integer variable x_e represents the capacity allocation on edge $e \in E$ and the real variable $f_{ij}^k(\mathbf{d})$ describes the amount of flow for commodity k routed on arc $(i, j) \in A$, when considering demand $\mathbf{d} \in \mathcal{U}$. We also define the star of node $i \in N$ as $N(i) = \{j \in N : \exists e = \{i, j\} \in E\}$.

$$fDRNL \min \sum_{e \in E} c_e x_e$$

$$\text{s.t.} \sum_{j \in N(i)} (f_{ji}^k(\mathbf{d}) - f_{ij}^k(\mathbf{d})) = \begin{cases} d^k & \text{if } i = t^k \\ 0 & \text{otherwise} \end{cases} \quad \begin{array}{l} i \in V \setminus \{s^k\}, \\ k \in K, \mathbf{d} \in \mathcal{U} \end{array} \tag{2a}$$

$$\sum_{k \in K} (f_{ij}^k(\mathbf{d}) + f_{ji}^k(\mathbf{d})) \leq x_e \quad e = \{i, j\} \in E, \mathbf{d} \in \mathcal{U} \tag{2b}$$

$$\mathbf{f}, \mathbf{x} \geq 0 \tag{2c}$$

$$\mathbf{x} \in \mathbb{Z}^{|E|} \tag{2d}$$

Constraints (2a) represent flow conservation constraints at every node of the network (constraints for s^k are not included because they are redundant) and constraints (2b) impose that the amount of flow on each edge does not exceed the available capacity on that edge. In the following, we call *routing* the flow function $\mathbf{f}$. We use the term *dynamic routing* when no particular assumption is made on admissible functions, as in $fDRNL$. Problem $fDRNL$ is a mixed integer linear programming problem with an infinite number of variables and constraints. However, we see easily that the problem can be discretized by considering the extreme points of the demand polyhedron, denoted by $\text{vert}(\mathcal{U})$,

yielding a finite mixed-integer linear formulation. However, when some uncertainty sets are used, the resulting formulation is extremely large, since the number of extreme points of $\mathcal{U}$ may grow exponentially. This is the case for $\mathcal{U}_b$, where

$$|\operatorname{vert}(\mathcal{U}_b)| = \sum_{l=0}^{\Gamma} \binom{\Gamma}{l}.$$

One way to cope with such a large formulation is to use a decomposition approach to generate only a subset of the extreme points on the fly in the course of branch-and-cut algorithms. We explain in the next section how Benders decomposition can be used to do that. Alternatively, we could restrict the routing to simple functions of $\mathbf{d}$.

2.2 Affine routing

Rather than letting $\mathbf{f}$ be an arbitrary function of $\mathbf{d}$, we can enforce that the following restrictions be satisfied.

$$f_{ij}^k(\mathbf{d}) = f_{ij}^{0k} + \sum_{h \in K} y_{ij}^{kh} d^h \quad k \in K, (i,j) \in A. \tag{3}$$

Constraints (3) yields what is known as affine decision rule or *affine routing* in the literature [40, 41, 44]. The constraints limit $\mathbf{f}$ to be an affine function of $\mathbf{d}$. Here we explain how to obtain a compact reformulation for the affine problem from $fDRNL$ using constraints (3) and linear programming duality, for $\mathcal{U}_b$. The reader is referred to [44] for further details. Plugging (3) into $fDRNL$ yields

$$\min \quad \sum_{e \in E} c_e x_e$$

$$\text{s.t.} \quad \sum_{j \in N(i)} \left(f_{ij}^{0k} + \sum_{h \in K} y_{ij}^{kh} d^h - (f_{ij}^{0k} + \sum_{h \in K} y_{ij}^{kh} d^h) \right) = \begin{cases} d^k & \text{if } i = t^k \\ 0 & \text{otherwise} \end{cases} \quad \begin{array}{l} i \in V \setminus \{s^k\}, \\ k \in K, \mathbf{d} \in \mathcal{U} \end{array} \tag{4a}$$

$$\sum_{k \in K} \left(f_{ij}^{0k} + \sum_{h \in K} y_{ij}^{kh} d^h + f_{ji}^{0k} + \sum_{h \in K} y_{ji}^{kh} d^h \right) \le x_e \quad \begin{array}{l} e = \{i,j\} \in E, \\ \mathbf{d} \in \mathcal{U} \end{array} \tag{4b}$$

$$f_{ij}^{0k} + \sum_{h \in K} y_{ij}^{kh} d^h \ge 0 \quad \begin{array}{l} (i,j) \in A, \\ k \in K \end{array} \tag{4c}$$

$$\mathbf{x} \ge 0, \mathbf{x} \in \mathbb{Z}^{|E|} \tag{4d}$$

Consider first equations (4a) and group the terms according to their dependency on $\mathbf{d}$. Then, we consider $\bar{\mathbf{d}} + \epsilon \mathbf{e}^h$ where $e_h^h = 1$ and $e_k^h = 0$ for each $k \ne h$. Subtracting (4a) written for $\bar{\mathbf{d}}$ from (4a) written for $\bar{\mathbf{d}} + \epsilon \mathbf{e}^h$ yields the reformulation (6a)–(6c). Consider now constraints (4b) and use the definition $d^k = \bar{d}^k + \delta^k \hat{d}^k$. They can be rewritten as

$$\sum_{k \in K} (f_{ij}^{0k} + f_{ji}^{0k}) + \sum_{h \in K} \bar{d}^h \sum_{k \in K} (y_{ij}^{kh} + y_{ji}^{kh}) + \max_{\boldsymbol{\delta} \in [0,1]^{|K|} : \sum_h \delta^h \le \Gamma} \sum_{h \in K} \delta^h \hat{d}^h \sum_{k \in K} (y_{ij}^{kh} + y_{ji}^{kh}) \le x_e. \tag{5}$$

Replacing the inner maximization problem of (5) by its dual yields (6d) and (6e), where z_e and p_e^k are the dual variables corresponding to constraints $\sum_h \delta^h \le \Gamma$ and $\delta^k \le 1$, respectively. Proceeding similarly with (4c) yields (6f) and (6g). Then, the flow formulation of the affine problem is the following.

$$fARNL \quad \min \quad \sum_{e \in E} c_e x_e$$

$$\text{s.t.} \quad \sum_{j \in N(i)} (y_{ij}^k - y_{ji}^k) = \begin{cases} 1 & \text{if } i = t^k \\ 0 & \text{otherwise} \end{cases} \quad \begin{array}{l} i \in V \setminus \{s^k\}, \\ k \in K \end{array} \tag{6a}$$

5

$$\sum_{j\in N(i)} (y_{ji}^{kh} - y_{ij}^{kh}) = 0 \qquad\qquad \begin{array}{l} i \in V \setminus \{s^k\}, \\ k \neq h \in K \end{array} \tag{6b}$$

$$\sum_{j\in N(i)} (f_{ji}^{0k} - f_{ij}^{0k}) = 0 \qquad\qquad \begin{array}{l} i \in V \setminus \{s^k\}, \\ k \in K \end{array} \tag{6c}$$

$$\Gamma z_e + \sum_{k\in K} \left(p_e^k + f_{ij}^{0k} + f_{ji}^{0k} + \sum_{h\in K}(y_{ij}^{kh} + y_{ji}^{kh})\bar{d}^h \right) \leq x_e \qquad e = \{i,j\} \in E \tag{6d}$$

$$z_e + p_e^k \geq \sum_{k\in K}(y_{ij}^{kh} + y_{ji}^{kh})\hat{d}^h \qquad\qquad \begin{array}{l} e = \{i,j\} \in E, \\ h \in K \end{array} \tag{6e}$$

$$\Gamma s_{ij} - f_{ij}^{0k} + \sum_{h\in K} \left(q_{ij}^{kh} - y_{ij}^{kh}\bar{d}^h \right) \leq 0 \qquad\qquad \begin{array}{l} (i,j) \in A, \\ k \in K \end{array} \tag{6f}$$

$$s_{ij} + q_{ij}^{kh} \geq y_{ij}^{kh}\hat{d}^h \qquad\qquad \begin{array}{l} (i,j) \in A, \\ k \in K, h \in K \end{array} \tag{6g}$$

$$\mathbf{z}, \mathbf{s}, \mathbf{p}, \mathbf{q}, \mathbf{x} \geq 0 \tag{6h}$$

$$\mathbf{x} \in \mathbb{Z}^{|E|} \tag{6i}$$

Variables $\mathbf{z}, \mathbf{s}, \mathbf{p}, \mathbf{q}$ come from dualizing the robust constraints, whereas $\mathbf{f}^0$ and $\mathbf{y}$ come from the restrictions imposed to the routing by (3). Although compact, formulation $fARNL$ is quadratic in $|K|$, which makes the problem significantly harder to solve than its deterministic counterpart. As we will show in our numerical experiments, the dimension of the reformulation is such that it even its linear programming relaxation can hardly be solved in reasonable amounts of time. To avoid this quadratic dependency on $|K|$, we can consider special cases of affine routing that contain less degrees of freedom than (3). Below we present two such cases and we show how the corresponding formulations can be derived from $fARNL$.

2.3 Volume routing

We enforce $\mathbf{f}$ to satisfy

$$f_{ij}^k(\mathbf{d}) = f_{ij}^{0k} + y_{ij}^{kk}d^k, \quad k \in K, (i,j) \in A \tag{7}$$

obtaining a subclass of affine routing known as *volume routing*. Roughly speaking, (7) means that the flows are defined by a set of paths from $s(k)$ to $t(k)$ (associated with $\mathbf{y}$), modified by the circulation described by $\mathbf{f}^0$. Volume routing has been originally introduced in [12]. Plugging constraints (7) into $fDRNL$ and applying the techniques mentioned above, yields the following reformulation for the problem:

$$fVRNL \quad \min \quad \sum_{e\in E} c_e x_e$$

$$\text{s.t.} \quad \sum_{j\in N(i)} (y_{ji}^k - y_{ij}^k) = \begin{cases} 1 & \text{if } i = t^k \\ 0 & \text{otherwise} \end{cases} \qquad \begin{array}{l} i \in V \setminus \{s^k\}, \\ k \in K \end{array} \tag{8a}$$

$$\sum_{j\in N(i)} (f_{ji}^{0k} - f_{ij}^{0k}) = 0 \qquad\qquad i \in V \setminus \{s^k\}, k \in K \tag{8b}$$

$$\Gamma z_e + \sum_{k\in K} \left(p_e^k + f_{ij}^{0k} + f_{ji}^{0k} + (y_{ij}^{kk} + y_{ji}^{kk})\bar{d}^k \right) \leq x_e \qquad e = \{i,j\} \in E \tag{8c}$$

$$z_e + p_e^k \geq (y_{ij}^{kk} + y_{ji}^{kk})\hat{d}^k \qquad\qquad \begin{array}{l} e = \{i,j\} \in E, \\ k \in K \end{array} \tag{8d}$$

$$f_{ij}^{0k} + y_{ij}^{kk}\bar{d}^k \geq 0 \qquad\qquad (i,j) \in A, k \in K \tag{8e}$$

$$f_{ij}^{0k} + y_{ij}^{kk}(\bar{d}^k + \hat{d}^k) \geq 0 \qquad\qquad (i,j) \in A, k \in K \tag{8f}$$

$$\mathbf{z}, \mathbf{p}, \mathbf{x} \geq 0 \tag{8g}$$

6

$$\mathbf{x} \in \mathbb{Z}^{|E|} \tag{8h}$$

Formulation (8) can be derived from formulation (6) by removing flow conservation constraints (6c), replacing robust non-negativity constraints (6f) and (6g) by (8e) – (8f) and removing the terms corresponding to y^{kh} for $h \neq k$ in robust capacity constraints (6d) – (6e). Notice that other authors have proposed different restrictions of affine routing to reduce the size of the reformulation [8].

2.4   Static routing

We enforce f to satisfy

$$f_{ij}^k(\mathbf{d}) = y_{ij}^{kk} d^k, \quad k \in K, (i,j) \in A, \tag{9}$$

obtaining a subclass of volume routing, known as *static routing*. Static routing is a well-known framework for the Robust Network Loading Problem and it has been studied long before affine routing was introduced [22, 23]. When constraints (9) are satisfied, variables $\mathbf{y}$ are often called *routing template*, because they represent the fractional splittings of demands along the paths from s^k to t^k for each $k \in K$. Plugging constraints (9) into $fDRNL$ and applying the techniques mentioned above, yields the following reformulation for the problem:

$$fSRNL \quad \min \sum_{e \in E} c_e x_e$$

$$\text{s.t.} \quad \sum_{j \in N(i)} (y_{ji}^k - y_{ij}^k) = \left\{ \begin{array}{ll} 1 & \text{if } i = t^k \\ 0 & \text{otherwise} \end{array} \right. \quad i \in V \setminus \{s^k\}, k \in K \tag{10a}$$

$$\Gamma z_e + \sum_{k \in K} \left(p_e^k + (y_{ij}^{kk} + y_{ji}^{kk}) \bar{d}^k \right) \leq x_e \quad e = \{i,j\} \in E \tag{10b}$$

$$z_e + p_e^k \geq (y_{ij}^{kk} + y_{ji}^{kk}) \hat{d}^k \quad e = \{i,j\} \in E, k \in K \tag{10c}$$

$$\mathbf{z}, \mathbf{p}, \mathbf{y}, \mathbf{x} \geq 0 \tag{10d}$$

$$\mathbf{x} \in \mathbb{Z}^{|E|} \tag{10e}$$

Formulation (10) can be derived from formulation (6) by removing flow conservation constraints (6b) – (6c), replacing robust non-negativity constraints (6f) – (6g) by $\mathbf{y} \geq \mathbf{0}$ and removing the terms corresponding to $\mathbf{f}^0$ and y^{kh} for $h \neq k$ in robust capacity constraints (6d) – (6e).

3   Benders formulations

Benders decomposition is a technique that projects out (part of) the continuous variables of a mixed integer linear programming problem, replacing them by a possibly exponential number of cutting planes that define the feasibility polyhedron for the variables that are not projected out. The cutting planes are usually generated on the fly in the course of branch-and-cut algorithms. It has been applied to many problems, ranging from black-out prevention [17] to shift-scheduling [35]. A standard choice in solving networks design problems consists of projecting out the flow variables working on a master problem which includes only the design variables (see [1, 6, 27, 31, 32, 33] and references therein). This corresponds to the practical decomposition of the decision process: in fact, the design variables correspond to long term decisions, whereas the flow variables correspond to decisions taken at the operational level (short term decisions). We explain in this section how to apply Benders decomposition to $fDRNL$ and $fARNL$. This is the first formal presentation of a Benders formulation for the problem with affine (and volume) routing. Benders reformulations for dynamic and static routing can be found in [27, 33]. We point out that the purposes of using Benders decomposition are different for $fDRNL$ and $fARNL$. For $fARNL$ (and its simplifications $fSRNL$ and $fVRNL$), Benders decomposition avoids solving a large linear program at each node of the branch-and-bound tree. For $fDRNL$, the mixed integer linear programming problem contains exponentially many variables and constraints (and we know from its $\mathcal{NP}$-hardness that no compact formulation exists). Hence, the use of Benders decomposition avoids to consider explicitly all vectors in $\text{vert}(\mathcal{U})$, as needed extreme points are generated on the fly by solving a separation problem.

3.1 Affine routing and simplifications

We project the flow variables out of formulation (6). Let $\mathcal{B}^{aff}$ be the projection of the set defined by constraints (6a)–(6h), formally:

$$\mathcal{B}^{aff} \equiv \{\mathbf{x} \in \mathbb{R}^{|E|} : \exists \mathbf{f}^0, \mathbf{y}, \mathbf{z}, \mathbf{s}, \mathbf{p}, \mathbf{q} \text{ that satisfy } (6a) - (6h)\}.$$

Different approaches can be used to test whether a given vector $\mathbf{x}$ belongs to $\mathcal{B}^{aff}$. In the following, we use a reformulation based on strong linear programming duality. In this end, given $\bar{\mathbf{x}} \in \mathbb{R}_+^{|E|}$, we introduce the feasibility problem associated with $\bar{\mathbf{x}}$

$$FeasAff(\bar{\mathbf{x}}) \ \min \ \alpha$$

$$\text{s.t.} \quad \sum_{j \in N(i)} (y_{ij}^k - y_{ji}^k) = \begin{cases} 1 & \text{if } i = t^k \\ 0 & \text{otherwise} \end{cases} \qquad \begin{aligned} & i \in V \setminus \{s^k\}, \\ & k \in K \end{aligned} \tag{11a}$$

$$\Gamma z_e + \sum_{k \in K} \left(p_e^k + f_{ij}^{0k} + f_{ji}^{0k} + \sum_{h \in K} (y_{ij}^{kh} + y_{ji}^{kh}) \bar{d}^h \right) \leq \bar{x}_e + \alpha \quad e = \{i,j\} \in E \tag{11b}$$

$$(6b), (6c), (6e), (6f), (6g)$$

where α represents the amount of extra capacity required to route all the demands on the network. One readily sees that $\bar{\mathbf{x}} \in \mathcal{B}^{aff}$ if and only if the optimal solution value of $FeasAff(\bar{\mathbf{x}})$ is non-positive. Let $\mathcal{D}^{aff}$ be the feasibility polyhedron of the dual of problem $FeasAff(\bar{\mathbf{x}})$, and let $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ denote the dual variables corresponding to constraints (11a) and (11b), respectively. To keep our exposition as simple as possible, we do not describe $\mathcal{D}^{aff}$ explicitly and we commit the following abuse of notation, indicating the vertices of the polyhedron by $(\boldsymbol{\pi}, \boldsymbol{\mu}) \in \text{vert}(\mathcal{D}^{aff})$. The Benders reformulation of $fARNL$ is given below.

$$bARNL \ \min \ \sum_{e \in E} c_e x_e$$

$$\text{s.t.} \quad \sum_{e \in E} \mu_e x_e \leq \sum_{k \in K} \pi_{t_k}^k \quad (\boldsymbol{\pi}, \boldsymbol{\mu}) \in \text{vert}(\mathcal{D}^{aff}) \tag{12a}$$

$$\mathbf{x} \geq 0, \mathbf{x} \in \mathbb{Z}^{|E|}$$

Problem $FeasAff(\bar{\mathbf{x}})$ is always feasible and bounded so that its optimal value is equal to the optimal value of its dual, which is obtained at an extreme point of $\mathcal{D}^{aff}$. In practice, this reformulation is addressed implicitly, by generating only the required constraints on the fly within a branch-and-cut algorithm, whose features are detailed in Section 5.2. An important property of constraints (12a) is that they can be separated in polynomial time by solving compact linear program $FeasAff(\bar{\mathbf{x}})$ or its dual. The Benders reformulations of $fVRNL$ and $fSRNL$ are obtained similarly. Namely, we define

$$\mathcal{B}^{vol} \equiv \{\mathbf{x} \in \mathbb{R}^{|E|} : \exists \mathbf{f}^0, \mathbf{y}, \mathbf{z}, \mathbf{p} \text{ that satisfy } (8a) - (8g)\}$$

$$\mathcal{B}^{stat} \equiv \{\mathbf{x} \in \mathbb{R}^{|E|} : \exists \mathbf{y}, \mathbf{z}, \mathbf{p} \text{ that satisfies } (10a) - (10d)\}$$

To check whether a given vector $\mathbf{x}$ belongs to $\mathcal{B}^{stat}$ or $\mathcal{B}^{vol}$, we can introduce feasibility problems $FeasStat(\bar{\mathbf{x}})$ and $FeasVol(\bar{\mathbf{x}})$ as before. We omit the formulations of these problems since they are obtained from $FeasAff(\bar{\mathbf{x}})$ in the same way $fSRNL$ and $fVRNL$ are obtained from $fARNL$. We denote the feasibility polyhedra of the duals of $FeasStat(\bar{\mathbf{x}})$ and $FeasVol(\bar{\mathbf{x}})$ by $\mathcal{D}^{stat}$ and $\mathcal{D}^{vol}$, respectively. The Benders reformulations $bVRNL$ and $bSRNL$ for the problem with volume and static routing are obtained from $bARNL$ by replacing $\text{vert}(\mathcal{D}^{aff})$ by $\text{vert}(\mathcal{D}^{vol})$ and $\text{vert}(\mathcal{D}^{stat})$.

3.2 Dynamic routing

Let $\mathcal{B}^{dyn}$ be the projection of the set defined by constraints (2a) – (2b):

$$\mathcal{B}^{dyn} \equiv \{\mathbf{x} \in \mathbb{R}^{|E|} : \exists \mathbf{f} : \mathrm{vert}(\mathcal{U}) \to \mathbb{R}_+^{|A|\times|K|} \text{ that satisfies (2a) } - \text{(2c)}\}.$$

In contrast with the other routing schemes, we can aggregate the commodities by source, as it is often done for the deterministic Network Loading Problem [6, 16]. We assume, without loss of generality, that K contains one commodity for each pair of nodes i, j in V, possibly with $d_{ij} = 0$. Our assumption implies that, for each $u \neq v \in V$, we can denote by $k(u, v)$ the commodity h in K such that $s^h = u$ and $t^h = v$. Each commodity in the new set can be identified by its source node u and contains $|V| - 1$ sink nodes. For each $u, v \in V$, we denote by d_v^u the demand at node v for the commodity corresponding to source node u. This value is equal to $-\sum_{v \in V\setminus\{u\}} d^{k(u,v)}$, if $u = v$, or to $d^{k(u,v)}$, if $u \neq v$. For aggregated commodities, uncertainty set $\mathcal{U}_b$ becomes

$$\mathcal{U}_b^{agg} \equiv \left\{ \mathbf{d} \in \mathbb{R}^{|V|\times|V|} \;\middle|\; \exists \boldsymbol{\delta} \in [0,1]^{|K|} : \sum_{k\in K} \delta^k \leq \Gamma, \begin{array}{l} d_v^u = \bar{d}^{k(u,v)} + \delta^{k(u,v)}\hat{d}^{k(u,v)} \\ d_u^u = -\sum_{v\in V\setminus\{u\}}(\bar{d}^{k(u,v)} + \delta^{k(u,v)}\hat{d}^{k(u,v)}) \end{array}, u \neq v \in V \right\}.$$

(13)

As we did before, we denote by $\mathcal{U}^{agg}$ the aggregated version of a generic uncertainty set $\mathcal{U}$ and by $\mathcal{U}_b^{agg}$ the aggregated version of $\mathcal{U}_b$. Let $f_{ji}^u(\mathbf{d})$ be a variable representing the flow for commodity u on edge $(i, j) \in A$ when realization $\mathbf{d}$ occurs. Vector $\bar{\mathbf{x}} \in \mathcal{B}^{dyn}$ if the problem below admits a non-positive solution.

$$\max_{\mathbf{d}\in\mathcal{U}^{agg}} \min \; \alpha$$

$$\text{s.t.} \quad \sum_{j\in N(i)} (f_{ji}^u(\mathbf{d}) - f_{ij}^u(\mathbf{d})) = d_i^u \qquad u \neq i \in V \tag{14a}$$

$$\sum_{u\in V} (f_{ij}^u(\mathbf{d}) + f_{ji}^u)(\mathbf{d}) \leq \bar{x}_e + \alpha \quad e = \{i,j\} \in E \tag{14b}$$

$$\mathbf{f}, \alpha \geq 0$$

Let $\boldsymbol{\pi}$ denote the dual variables corresponding to constraints (14a) and let $\boldsymbol{\mu}$ denote the dual variables of constraints (14b). If we restrict to $\mathcal{U}_b^{agg}$ and dualize the inner minimization problem, we obtain the following bilinear program:

$$\max \; -\sum_{e\in E} \mu_e \bar{x}_e + \sum_{u\neq v\in V} (\bar{d}^{k(u,v)} + \delta^{k(u,v)}\hat{d}^{k(u,v)})\pi_v^u$$

$$\text{s.t.} \quad \sum_{k\in K} \delta^k \leq \Gamma \qquad\qquad\qquad k \in K \tag{15a}$$

$$\pi_i^u - \pi_j^u \leq \mu_{\{i,j\}} \qquad\qquad\qquad a = (i,j) \in A, u \in V \tag{15b}$$

$$\sum_{e\in E} \mu_e \leq 1 \qquad\qquad\qquad\qquad \tag{15c}$$

$$\pi_v^v = 0 \qquad\qquad\qquad\qquad v \in V \tag{15d}$$

$$\boldsymbol{\delta} \in [0,1]^{|K|} \qquad\qquad\qquad\qquad \tag{15e}$$

$$\boldsymbol{\mu}, \boldsymbol{\pi} \geq 0. \qquad\qquad\qquad\qquad \tag{15f}$$

One easily sees that the optimal solution is reached at some binary vector $\boldsymbol{\delta}$, so that we can replace (15e) by $\boldsymbol{\delta} \in \{0,1\}^{|K|}$. By representing the product $\delta^{k(u,v)}\pi_v^u$ by an auxiliary variable ρ_v^u for each $u \neq v \in V$, we obtain the problem below. The replacement can be done without introducing big-M coefficients, because any solution of (15) satisfies $\pi_v^u \leq 1$ for each $u, v \in V$.

$$FeasDyn(\bar{\mathbf{x}}) \quad \max \; -\sum_{e\in E} \mu_e \bar{x}_e + \sum_{u\neq v\in V} (\bar{d}^{k(u,v)}\pi_v^u + \hat{d}^{k(u,v)}\rho_v^u) \tag{16a}$$

9

$$\text{s.t.} \quad \rho_v^u \leq \delta^{k(u,v)} \qquad\qquad\qquad u \neq v \in V \qquad\qquad (16\text{b})$$

$$\rho_v^u \leq \pi_v^u \qquad\qquad\qquad\qquad u \neq v \in V \qquad\qquad (16\text{c})$$

$$\rho_v^u \geq \pi_v^u + \delta^{k(u,v)} - 1 \qquad\quad u \neq v \in V \qquad\qquad (16\text{d})$$

$$\sum_{k \in K} \delta^k \leq \Gamma \qquad\qquad\qquad\qquad k \in K \qquad\qquad\quad (16\text{e})$$

$$\pi_i^u - \pi_j^u \leq \mu_{\{i,j\}} \qquad\qquad a = (i,j) \in A, u \in V \qquad (16\text{f})$$

$$\sum_{e \in E} \mu_e \leq 1 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad (16\text{g})$$

$$\pi_v^v = 0 \qquad\qquad\qquad\qquad\qquad v \in V \qquad\qquad\quad (16\text{h})$$

$$\delta \in \{0,1\}^{|K|} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad (16\text{i})$$

$$\mu, \pi \geq 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad (16\text{j})$$

$$\rho \geq 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (16\text{k})$$

Let $\mathcal{D}^{dyn}$ be the polytope defined by constraints (16f), (16g) and (16j). Then, the Benders formulation of the Robust Network Loading Problem with dynamic routing is the following.

$$bDRNL \quad \min \sum_{e \in E} c_e x_e$$

$$\text{s.t.} \quad \sum_{e \in E} \mu_e x_e \geq \sum_{u \neq v \in V} d^{k(u,v)} \pi_v^u \quad (\boldsymbol{\pi}, \boldsymbol{\mu}) \in \text{vert}(\mathcal{D}^{dyn}) \qquad (17\text{a})$$

$$\mathbf{x} \geq 0, \mathbf{x} \in \mathbb{Z}^{|E|}$$

4 Polyhedral results

In this section we illustrate, from a polyhedral point of view, what happens when the routing or the flows policy changes. Differently from the previous section, here we consider a general demand polyhedron $\mathcal{U}$, four routing policies (static, volume, affine and dynamic routing) and two flows policies (splittable and unsplittable flows). Let $R = \{stat, vol, aff, dyn\}$ be the set of the considered routing schemes and let $F = \{spl, uns\}$ be the set of the flows policies. Comments about the possibility to generalize some results for dynamic routing and splittable flows to the problem with static/affine routing or unsplittable flows were already made in [33]. We integrate and formalize them in the results below. In Section 4.1 we give general results characterizing the polyhedra corresponding to the considered routing and flows policies. We also state conditions explaining when and how valid inequalities and facets of the polyhedron corresponding to one routing/flows pair are related to valid inequalities and facets of the polyhedron for another routing/flows pair. Following [2, 33], we formally prove that facets for a given problem can be derived from facets of a reduced problem corresponding to a partition of the node set, under mild assumptions. In Section 4.2 we provide examples of inequalities (non-negativity constraints and robust cutset inequalities [26, 33, 34]) being facets for all the considered polyhedra under the same assumptions. In Section 4.3 we present a new class of valid inequalities, namely, the robust 3-partition inequalities, which represents the first class of valid inequalities that can be proved to provide facets for some routing/flow policies, but not for all of them, under the same assumptions. The robust 3-partition inequalities are the robust version of the 3-partition inequalities in [2, 29].

With little abuse of notation, we define the inclusion between routing and flows policies as below.

Definition 1. *Let r and r' be two routing policies and let p, p' be two flows policies, then:*

1. *r' includes r ($r' \supseteq r$ or, equivalently, $r \subseteq r'$), if any routing that is feasible for policy r is also feasible for policy r';*

2. *p' includes p ($p' \supseteq p$ or, equivalently, $p \subseteq p'$), if any flows that are feasible for policy p are also feasible for policy p'.*

It is easy to see that Definition 1 is independent of $\mathcal{U}$. We denote by $\mathcal{B}_I^r(p)$ the convex-hull of the integer vectors $\mathbf{x}$ supporting a feasible routing for the demands when routing policy r and flows policy p are considered. Sets $\mathcal{B}_I^r(spl)$ for $r \in R$ and $\mathcal{U} = \mathcal{U}_b$, correspond to the convex-hull of the integer feasible solutions of the formulations presented in Section 3.

4.1 General results

We first state conditions that have some relevant consequences.

Theorem 2. *It holds that:*

1. $\mathcal{B}_I^r(p) \subseteq \mathcal{B}_I^{r'}(p) \subseteq \mathbb{R}_+^{|E|} \quad \forall r, r' : r \subseteq r', \forall p;$

2. $\mathcal{B}_I^r(p) \subseteq \mathcal{B}_I^r(p') \subseteq \mathbb{R}_+^{|E|} \quad \forall p, p' : p \subseteq p', \forall r;$

Proof. The proof is in two parts.

Part 1. If $r \subseteq r'$, then any routing that is feasible for policy r is also feasible for policy r', hence any solution $\mathbf{x}$ that supports policy r also supports policy r' under the same flow policy p.

Part 2. If $p \subseteq p'$, then any flows that are feasible for policy p are also feasible for policy p', hence any solution $\mathbf{x}$ that supports policy p also supports policy p' under the same routing policy r. $\qquad\square$

Indeed, Theorem 2 implies the following result.

Corollary 3. *It holds that:*

1. *if $\mathbf{a}^T \mathbf{x} \geq b$ is valid for $\mathcal{B}_I^r(p)$, then $\mathbf{a}^T \mathbf{x} \geq b$ is valid for $\mathcal{B}_I^{r'}(p')$, for any $r' \subseteq r$, $p' \subseteq p$;*

2. *if $\mathbf{a}^T \mathbf{x} \geq b$ is not valid for $\mathcal{B}_I^r(p)$, then it is not valid for $\mathcal{B}_I^{r'}(p')$, for any $r' \supseteq r$, $p' \supseteq p$.*

Proof. It follows directly from Theorem 2. $\qquad\square$

Another consequence of Theorem 2 is Corollary 4 below.

Corollary 4. *It holds that:*

1. $\mathcal{B}_I^{stat}(p) \subseteq \mathcal{B}_I^{vol}(p) \subseteq \mathcal{B}_I^{aff}(p) \subseteq \mathcal{B}_I^{dyn}(p) \subseteq \mathbb{R}_+^{|E|} \quad p \in F;$

2. $\mathcal{B}_I^r(uns) \subseteq \mathcal{B}_I^r(spl) \quad r \in R.$

Proof. The proof is in three parts.

Part 1. As explained in Section 2, volume and static routing are subclasses of affine routing. Static routing is obtained from volume routing by settings $\mathbf{f}^0 = \mathbf{0}$. Affine routing is obtained from dynamic routing by imposing to satisfy additional constraints (3). Then, $stat \subseteq vol \subseteq aff \subseteq dyn$ and the result follows from Theorem 2.

Part 2. Unsplittable flows are obtained from splittable ones by imposing additional integrality requirements on the $\mathbf{f}$ variables. Then, $uns \subseteq spl$ and again the result follows from Theorem 2. $\qquad\square$

A vector $\mathbf{a} \in \mathbb{R}_+^{|E|}$ is a metric if, for any $e = \{i, j\} \in E$, it holds that $a_e \leq \sum_{t \in P_e} a_t$, where P_e is a ij-shortest path in $G(V, E)$ according to weights $\mathbf{a}$. Denote by $R_{\mathbf{a}}(r, p)$ the optimal cost of the Robust Network Loading Problem when $\mathbf{a}$ are the objective costs, r is the routing policy and p is the flows policy. An inequality $\mathbf{a}^T \mathbf{x} \geq b$ with $\mathbf{a}$ metric and $b = R_{\mathbf{a}}(r, p)$ is often referred to as *tight metric* [6, 30, 33].

Theorem 5. *For any $r \in R$ and $p \in F$, the following holds:*

1. $\mathcal{B}_I^r(p)$ is full-dimensional;

2. if $\mathbf{a}^T\mathbf{x} \geq b$ is a valid inequality for $\mathcal{B}_I^r(p)$, then $\mathbf{a} \geq 0$;

3. if $\mathbf{a}^T\mathbf{x} \geq b$ is a facet for $\mathcal{B}_I^r(p)$, then $\mathbf{a}$ is a metric and $b = R_{\mathbf{a}}(r,p)$.

Proof. The proof is in three parts.

Part 1. Let $M > \left\lceil \max_{\mathbf{d}\in\mathcal{U}} \sum_{k\in K} d_k \right\rceil$ be a suitably large number. For every $e \in E$, let $\mathbf{x}^e \in \mathbb{Z}_+^{|E|}$ be the vector having $x_e^e = M+1$ and $x_h^e = M$ for all $h \neq e$. Let $\mathbf{x}^M \in \mathbb{Z}_+^{|E|}$ be the vector having all entries equal to M. Vectors $\mathbf{x}^M$ and $\mathbf{x}^e$ for $e \in E$ are $|E|+1$ affinely independent vectors of $\mathcal{B}_I^r(p)$ for any $r \in R$ and $p \in F$.

Part 2. Assume that $\mathbf{a}^T\mathbf{x} \geq b$ is a valid for $\mathcal{B}_I^r(p)$ and that there exists e such that $a_e < 0$. Let $\bar{\mathbf{x}}$ be a feasible solution. Consider the solution $\mathbf{x}^M$ having $x_h^M = \bar{x}_h$ for $h \in E \setminus \{e\}$ and $x_e^M = \bar{x}_e + M$. Vector $\mathbf{x}^M$ is feasible but, for M sufficiently large, $\mathbf{a}^T\mathbf{x}^M < b$.

Part 3. Assume that $\mathbf{a}^T\mathbf{x} \geq b$ is a facet and that $\mathbf{a}$ is not a metric. That is, there exist an edge e and a path P_e between the endpoints of e such that $\sum_{h\in P_e} a_h < a_e$. Let $\boldsymbol{\mu} \in \mathbb{R}_+^{|E|}$ be the metric having $\mu_h = a_h$ for $h \in E \setminus \{e\}$ and $\mu_e = \sum_{h\in P_e} a_h$. If $\boldsymbol{\mu}^T\mathbf{x} \geq b$ is valid, then $\mathbf{a}^T\mathbf{x} \geq b$ is not a facet, because it is dominated by $\boldsymbol{\mu}^T\mathbf{x} \geq b$, as $\boldsymbol{\mu} < \mathbf{a}$ by construction. Suppose that $\boldsymbol{\mu}^T\mathbf{x} \geq b$ is not valid and let $\mathbf{w}$ be a feasible solution such that $\boldsymbol{\mu}^T\mathbf{w} < b$. Let $\bar{\mathbf{w}}$ be the feasible vector having $\bar{w}_e = 0$, $\bar{w}_h = w_e + w_h$ for $h \in P_e$, $\bar{w}_h = w_h$ otherwise. Then, $\mathbf{a}^T\bar{\mathbf{w}} = \boldsymbol{\mu}^T\mathbf{w} < b$ and, hence, $\mathbf{a}^T\mathbf{x} \geq b$ is not valid too. It follows that $\boldsymbol{\mu}^T\mathbf{x} \geq b$ must be valid and, therefore, $\mathbf{a}^T\mathbf{x} \geq b$ is not a facet. Any inequality with $b > R_{\mathbf{a}}(r,p)$ is trivially not valid. Suppose now that $b < R_{\mathbf{a}}(r,p)$. If so, it is easy to see that inequality $\mathbf{a}^T\mathbf{x} \geq b$ is dominated by $\mathbf{a}^T\mathbf{x} \geq R_{\mathbf{a}}(r,p)$. $\square$

We now establish relations among the facets of the polyhedra corresponding to different routing schemes.

Theorem 6. *For any $r \in R$, $p \in F$, the following holds:*

1. *if $\mathbf{a}^T\mathbf{x} \geq b$ is not a facet of $\mathcal{B}_I^r(p)$, then it is not a facet of $\mathcal{B}_I^{r'}(p')$, for any $r' \subseteq r$, $p' \subseteq p$;*

2. *if $\mathbf{a}^T\mathbf{x} \geq b$ is a facet of $\mathcal{B}_I^r(p)$ and it is valid for $\mathcal{B}_I^{r'}(p')$, then it is a facet of $\mathcal{B}_I^{r'}(p')$, for any $r' \supseteq r$, $p' \supseteq p$.*

Proof. The proof is in two parts.

Part 1. By Theorem 2, $\mathcal{B}_I^r(p) \supseteq \mathcal{B}_I^{r'}(p')$ for any $r \supseteq r'$ and $p \supseteq p'$ and by Theorem 5, for all $r \in R$ and $p \in F$ the corresponding $\mathcal{B}_I^r(p)$ is full-dimensional. If there are no $|E|$ affinely independent feasible solutions in $\mathcal{B}_I^r(p)$ satisfying the inequality with equality, then such vectors cannot be found in any $\mathcal{B}_I^{r'}(p') \subseteq \mathcal{B}_I^r(p)$.

Part 2. Using the same argument, if there exist $|E|$ affinely independent feasible solutions in $\mathcal{B}_I^r(p)$ satisfying $\mathbf{a}^T\mathbf{x} \geq b$ with equality, the same solutions also belong to $\mathcal{B}_I^{r'}(p')$ for any $r' \supseteq r, p' \supseteq p$. $\square$

Let $V_l = [S_1 : \ldots : S_l]$ be a partition of the node set and let the corresponding l-node problem be the problem obtained by shrinking each subset of the partition into a single node. We may have parallel edges and commodities. Let $lG(V_l, lE)$ be the corresponding graph. Let $E_{\{i,j\}} \subseteq E$ be the edges having one endpoint in set S_i and the other in set S_j. The parallel edges in $E_{\{i,j\}}$ can be merged into a unique edge $\{i,j\} \in lE$. The parallel commodities can be merged into a unique commodity for splittable flows, but not for the unsplittable ones. We say that a partition V_l is connected if S_i is connected for any $S_i \in V_l$. Let $l\mathcal{B}_I^r(p)$ be the convex-hull of the integer feasible solutions of the l-node problem. The following result holds.

Theorem 7. *Let $\sum_{e\in lE} a_e x_e \geq b$ be a valid inequality for $l\mathcal{B}_I^r(p)$, then:*

1. inequality $\sum_{e=\{i,j\}\in lE}\sum_{q\in E_{\{i,j\}}} a_e x_q \geq b$ is valid for $\mathcal{B}_I^{r'}(p')$, for any $l \geq 2$, $r' \supseteq r$, $p' \supseteq p$;

2. if it is a facet of $l\mathcal{B}_I^r(p)$, then $\sum_{e=\{i,j\}\in lE}\sum_{q\in E_{\{i,j\}}} a_e x_q \geq b$ is a facet of $\mathcal{B}_I^{r'}(p')$, for any $l \geq 2$, V_l connected, $r' \supseteq r$, $p' \supseteq p$.

Proof. We first prove that the results holds for $l\mathcal{B}_I^r(p)$ and $\mathcal{B}_I^r(p)$. Then, we show that it holds for any $\mathcal{B}_I^{r'}(p')$ with $r' \supseteq r$, $p' \supseteq p$.

Part 1. Suppose that $\sum_{e\in lE} a_e x_e \geq b$ is valid for $l\mathcal{B}_I^r(p)$, but $\sum_{e=\{i,j\}\in lE}\sum_{q\in E_{\{i,j\}}} a_e x_q \geq b$ is not valid for $\mathcal{B}_I^r(p)$. Let $\mathbf{w} \in \mathcal{B}_I^r(p)$ be a solution such that $\sum_{e=\{i,j\}\in lE}\sum_{q\in E_{\{i,j\}}} a_e w_q < b$. Let $\bar{\mathbf{w}} \in l\mathcal{B}_I^r(p)$ be the vector having $\bar{\mathbf{w}}_e = \sum_{q\in E_{\{i,j\}}} w_q$ for each $e = \{i,j\} \in lE$. Since $\mathbf{w}$ is feasible for the original problem, then $\bar{\mathbf{w}}$ is feasible for the l-node problem. It holds that $\sum_{e\in lE} a_e \bar{w}_e = \sum_{e=\{i,j\}\in lE}\sum_{q\in E_{\{i,j\}}} a_e w_q < b$ and, hence, $\sum_{e\in lE} a_e x_e \geq b$ is not valid for $l\mathcal{B}_I^r(p)$.

Part 2. Let $\mathbf{a}^T\mathbf{x} \geq b$ be a facet of $l\mathcal{B}_I^r(p)$. Then there exist $v^1,\ldots,v^{|lE|}$ affinely independent vectors of $l\mathcal{B}_I^r(p)$ satisfying it with equality. Let $E_I = E \setminus \cup_{\{i,j\}\in lE} E_{\{i,j\}}$ represent the edges of E having both endpoints in the same set of the partition. For every set $E_{\{i,j\}}$, choose a representative edge $t_{\{i,j\}} \in E_{\{i,j\}}$. Let M be a suitably large value. For any $e = \{i,j\} \in lE$ and $f \in E_{\{i,j\}}$ define vector $\mathbf{w}^{ef} \in \mathcal{B}_I^r(p)$ as: $w_h^{ef} = M$ for $h \in E_I$; $w_f^{ef} = v_e^e$; $w_h^{ef} = 0$ for $h \in E_{\{i,j\}} \setminus \{f\}$; $w_h^{ef} = v_{uv}^e$ for $h = t_{\{u,v\}}$, $E_{\{i,j\}} \neq E_{\{u,v\}}$; $w_h^{ef} = 0$ for $h \in E_{\{u,v\}} \setminus \{t_{\{u,v\}}\}$, $E_{\{i,j\}} \neq E_{\{u,v\}}$. Let $\bar{\mathbf{w}}$ be one of such vectors. For any $e \in E_I$, let $\bar{\mathbf{x}}^e \in \mathcal{B}_I^r(p)$ be the vector having $\bar{x}_q^e = \bar{w}_q$ for $q \neq e$ and $\bar{x}_e^e = M + 1$. Vectors $\bar{\mathbf{x}}^e$ for $e \in E_I$ and $\mathbf{w}^{ef}$ for $e = \{i,j\} \in lE, f \in E_{\{i,j\}}$ are $|E|$ affinely independent vectors of $\mathcal{B}_I^r(p)$ satisfying $\mathbf{a}^T\mathbf{x} \geq b$ with equality.

Part 3. By Corollary 3 and Theorem 6, if $\mathbf{a}^T\mathbf{x} \geq b$ is (valid) facet for $\mathcal{B}_I^r(p)$, then it is (valid) facet for any $\mathcal{B}_I^{r'}(p')$ with $r' \supseteq r$, $p' \supseteq p$. $\square$

4.2 Non-negativity constraints and robust cutset inequalities

We now provide examples of inequalities that are facets for $\mathcal{B}_I^r(p)$ for any routing policy $r \in R$ and flows policy $p \in F$. Indeed, to the best of our knowledge, these are the only examples of inequalities having this property. These inequalities are already known in the literature and their facet status for some of the considered polyhedra has already been established (see for example [33]). Here we provide a comprehensive proof, showing that they are facets in all the considered settings. We say that an edge is a bridge if its removal disconnects at least one origin-destination pair s_k-t_k having $d_k > 0$ for at least one $\mathbf{d} \in \mathcal{U}$.

Lemma 8. *Inequality $x_e \geq 0$ is a facet of $\mathcal{B}_I^r(p)$ for any r and p, if and only if e is not a bridge.*

Proof. If e is a bridge, no feasible solution can satisfy the inequality with equality. For any $t \in E \setminus \{e\}$ consider vector $\mathbf{x}^t$ having $x_e^t = 0$, $x_t^t = M + 1$ and $x_h^t = M$ otherwise. For M suitably large, since e is not a bridge, $\mathbf{x}^t$ is feasible and satisfies the inequality with equality. The same holds for vector $\mathbf{x}^M$ having $x_e^M = 0$ and $x_h^M = M$ for all $h \neq e$. Moreover, $\mathbf{x}^t$ for $t \in E \setminus \{e\}$ and $\mathbf{x}^M$ are $|E|$ affinely independent vectors of $\mathcal{B}_I^r(p)$. $\square$

Consider a partition of V given by sets S_1 and S_2, and let $E(S_1, S_2)$ and $K(S_1, S_2)$ be the set of the edges and the commodities with endpoints in different sets of the partition. The robust cutset inequality associated with partition $V_2 = [S_1 : S_2]$ states that the amount of capacity installed on edges in $E(S_1, S_2)$ should not be less than the rounded up sum of the maximum demands of the commodities in $K(S_1, S_2)$. Formally,

$$\sum_{e\in E(S_1,S_2)} x_e \geq \left\lceil \max_{d\in\mathcal{U}} \sum_{k\in K(S_1,S_2)} d^k \right\rceil. \tag{18}$$

It is easy to see that inequalities (18) are valid, as they correspond to necessary conditions [34]. Here we give a proof that they are facets, based on Theorem 7, that is, we first prove that they are facet-defining for a 2-node problem and then we use Theorem 7 to extend the result to the original problem. Let $V_2 = [S, V \setminus S]$ be a partition of the node set. Denote by $2G(\{u_1, u_2\}, \{\{u_1, u_2\}\})$ be the graph corresponding to the 2-node problem related to partition V_2 and let $2\mathcal{B}_I^r(p)$ be the associated polyhedron.

Theorem 9. *Robust cutset inequality (18) is facet defining for $2\mathcal{B}_I^r(p)$, for any $r \in R, p \in F$ and $\mathcal{U}$, if and only if* $\left\lceil \max_{d \in \mathcal{U}} \sum_{k \in K(S, V \setminus S)} d^k \right\rceil > 0$.

Proof. Consider the unique non-trivial cut $[\{u_1\} : \{u_2\}]$ of the 2-node problem and let D_2 be the right-hand-side of the corresponding robust cutset inequality. If $D_2 = 0$, then the inequality is dominated by the non-negativity constraints. The 2-node problem has one edge $\{u_1, u_2\}$ and possibly multiple commodities. To prove that an inequality is a facet it is sufficient to provide a not identically zero vector satisfying it with equality. The vector is $x_{u_1 u_2} = D_2$. □

Corollary 10. *If V_2 is connected, the robust cutset inequality (18) is facet defining for $\mathcal{B}_I^r(p)$, for any $r \in R, p \in F$ and $\mathcal{U}$.*

Proof. It follows from Theorems 7 and 9. □

4.3 The robust 3-partition inequalities

In this section we present a new class of valid inequalities and illustrate their polyhedral properties. They can be proved to be facets for some routing/flows policies, but not for all of them under the same assumptions. Indeed, they are the first example of valid inequalities with this property. Consider a 3-partition of the nodes $V_3 = [S_1 : S_2 : S_3]$ and let $\bar{S}_i = V \setminus S_i$ for $i = 1, 2, 3$. The robust 3-partition inequality associated with the partition is obtained by summing the three robust cutset inequalities associated with partitions $V_2 = [S_i : \bar{S}_i]$ for $i = 1, 2, 3$, dividing by two each side of the resulting inequality and rounding up its right-hand-side. Namely, let rhs_i be the right-hand-side of inequalities (18) with $V_2 = [S_i : \bar{S}_i]$ for $i = 1, 2, 3$, and let $E(S_1, S_2, S_3)$ be the set of edges with endpoints in different sets of the partition, the robust 3-partition inequality is

$$\sum_{e \in E(S_1, S_2, S_3)} x_e \geq \left\lceil \frac{1}{2}(rhs_1 + rhs_2 + rhs_3) \right\rceil. \tag{19}$$

In the following we prove under which conditions (19) is facet-defining for the problem with dynamic routing and splittable flows. We also provide examples showing that the result does not extend to the other routing/flows policies considered in this paper.

Theorem 11. *Inequality (19) is facet-defining for $3\mathcal{B}_I^{dyn}(spl)$ if and only if the following conditions are satisfied:*

1. *$rhs_1 + rhs_2 + rhs_3$ is odd;*

2. *$rhs_i > 0$ for each $i = 1, \dots, 3$;*

3. *$q = \left\lceil \frac{rhs_1 + rhs_2 + rhs_3}{2} \right\rceil > rhs_i$ for each $i = 1, \dots, 3$.*

Proof. We first prove the necessity and then the sufficiency.

Necessity. Suppose that condition 1 is not satisfied. Then, the robust 3-partition inequality is dominated by the sum of the robust cutset inequalities. Therefore it cannot be facet-defining. Suppose that condition 2 is not satisfied. Then, there exists a cutset inequality i such that $rhs_i = 0$. Let us suppose that $i = 1$. It follows that $rhs_2 = rhs_3$ and hence $rhs_1 + rhs_2 + rhs_3$ is not odd. Thus, by the proof of condition 1, the robust 3-partition inequality cannot be facet-defining. Suppose that

	x_{12}	x_{13}	x_{23}
	$q - rhs_3$	rhs_3	0
case (i)	rhs_1	0	$q - rhs_1$
	0	$q - rhs_2$	rhs_2
	$q - rhs_3$	rhs_3	0
case (ii)	rhs_1	0	$q - rhs_1$
	$rhs_1 + rhs_2 - q$	$q - rhs_2$	$q - rhs_1$
	$q - rhs_3$	rhs_3	0
case (iii)	$q - rhs_3$	$rhs_1 + rhs_3 - q$	$q - rhs_1$
	$rhs_1 + rhs_2 - q$	$q - rhs_2$	$q - rhs_1$
	$q - rhs_3$	$q - rhs_2$	$rhs_2 + rhs_3 - q$
case (iv)	$q - rhs_3$	$rhs_1 + rhs_3 - q$	$q - rhs_1$
	$rhs_1 + rhs_2 - q$	$q - rhs_2$	$q - rhs_1$

Table 1: Affinely independent vectors for Theorem 11.

condition 3 is not satisfied. Then, there exists robust cutset inequality i such that $rhs_i \geq q$. Hence, the robust 3-partition inequality is dominated by the robust cutset inequality i.

Sufficiency. We suppose without loss of generality that $rhs_1 \geq rhs_2 \geq rhs_3$. According to the values of q and rhs_i for $i = 1, \ldots 3$, we distinguish four case: (i) $q \geq rhs_1 + rhs_2 \geq rhs_1 + rhs_3 \geq rhs_2 + rhs_3$; (ii) $rhs_1 + rhs_2 > q \geq rhs_1 + rhs_3 \geq rhs_2 + rhs_3$; (iii) $rhs_1 + rhs_2 \geq rhs_1 + rhs_3 > q \geq rhs_2 + rhs_3$; (iv) $rhs_1 + rhs_2 \geq rhs_1 + rhs_3 \geq rhs_2 + rhs_3 > q$. For each case we provide in Table 4.3 three affinely independent vectors that are feasible for the problem and satisfy the robust 3-partition inequality with equality. Since the dynamic problem on 3 nodes has the robust cut property [34], the feasibility of a solution can be tested checking the robust cutset inequalities only. It is easy to see that for each case the listed vectors satisfy the robust cutset inequalities. Moreover, one readily verifies that Conditions 1–3 imply that the four 3×3 matrices from Table 4.3 are non-singular. $\square$

Therefore, we can prove what follows.

Corollary 12. *Inequalities* (19) *corresponding to connected partitions* $[S_1 : S_2 : S_3]$ *that satisfy the conditions of Theorem 11 are facet-defining for* $\mathcal{B}_I^{dyn}(spl)$.

Proof. The result follows from Theorems 7 and 11. $\square$

We provide next an example of robust 3-partition inequality that satisfies the conditions of Theorem 11, but it is not facet-defining for static routing, not even for the budgeted uncertainty set.

Example 13. *Consider a complete undirected graph with 3 nodes and let* $\mathcal{U} = \mathcal{U}_b$. *Let the nominal demands be* $\bar{d}^{12} = \bar{d}^{13} = 1$, $\bar{d}^{23} = 0$, *the deviations be* $\hat{d}^{12} = \hat{d}^{13} = 1$, $\hat{d}^{23} = 0$, *and let* $\Gamma = 1$. *The robust cutset inequalities and the corresponding robust 3-partition inequality are reported below.*

$$c_1 : \quad x_{12} + x_{13} \geq 3$$
$$c_2 : \quad x_{12} + x_{23} \geq 2$$
$$c_3 : \quad x_{13} + x_{23} \geq 2$$
$$3p : \quad x_{12} + x_{13} + x_{23} \geq 4$$

The conditions of Theorem 11 are satisfied, but it is not possible to find three affinely independent integer vectors in $3\mathcal{B}_I^{stat}(spl)$ *that satisfy* $3p$ *with equality. In fact, the only capacity allocation* $\mathbf{x} \in 3\mathcal{B}_I^{stat}(spl)$ *that satisfies* $3p$ *with equality is* $x_{12} = x_{13} = 2$, $x_{23} = 0$.

One can construct similar examples for $3\mathcal{B}_I^{vol}(spl)$ and $3\mathcal{B}_I^{aff}(spl)$, using the property that affine routing reduces to static routing whenever the uncertainty set contains the corner of $\mathbb{R}_+^{|K|}$ [44]. Example 14 shows that 3-partitions are facet for $\mathcal{B}_I^{dyn}(spl)$, but not for $\mathcal{B}_I^{dyn}(uns)$.

Example 14. *Consider the problem in Example 13 and assume that the routing is dynamic, but the flows are unsplittable. The uncertainty set includes traffic matrices A and B, where $d_{12}^A = 2$, $d_{13}^A = 1$, $d_{12}^B = 1$, $d_{13}^B = 2$, $d_{23}^A = d_{23}^B = 0$. The only solution of value 4 that allows an unsplittable routing for both A and B is $x_{12} = x_{13} = 2$, $x_{23} = 0$.*

By Theorem 6, if the robust 3-partitions are not facets for the dynamic problem, they cannot be facets for the static, affine or volume problem under unplittable flows. Indeed, the 3-partitions are not even facets for the deterministic problem, when the flows are unsplittable.

Example 15. *Consider a complete undirected graph with 3 nodes and demands $d_{12} = d_{23} = 0.4, d_{13} = 0.7$. Cutset and three-partition inequalities are:*

$$
\begin{aligned}
c_1: & \quad x_{12} + x_{13} \geq 2 \\
c_2: & \quad x_{12} + x_{23} \geq 1 \\
c_3: & \quad x_{13} + x_{23} \geq 2 \\
3p: & \quad x_{12} + x_{13} + x_{23} \geq 3
\end{aligned}
$$

The only unsplittable solution of value 3 is $x_{12} = x_{13} = x_{23} = 1$, then 3p is not a facet.

We remark again the all the results in Section 4 are independent of the uncertainty set $\mathcal{U}$. Moreover, any inequality that is valid for the capacity formulation is also valid for the flow formulation of the corresponding problem.

5 Test-bed and implementation

Now we consider the Benders formulations from a computational perspective. We restrict to splittable flows and to $\mathcal{U}_b$. The purpose of the computational experiments presented in this section is two-fold. First, we test the solvability of the Robust Network Loading Problem using different formulations and settings, on real-life instances. The scope is to show which are the approaches that can be used on problems coming from real applications and which are the approaches that are computationally problematic on such instances. Second, we compare the optimal solution values on both realistic instances and randomly generated ones. Let $\mathrm{opt}(r)$ denote the optimal solution of the problem when routing policy r is considered. Then, we see immediately that the following ordering holds:

$$\mathrm{opt}(dyn) \leq \mathrm{opt}(aff) \leq \mathrm{opt}(vol) \leq \mathrm{opt}(stat).$$

Our purpose is to show the trade-off between the value of the produced solution (flexibility) and the required time.

5.1 The test-bed

Our test bed contains two classes of instances, that we denote by *realistic* and *random*. The first class of instances includes 12 realistic network instances available from SNDlib [39]. Networks *abilene*, *germany*, and *geant* come from [26], where the authors build nominal demand values and deviations according to historical traffic data. Notice that our instances may differ slightly from the instances from [26], because we kept only commodities whose nominal demand value was greater than 0.001, to avoid numerical issues. For the other seven networks, we define the nominal demand value as the deterministic one and let the deviation be 50% of the nominal demand, as in [8, 44]. The purpose of the realistic instances is to test the performances of the different approaches on real-life instances. The second class of instances includes 12 randomly generated instances. Networks are randomly generated so that each node has at least two incoming edges to avoid trivial problems. Demands are computed generating random numbers is $[1, 9]$ and then dividing them by 10. Edge costs are randomly generated integer numbers in $\{1, \ldots, 5\}$. The purpose of the random instances is to provide instances that can be solved to optimality by all the approaches to compare the corresponding costs, without being possibly biased by the unsolved instances. Hence, they are much easier to solve than the realistic instances, with minimal differences in the computing times depending on the routing

policy.

We choose the value of Γ according to the probabilistic bound introduced in [15]. Namely, we set four levels of guaranteed probabilistic bound (denoted ϵ): $0.25, 0.10, 0.05$, and 0.01. Then, for each value of ϵ, the corresponding Γ^ϵ is such that all feasible solutions for the dynamic and the affine problem satisfy the following property: if demands are symmetric and independent random variables distributed in $[\bar{d} - \hat{d}, \bar{d} + \hat{d}]$, then, for each $a \in A$, the probability that the flow exceeds the capacity installed on arc a is less than ϵ. In this way we obtain 48 instances for the realistic instances and 48 instances for the random instances. A description of the realistic instances is reported in Table 2, while Table 3 refers to the random instances.

name	$|V|$	$|E|$	$|K|$	$\Gamma^{0.25}$	$\Gamma^{0.10}$	$\Gamma^{0.05}$	$\Gamma^{0.01}$
abilene1	12	15	66	6	11	14	19
abilene2	12	15	65	6	11	14	19
germany17	17	26	106	7	14	18	24
geant1	22	36	181	10	18	23	32
geant2	22	36	170	9	17	22	31
di-yuan	11	42	22	4	7	8	11
pdh	11	34	24	4	7	9	12
polska	12	18	66	6	11	14	19
nobel-us	14	21	91	7	13	16	23
atlanta	15	22	105	7	14	17	24
newyork	16	49	120	8	15	19	26
france	25	45	300	12	23	29	41

Table 2: Realistic instances description

name	$|V|$	$|E|$	$|K|$	$\Gamma^{0.25}$	$\Gamma^{0.10}$	$\Gamma^{0.05}$	$\Gamma^{0.01}$
n10e14d14_1	10	14	14	3	6	7	10
n10e14d14_2	10	14	14	3	6	7	10
n10e14d14_3	10	14	14	3	6	7	10
n10e14d19_1	10	14	19	4	6	8	11
n10e14d19_2	10	14	19	4	6	8	11
n10e14d19_3	10	14	19	4	6	8	11
n10e19d14_1	10	19	14	3	6	7	10
n10e19d14_2	10	19	14	3	6	7	10
n10e19d14_3	10	19	14	3	6	7	10
n10e19d19_1	10	19	19	4	6	8	11
n10e19d19_2	10	19	19	4	6	8	11
n10e19d19_3	10	19	19	4	6	8	11

Table 3: Random instances description

5.2 Implementation

The solution approaches have been coded in JAVA using Cplex Concert Technology using default cut generation parameters. All computations were run on a computer equipped with an Intel(R) Xeon(R) CPU E5540 2.53GHz processor and 16 GB of RAM, using CPLEX 12.6 [21]. We allow 7200 seconds of computing time for each instance. For the affine and the dynamic problem only the Benders formulation is considered. For the static and the volume problem, two approaches are used. The first approach solves the compact formulations $fVRNL$ and $fSRNL$, enhanced by the separation of robust cutset and robust 3-partition inequalities at the root node. The second approach addresses the problems via a Benders decomposition algorithm. The algorithm starts with a master

problem that contains one robust cutset inequality for each node of the network, then the problem is solved by branch-and-cut generating robust cutset, robust 3-partition and Benders inequalities according to the considered setting. In the following, we refer shortly to these two approaches as Compact and Benders, respectively. Then, the Benders inequalities are generated at each integer solution and, possibly, at the root node on any solution. We generate robust cutset and 3-partition inequalities, according to one of the following configurations:

0 : No cuts.

1 : Heuristic and exact separation of robust cutset inequalities at the root node only.

2 : 1+ heuristic separation of robust 3-partition inequalities at the root node only.

3 : Heuristic and exact separation of robust cutset inequalities at the root node and at integer solutions.

4 : 3+ heuristic separation of robust 3-partition inequalities at the root node and at integer solutions.

Notice that for the Compact formulation approach, only the first three implementations are tested. We also test whether it is better to generate Benders inequalities only at integer solutions (I) or at integer solutions and at the root node (R&I). Hence, we compare 10 approaches for Benders decomposition: R&I and I for each $m \in \{0, 1, 2, 3, 4\}$. For each of these configuration, the inequalities are separated in this order: 1) heuristic separation of robust cutset inequalities, 2) heuristic separation of robust 3-partition inequalities, 3) exact separation of robust cutset inequalities, 4) separation of Benders inequalities (only for Benders decomposition approaches). The problems are solved by a branch-and-cut algorithm, both for the Compact and for the Benders approach. That is, instead of solving the Benders master problem to integrality at every iteration as in [27], we solve the linear relaxation, as in a traditional branch-and-cut framework. This approach is known to yield much faster algorithms (see for example [24]). As soon as an inequality is found violated, the other inequalities are skipped and the linear programming relaxation is solved again. Another important characteristic of the Benders decomposition algorithm is the primal heuristic provided at the root node. Namely, the linear programming relaxation is first solved for the static problem using the compact formulation and the resulting capacities are rounded up to the nearest integer values. Although negligible, the solution time of this heuristic is included in the total solution time. We also tried, in some preliminary computations, the automatic Benders decomposition of CPLEX 12.7, but our ad-hoc approach turned out to have better performances.

Separation of robust cutset inequalities When the budgeted uncertainty set is used, the robust cutset inequalities are rewritten as:

$$\sum_{e \in E(S_1, S_2)} x_e \geq \left\lceil \sum_{k \in K(S_1, S_2)} \bar{d}^k + \max_{Q \subseteq K(S_1, S_2), |Q| \leq \Gamma} \sum_{k \in Q} \hat{d}^k \right\rceil.$$

We separate them using two approaches. The first approach separates the cut heuristically as follows. We randomly partition the nodes into two subsets and then perform 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 has been found, we choose another partition, up to a maximum of 5 iterations. The second approach separates the inequalities exactly through the following formulation.

$$\max \; -\sum_{e \in E} x_e \mu_e + \beta$$

$$\text{s.t.} \quad \mu_e \geq \max\{r_i - r_j, r_j - r_i\} \qquad e \in E \tag{20a}$$
$$\mu_e \leq \min\{r_i + r_j, 2 - r_j - r_i\} \qquad e \in E \tag{20b}$$
$$\nu_k \geq \max\{r_{s^k} - r_{t^k}, r_{t^k} - r_{s^k}\} \qquad k \in K \tag{20c}$$
$$\nu_k \leq \min\{r_{s^k} + r_{t^k}, 2 - r_{s^k} - r_{t^k}\} \qquad k \in K \tag{20d}$$

18

$$\ell_k \leq \min\{\gamma_k, \nu_k\} \qquad\qquad\qquad k \in K \qquad\qquad\qquad (20e)$$

$$\beta \leq \sum_{k \in K} \bar{d}_k \nu_k + \sum_{k \in K} \hat{d}^k \ell_k + 1 - \epsilon \qquad\qquad\qquad (20f)$$

$$\sum_{k \in K} \gamma_k = \Gamma$$

$$\boldsymbol{\gamma}, \boldsymbol{\ell} \in \{0,1\}^{|K|}, \boldsymbol{\mu} \in \{0,1\}^{|E|}, \boldsymbol{\nu} \in \{0,1\}^{|K|}$$

$$\beta \in \mathbb{Z}, \mathbf{r} \in \{0,1\}^{|V|}$$

Variable r_i is one if node i belongs to set S of the partition and zero otherwise. Variable ℓ_k and constraints (20e) represent product $\gamma_k \mu_k$. Constraints (20a)–(20d) ensure that μ_e (resp. ν_k) are equal to one if and only if the endpoints of the edge (resp. commodity) belong to different subsets of the partition. We remind that the robust cutset inequalities belong to the first Chvátal closure and, under some assumptions, they are facets of the problem for all the considered routing policies.

Separation of robust 3-partition inequalities We separate the robust 3-partition inequalities heuristically as follows. We randomly partition the nodes into three subsets and then perform a local search picking one node and moving it to another subset, until there is no more improvement in the violation. If no violated inequality has been found, we choose another partition, up to a maximum of 5 re-starts. Is is also possible to develop an exact algorithm based on a generalization of formulation (20). We have a set of variables and constraints for every cut, plus extra binary variables η and integer Θ representing the 3-partition and the right-and-side of the inequality, to be computed rounding the right-hand-sides of the robust cutset inequalities. However, a preliminary testing proved this formulation to be quite slow in practice, therefore we rely on the heuristic approach, as usually done in the literature for partition-based inequalities other than cutsets, even for the problem without uncertainty [1]. We remark that the robust 3-partition inequalities are facets under some conditions of the problem with dynamic routing, while they belong to the second Chvátal closure for the other routing schemes, as they are obtained combining the robust cutset inequalities, that belong to the first Chvátal closure.

Separation of Benders inequalities Benders inequalities are separated exactly by solving problems (11) and (16), depending on the routing scheme. After finding a violated inequality $\boldsymbol{\mu}^T \mathbf{x} \geq b$, its coefficients and right-hand-side are rounded using the following approach, originally proposed in [16]. It consists of replacing the inequality by

$$\sum_{e \in E} \left\lceil \frac{\mu_e}{\mu^{min}} \right\rceil x_e \geq \left\lceil \frac{b}{\mu^{min}} \right\rceil,$$

where μ^{min} is the smallest positive entry of $\boldsymbol{\mu}$. In the unlikely situation where the rounded cut is not violated, instead, we add the original cut. We note that one could directly separate inequalities with integer $\boldsymbol{\mu}$ and b being the upper integer of the corresponding Benders cut (also known as *rounded metric inequalities*). However, a preliminary testing proved that separating Benders inequalities in their non-rounded form and then strengthening them, is computationally more efficient than solving the integer separation problem. We note that the strengthened Benders cuts we use here belong to the first Chvátal closure, as they are obtained (heuristically) rounding a single inequality of the formulation.

6 Results

This section is organized as follows. First we discuss the information obtained about the costs corresponding to the different routing schemes using the random instances, then we analyze in detail the results obtained on the realistic instances.

name	$1-\epsilon$	opt_{stat}	Solution costs $red_{vol}(\%)$	$red_{aff}(\%)$	$red_{dyn}(\%)$	Best solution time/Best gap $stat$	vol	aff	dyn
abilene1	0.25	31	0.0	0.0	0.0	1	3	2416	8
	0.1	32	0.0	0.0	0.0	1	3	1379	13
	0.05	33	0.0	0.0	0.0	2	3	1532	5
	0.01	33	0.0	0.0	0.0	1	4	1780	3
abilene2	0.25	20	5.0	5.0	5.0	2	2	1568	16
	0.1	22	0.0	0.0	0.0	2	3	3986	7
	0.05	22	0.0	0.0	0.0	1	2	1838	3
	0.01	22	0.0	0.0	0.0	1	3	1686	3
germany17	0.25	35	2.9	≥ 0.0	2.9	65	12	**28%**	54
	0.1	36	0.0	0.0	0.0	7	20	**27%**	32
	0.05	36	0.0	0.0	0.0	8	14	**28%**	932
	0.01	36	0.0	0.0	0.0	7	10	**26%**	13
geant1	0.25	30	0.0	0.0	0.0	72	142	**M**	2551
	0.1	31	3.2	≥ 0.0	≥ 0.0	1849	138	**M**	**35%**
	0.05	31	3.2	≥ 0.0	≥ 0.0	200	142	**M**	**33%**
	0.01	31	0.0	≥ 0.0	≥ 0.0	78	65	**M**	**22%**
geant2	0.25	34	2.9	≥ 0.0	2.9	662	134	**M**	837
	0.1	34	0.0	≥ 0.0	≥ 0.0	72	53	**M**	**18%**
	0.05	34	0.0	≥ 0.0	≥ 0.0	79	126	**M**	**36%**
	0.01	34	0.0	≥ 0.0	≥ 0.0	343	93	**M**	**15%**
di-yuan	0.25	5240900	8.5	9.6	9.6	79	6	118	4
	0.1	5366800	1.9	3.6	3.7	8	119	1697	17
	0.05	5371400	1.1	2.6	2.8	6	32	5390	6
	0.01	5371400	0.0	0.0	0.0	1	2	96	1
pdh	0.25	850604.8	4.8	6.2	6.5	279	282	4202	12
	0.1	852303.1	0.1	0.0	0.6	41	149	**1%**	15
	0.05	852585.8	0.0	0.0	0.1	8	22	1558	3
	0.01	853009.8	0.0	0.0	0.0	6	10	913	3
polska	0.25	261.2	12.4	≥ 0.0	≥ 0.0	21	44	**18%**	**18%**
	0.1	287.4	12.8	≥ 9.3	≥ 0.0	31	17	**20%**	**19%**
	0.05	293.5	10.9	14.5	≥ 0.0	878	95	6337	**17%**
	0.01	295.1	5.1	8.8	≥ 0.0	101	28	6329	**12%**
nobel-us	0.25	294886.5	10.5	≥ 0.0	≥ 0.0	37	266	**18%**	**18%**
	0.1	315622.5	9.2	≥ 0.0	≥ 0.0	34	1137	**16%**	**16%**
	0.05	319814.5	7.9	≥ 0.0	≥ 0.0	113	324	**15%**	**15%**
	0.01	322963.5	4.3	≥ 0.0	≥ 0.0	499	956	**12%**	**12%**
atlanta	0.25	200105	4.7	≥ 0.0	5.4	20	163	**11%**	1902
	0.1	209610	3.4	≥ 0.0	≥ 0.0	41	74	**8%**	**8%**
	0.05	211680	2.7	≥ 0.0	≥ 0.0	130	26	**8%**	**8%**
	0.01	214480	1.7	≥ 0.0	≥ 0.0	81	27	**6%**	**6%**
newyork	0.25	985.2	0.0	0.0	0.0	49	110	**82%**	22
	0.1	985.2	0.0	0.0	0.0	47	140	**82%**	32
	0.05	985.2	0.0	0.0	0.0	51	114	**83%**	34
	0.01	985.2	0.0	0.0	0.0	38	162	**83%**	20
france	0.25	10.4	7.7	≥ 0.0	≥ 0.0	4074	2196	**M**	**20%**
	0.1	11	6.4	≥ 0.0	≥ 0.0	678	4136	**M**	**21%**
	0.05	11.2	5.4	≥ 0.0	≥ 0.0	599	3961	**M**	**73%**
	0.01	11.5	4.3	≥ 0.0	≥ 0.0	**1%**	**1%**	**M**	**19%**

Table 4: Overview of the results.

(a) Varying ϵ.

(b) Varying $|K|$.
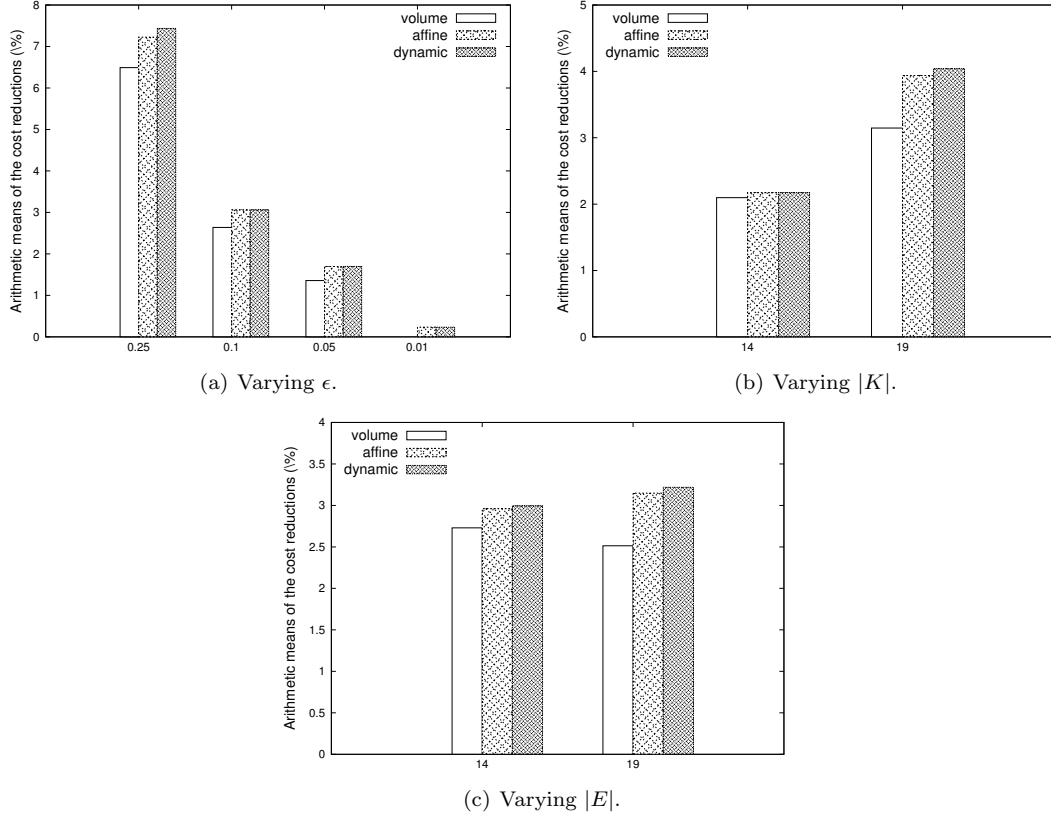
(c) Varying $|E|$.

Figure 1: Cost reductions from the static routing solution on the instances from Table 3.

Random instances The purpose of the experiments here is to compare the costs associated with the different routing policies. In order to do that, we compare the objective values obtained solving the problem corresponding to each routing policy. Hence, as said above, these instances are generally much easier to solve than the realistic instances, with no relevant differences between the routing policies. We present in Figure 1 the average cost reductions over static routing for the instances described in Table 3. Globally, the results show that the cost reductions obtained for volume and affine routing are close to those obtained for dynamic routing. In addition, Figure 1(a) highlights that decreasing the value of ϵ (i.e. increasing the value of Γ) reduces significantly all cost reductions, while Figure 1(b) illustrates how increasing the number of commodities increases all cost reductions. Regarding the number of edges of the network, Figure 1(b) shows that networks with more edges yield a slight increase in the cost reduction obtained by affine and dynamic routing while the opposite holds for volume routing.

Outline of the results on realistic instances We present first an overview of the results, paying a particular attention to the cost reductions offered by volume and dynamic routing. Then, we compare the efficiency of the different algorithms for solving each type of routing, but affine routing, where our approach could solve only one third of the instances. This step is carried out by comparing the arithmetic and the geometric means of the solution times and the number of unsolved instances. It is worth recalling that the arithmetic mean gives more weight to hard instances, while the geometric mean considers equally all instances, regardless of their difficulty. Notice also that this approach hides a part of the difficulty of the unsolved instances, since their solution times count for 7200 seconds in all computations. Hence, we report actual lower bounds for the true (unknown) means. This comment is particularly important for dynamic routing, for which some instances could not be solved. In spite of this, these aggregated results give us valuable insight for choosing the approach that seems the best for each routing. After studying each routing individually, we compare

Formulation	Compact			Benders									
Inequalities	0	1	2	I0	I1	I2	I3	I4	R&I0	R&I1	R&I2	R&I3	R&I4
Arithm mean	2467	1935	1592	2308	1858	1629	1466	1535	1941	1490	1338	1017	**930**
Geom mean	309	228	130	449	266	217	135	125	304	224	168	125	**93**
Unsolved	13	7	6	6	5	**4**	**4**	**4**	9	5	**4**	**4**	**4**

Table 5: General comparison of the approaches for static routing.

the best approaches for the different routings, study their sensitivity to the value of ϵ and study the gap closed by the valid inequalities.

6.1 Realistic instances: a global overview

We provide in Table 4 a global view of our computational results. The first and second columns describe the instance and the level of protection, respectively. Column opt_{stat} reports the optimal solution cost with static routing, while columns red_{vol}, red_{aff}, and red_{dyn} report the percentage decrease in solution costs with volume routing, affine routing, and dynamic routing, respectively. When the problems could not be solved to optimality, we report these reductions preceded by symbol $\geq$. The next columns provide insights into the computational difficulty of the optimization problems. Namely, columns *Best solution time/Best gap* provide the solution time in seconds of the best approach if the instance could be solved to optimality. Otherwise, the column reports the best gap found after 7200 seconds of computing, unless a memory limit hit happened while constructing the linear program, which is denoted by **M**.

The table shows that solution times for volume routing are not much higher than those for static routing; in some cases, they are even smaller. Recall, however, that this comparison is not rigorous because we compare different algorithms for different types of routing and instances. A better comparison is realized below, after having selected the best algorithm for each routing. Regarding the solution costs, we see that volume routing yields a positive cost reduction in 26 instances out of 48, which ranges up to 12.8 %. In most cases, the cost reduction is higher when the protection level is high (and thus, Γ is small). Notice also the limited variations in the solution costs of instances *abilene1, abilene2, germany, geant1, and geant2*, which is due to the particular cost structure of these instances ($c_e = 1$ for each $e \in E$). Dynamic routing is, as expected, harder to solve than static and volume and 21 instances could not be solved within the time limit. However, for *pdh, di-yuan,* and *newyork*, dynamic routing is easier to solve than the two others. Dynamic routing improves over volume routing by up to 1.8 additional percent (reached for *di-yuan*, $\epsilon = 0.1$). The results for affine routing are quite disappointing. This routing scheme is even harder to solve than dynamic routing, with 31 unsolved instances due either to time or memory restrictions. Not surprisingly, the difficulty of affine routing is essentially due to the large formulations since even the linear programming relaxation can be hard to solve exactly. For instance, the linear programming relaxation could not be solved for instances *geant1, geant2*, and *france* because of memory limitations. In spite of this, affine routing shows interesting cost reductions for *polska*, improving over the reduction offered by the volume routing for two cases for which the dynamic routing could not be solved close to optimality. Because of the high number of unsolved instances, we disregard affine routing in the detailed study below. One can notice that for networks *polska, nobel-us*, and *atlanta* some of the end gaps are identical for the affine and dynamic models. This is due to the fact that the best upper bound is provided by the root heuristic described in Section 5.2, while the lower bound is provided by robust cutset and 3-partition inequalities, which are identical for the two routings.

6.2 Realistic instances: a more detailed analysis

Here we analyze the performances of each routing policy, compare them, discuss the sensitivity with respect to ϵ and the performances of the considered inequalities. Finally, we report the complete tables of the results for the best approaches.

Formulation	Compact			Benders									
Inequalities	0	1	2	I0	I1	I2	I3	I4	R&I0	R&I1	R&I2	R&I3	R&I4
Arithm mean	3338	2995	2544	2590	2180	1854	1005	1029	2539	1890	1642	**961**	1003
Geom mean	972	515	311	713	380	296	109	**96**	566	378	298	123	97
Unsolved	16	15	16	6	4	2	2	**1**	7	5	3	2	3

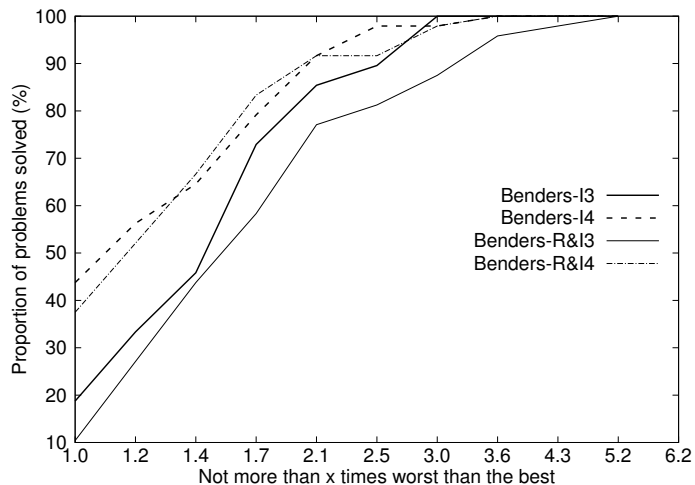Table 6: General comparison of the approaches for volume routing.

Figure 2: Performance profile comparing Benders-R&I3 and Benders-I4 for volume routing.

Static routing Table 5 presents an aggregated comparison of the solution times for the 13 approaches. Solution times of unsolved instances are set to 7200 seconds. The table shows the significant improvement offered by the robust 3-partition inequalities: Compact-2 is much faster than Compact-1 and leaves fewer instance unsolved; the inclusion of partition inequalities yields similar comments for the Benders approaches. Namely, Benders-I2 is faster than Benders-I1, Benders-I4 is faster than Benders-I3, Benders-R&I2 is faster than Benders-R&I1, and Benders-R&I4 is faster than Benders-R&I3. The table shows that Benders-R&I4 is the fastest solution algorithm. The full statistics are provided in Table 8.

Volume routing Table 6 is the counterpart of Table 5 for volume routing, presenting an aggregated comparison of the solution times for the thirteen approaches. The table shows again the improvement offered by the robust 3-partition inequalities which holds for the aforementioned pair of algorithms but Benders-R&I4 and Benders-R&I3 for which the latter is not outperformed by the former. In contrast with the situation observed in Table 5, there is no absolute winner for volume routing. Namely, Benders-R&I3 yields the best arithmetic mean, while Benders-I4 has the best geometric mean and solves more instances than the others. Benders-R&I3 and Benders-I4 (together with Benders-I3 and Benders-R&I4) are further compared through the performance profile shown in Figure 2, which confirms the modest advantage of Benders-I4 over the other approaches. The full details of Benders-I4 are provided in Table 9. Interestingly, compact formulations are less efficient for the problem with volume routing than they are for the one with static routing, which is probably due to the larger number of variables and constraints present in $fVRNL$. Nevertheless, Benders decomposition algorithms seem to perform comparably well for both routings. Analyzing tables 8 and 9, one can explain these good results by the numbers and the efficiency of the generated valid inequalities. On the one hand, Volume-Benders-I4 loses more time generating Benders cuts than Static-Benders-RI4 because the separation problem for Benders cuts is more difficult for volume routing than for static routing. On the other hand, robust 3-partition inequalities are tighter for the problem with volume routing than they are for the one with static routing, as $\mathcal{B}_I^{stat}(spl) \subseteq B^{vol}(spl)$.

23

Formulation					Benders					
Inequalities	I0	I1	I2	I3	I4	R&I0	R&I1	R&I2	R&I3	R&I4
Arithm mean	4565	4335	4125	3546	**3520**	4928	4582	4345	3893	3834
Geom mean	971	703	664	393	**349**	1765	885	739	600	524
Unsolved	29	28	26	23	**22**	30	29	27	25	24

Table 7: General comparison of the approaches for dynamic routing.
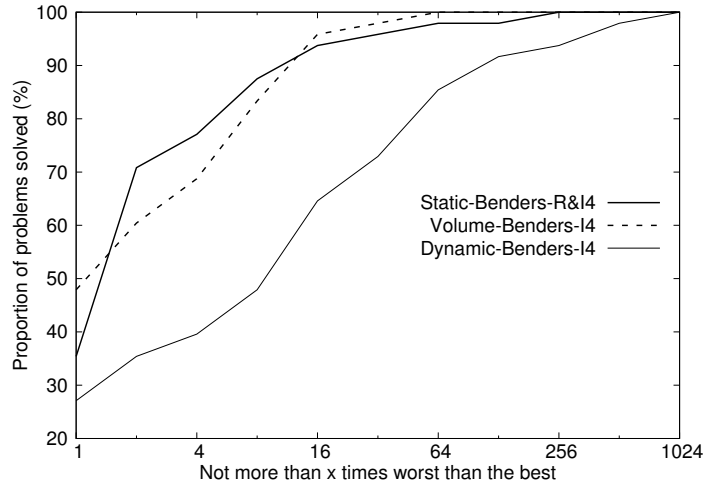

Figure 3: Performance profile comparing Static-Benders-R&I4, Volume-Benders-I4, and Dynamic-Benders-I4.

Dynamic routing   Table 7 is the counterpart of Tables 5 and 6 for dynamic routing, presenting an aggregated comparison of the solution times for the twenty approaches. The table shows again the constant improvement offered by the robust 3-partition inequalities. However, these improvements are still not enough for many of the instances and we see that each algorithm leaves many more unsolved instances than the algorithms presented for the other routings. In view of the high numbers of unsolved instances, the reported means should be taken very lightly. Still, the results seem to indicate that the winner among all approaches is Benders-I4, followed closely by Benders-I3. The full details of Benders-I4 are provided in Table 10.

Comparing the different routings and sensitivity to ϵ   We present in Figure 3 a performance profile that compares the best algorithms for the three routings. Figure 3 confirms that the efficiency of the approaches for static routing and volume routing can hardly be ordered. The plot also shows that Dynamic-Benders-I4 is usually slower than the other algorithms. We present in Figure 4 the sensitivity of the three best algorithms to the variations of ϵ. The figure shows that the geometric means of the solution times for Volume-Benders-I4 and Dynamic-Benders-I4 are not monotonically impacted by the value of ϵ. The results are different for Static-Benders-R&I4, however, for which larger values of ϵ consistently yield harder optimization problems.

Gap closed by the inequalities   We study next the effect of the robust cutset and 3-partition inequalities on the gap at the root node. For each network and static or volume routing, let $RootGap$ be the root gap obtained from the linear relaxation of $fSRNL$ or $fVRNL$, respectively, and let $CutsetGap$ and $ThreePartitionGap$ be the root gaps obtained after having separated the robust cutset and 3-partition inequalities, respectively. Then, we compute the proportion of the gap closed by the robust cutset and 3-partition inequalities as

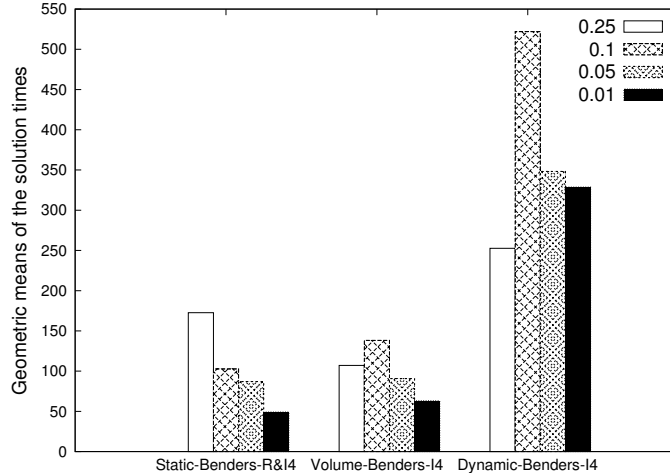$$\frac{RootGap - CutsetGap}{RootGap}, \quad \text{and} \quad \frac{CutsetGap - ThreePartitionGap}{CutsetGap},$$

Figure 4: Sensitivity of Static-Benders-R&I4, Volume-Benders-I4, and Dynamic-Benders-I4 to the variations of ϵ.

(a) *RootGap* reduction by cutset inequalities.

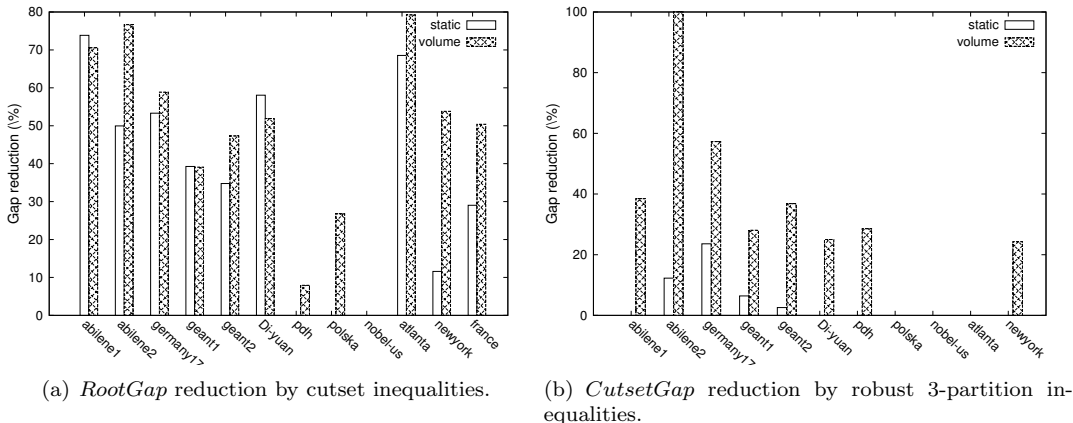(b) *CutsetGap* reduction by robust 3-partition inequalities.

Figure 5: Arithmetic means of the gap closed by the valid inequalities.

We report in Figure 5 the arithmetic means of these gaps taken over the four different values of ϵ. For Figure 5(a), we have set to 0 the negative value obtained for *abilene1* and static routing, see the explanation below. Network *france* is not included for volume routing because its optimal solution is unknown. Two important conclusions can be drawn from the figures. First, the robust 3-partition inequalities succeed in closing a large part of the *CutsetGap*, confirming again their efficiency. Second, gap reductions are almost always more significant for volume routing than for static routing, which was expected because of Theorem 2. We also mention that cuts are heuristically separated through randomly generated partitions and this may also affect the results, because different cuts may be generated for the different approaches. A similar argument explains the presence of the negative number set to zero for Figure 5(a): the robust cutset inequalities generated when computing the *CutsetGap* and the *ThreePartitionGap* may not have been the same. Unfortunately, we cannot provide similar results for dynamic routing, for which no compact formulation is available.

Detailed results   Tables 8–10 report the full results for the three most efficient algorithms. Columns *TTime*, *CTime*, *3PTime*, and *BTime* report the total solution times and the time to generate, respectively, robust cutset inequalities, robust 3-partition inequalities, and Benders inequalities. Columns *CCuts*, *3PCuts*, and *BCuts* report the number of cutting planes generated, respectively, robust cutset inequalities, robust 3-partition inequalities, and Benders inequalities; *C3PICuts* further reports

25

the number of robust cutset and robust 3-partition inequalities generated at integer nodes. Column *endGap* provides the gap at the of the algorithm (equal to 0 when solved to optimality) and column *nodes* provides the number of nodes searched along the branch-and-bound algorithm. We see that separating robust 3-partition inequalities can be done in a negligible amount of time. Robust cutset inequalities take more time to separate. For all Benders decomposition algorithms, we see that separating Benders cuts takes large amounts of time. For static and volume routings, *CTime* and *BTime* consume both large and comparable proportions of the total computational times, *BTime* being larger, on average. The situation is different for dynamic routing since Table 10 shows that almost all computational time is spent in the separation of the Benders inequalities.

7   Conclusion

The main scope of the paper is to study the Robust Network Loading Problem. We show what happens from the theoretical point of view when the flows or the routing policy changes. We also provide computational results on the capacity formulation of the problem with splittable flows and the budgeted uncertainty set illustrating what happens, from a computational perspective, when different routing policies are considered. Results for the compact formulations for the static, affine and volume problem are discussed, as well. It is the first time that such a comprehensive study is accomplished.

The theoretical study allowed to characterize the polyhedra corresponding to the capacity formulations of all the considered routing schemes, under two different flows policies, namely splittable and unsplittable flows. We also underlined the relations that exist among the polyhedra corresponding to different routing/flows policies. We discussed two classes of inequalities (non-negativity constraints and robust cutset inequalities) that can be proved to be facet defining under the same assumptions in all the considered settings. So far, these are the only examples of inequalities having this property. Then, we introduced a new class of valid inequalities, the robust 3-partition inequalities. We proved that they are facets for the problem with dynamic routing and splittable flows, but they are not facets, under the same assumption, for the problem with unsplittable flows or for other routing schemes. This is the first example of inequalities with this property.

The computational study shed light on the following issues. First, affine routing is hardly tractable, even using decomposition algorithms, due to the large linear programming formulation. Second, volume routing, obtained from affine routing by keeping only two non-zero coefficients for each affine function, behaves extremely well. Namely, our results suggest that volume routing yields cost reductions close to those obtained using dynamic routing but requires computational times similar to those obtained for static routing. Volume routing seems to offer the best trade-off between flexibility and tractability, while requiring as little information as static routing when it comes to decentralized implementations. While for static routing compact formulations can be as fast as Benders decomposition algorithms, the situation is different for volume routing for which Benders decomposition clearly outperforms the compact formulation. Third, we show that dynamic routing can be used to solve many real-life instances and, although some instances cannot be solved to optimality in the considered time limit, it performs better than affine routing. Finally, we confirm the efficiency of robust cutset inequalities and show that the generalization of the 3-partition inequalities to the robust context further reduces root gaps and computational times.

References

[1] Y. Agarwal. Design of survivable networks using three- and four-partition facets. *Operations Research*, 61(1):199–213, 2015.

[2] Y. K. Agarwal. *k*-partition-based facets of the network design problem. *Networks*, 47(3):123–139, 2006.

name	ϵ	TTime	CTime	3PTime	BTime	CCuts	3PCuts	C3PICuts	BCuts	nodes	endGap
abilene1	0.25	1	0	0	0	14	2	9	0	0	0
	0.1	2	2	0	0	22	2	12	0	5	0
	0.05	2	1	0	0	24	0	14	0	2	0
	0.01	1	0	0	0	18	0	10	0	0	0
abilene2	0.25	6	5	0	1	19	9	7	14	49	0
	0.1	3	2	0	0	21	2	7	1	8	0
	0.05	3	3	0	0	23	2	7	4	11	0
	0.01	1	1	0	0	23	3	11	1	0	0
germany17	0.25	54	17	0	36	91	5	70	37	3088	0
	0.1	13	9	0	3	74	32	11	3	97	0
	0.05	22	10	0	10	67	36	11	8	143	0
	0.01	16	8	0	7	61	13	7	5	43	0
geant1	0.25	140	45	0	90	268	83	182	17	5873	0
	0.1	1794	211	0	1566	382	90	199	394	14506	0
	0.05	245	65	0	176	281	64	140	40	2767	0
	0.01	85	22	0	59	337	25	323	6	5110	0
geant2	0.25	776	130	0	637	382	94	191	172	6915	0
	0.1	56	40	0	14	185	71	98	1	1222	0
	0.05	135	46	0	86	262	70	159	19	1488	0
	0.01	323	80	0	238	356	80	210	67	3897	0
di-yuan	0.25	237	16	0	7	30	10	10	138	567769	0
	0.1	19	9	0	6	34	14	3	53	8308	0
	0.05	14	7	0	4	35	6	7	38	8015	0
	0.01	1	0	0	0	31	10	5	0	25	0
pdh	0.25	3001	186	0	60	49	11	12	573	3570973	0
	0.1	81	51	0	21	52	21	13	129	18667	0
	0.05	18	13	0	3	76	29	33	18	1977	0
	0.01	5	4	0	0	61	57	18	0	1143	0
polska	0.25	1069	275	0	65	48	6	6	973	1115041	0
	0.1	1164	287	0	104	56	6	60	1012	998910	0
	0.05	1133	106	0	38	48	6	7	354	2577564	0
	0.01	1193	39	0	34	57	2	51	126	3878157	0
nobel-us	0.25	466	238	0	90	83	10	4	639	210558	0
	0.1	602	267	0	121	71	15	3	651	317020	0
	0.05	778	279	0	175	71	12	2	696	519640	0
	0.01	768	166	0	187	74	9	2	365	815563	0
atlanta	0.25	156	80	0	57	47	10	6	234	39653	0
	0.1	397	156	0	200	46	9	5	398	73843	0
	0.05	572	210	0	297	39	15	10	480	116448	0
	0.01	269	101	0	154	40	6	10	272	22357	0
newyork	0.25	50	10	0	19	262	72	247	0	35205	0
	0.1	57	13	0	26	312	69	294	0	32794	0
	0.05	47	12	0	17	321	69	306	0	32407	0
	0.01	50	14	0	18	263	61	242	0	34166	0
france	0.25	TL	6647	0	554	170	23	2	47	0	14.7
	0.1	TL	652	0	6530	148	35	2	611	0	14.4
	0.05	TL	TL	0	41	122	5	4	3	0	19.6
	0.01	TL	722	0	6417	129	27	3	609	0	12.7
Arithmetic mean		930	384	0	379	118	27	63	192	313363	1

Table 8: Details of Benders-R&4 for static routing.

[3] A. Altın, E. Amaldi, P. Belotti, and M. Pınar. Provisioning virtual private networks under traffic uncertainty. *Networks*, 49(1):100–115, 2007.

[4] A. Altin, H. Yaman, and M. c. Pinar. The robust network loading problem under hose demand uncertainty: Formulation, polyhedral analysis, and computations. *INFORMS Journal on*

name	ϵ	TTime	CTime	3PTime	BTime	CCuts	3PCuts	C3PICuts	BCuts	nodes	endGap
abilene1	0.25	3	1	0	1	18	3	9	0	4	0
	0.1	3	1	0	2	21	3	16	0	2	0
	0.05	5	2	0	3	26	7	11	0	14	0
	0.01	3	1	0	2	22	6	10	0	2	0
abilene2	0.25	4	2	0	1	24	13	6	0	5	0
	0.1	4	2	0	2	21	4	6	0	8	0
	0.05	3	1	0	1	19	4	12	0	7	0
	0.01	5	2	0	2	20	4	9	0	4	0
germany17	0.25	11	4	0	6	70	17	8	0	61	0
	0.1	17	7	0	9	70	30	16	0	103	0
	0.05	50	6	0	43	84	2	75	0	789	0
	0.01	9	2	0	6	59	11	5	0	15	0
geant1	0.25	86	32	0	50	251	87	163	0	4557	0
	0.1	73	36	0	34	148	74	47	0	1197	0
	0.05	113	28	0	83	186	64	87	0	1097	0
	0.01	133	15	0	112	328	23	327	0	8209	0
geant2	0.25	179	26	0	151	182	57	68	0	818	0
	0.1	141	27	0	111	227	64	122	1	1451	0
	0.05	138	30	0	103	184	52	95	1	2089	0
	0.01	150	41	0	107	169	91	70	2	615	0
di-yuan	0.25	39	6	0	33	21	7	8	38	756	0
	0.1	272	23	0	156	37	10	12	178	214309	0
	0.05	86	9	0	42	29	10	11	54	103682	0
	0.01	2	0	0	1	23	3	10	1	6	0
pdh	0.25	472	105	0	232	41	21	12	312	218917	0
	0.1	195	64	0	118	46	25	17	171	25674	0
	0.05	26	10	0	15	75	23	27	14	1911	0
	0.01	16	6	0	8	74	45	41	0	2681	0
polska	0.25	73	30	0	40	31	5	5	81	7568	0
	0.1	118	47	0	67	41	7	7	147	13347	0
	0.05	291	80	0	183	40	6	6	264	74134	0
	0.01	328	101	0	200	38	6	5	335	62788	0
nobel-us	0.25	TL	330	0	2068	102	4	97	880	5294609	0
	0.1	4656	359	0	1965	62	7	2	918	3307859	0
	0.05	4551	454	0	2503	68	6	3	1150	2010349	0
	0.01	3189	355	0	2043	56	4	6	830	1234300	0
atlanta	0.25	255	28	0	226	57	1	54	67	3626	0
	0.1	130	24	0	106	44	2	15	38	789	0
	0.05	34	12	0	22	38	3	10	12	28	0
	0.01	80	14	0	65	44	0	13	26	168	0
newyork	0.25	190	15	0	162	335	86	328	0	22411	0
	0.1	154	17	0	119	361	91	348	0	28640	0
	0.05	171	13	0	145	310	60	296	0	24778	0
	0.01	157	16	0	129	320	79	302	0	22040	0
france	0.25	2725	209	0	2512	143	15	4	63	211	0
	0.1	TL	TL	0	83	114	2	4	2	0	19.9
	0.05	TL	316	0	6881	138	20	3	174	0	19.9
	0.01	TL	312	0	6887	123	27	2	167	0	16.5
Arithmetic mean		1003	215	0	580	103	25	59	123	264513	1

Table 9: Details of Benders-I4 for volume routing.

Computing, 23(1):75–89, 2011.

[5] R. Andrade, A. Lisser, and N. Maculan. Multi-service multi-facility network design under uncertainty. *Annals of Operations Research*, 199(1):157–178, 2012.

[6] P. Avella, S. Mattia, and A. Sassano. Metric inequalities and the network loading problem.

name	ϵ	TTime	CTime	3PTime	BTime	CCuts	3PCuts	C3PICuts	BCuts	nodes	endGap
abilene1	0.25	8	0	0	7	51	6	17	0	3	0
	0.1	21	3	0	18	55	4	21	0	8	0
	0.05	5	4	0	1	46	8	17	0	3	0
	0.01	5	3	0	1	41	10	11	0	7	0
abilene2	0.25	39	2	0	37	49	10	17	0	3	0
	0.1	117	2	0	115	56	8	10	0	6	0
	0.05	3	2	0	0	50	3	9	0	3	0
	0.01	3	2	0	0	53	1	18	0	9	0
germany17	0.25	54	15	0	38	144	17	20	0	155	0
	0.1	124	14	0	110	123	27	16	0	56	0
	0.05	1673	9	0	1664	123	33	11	0	31	0
	0.01	7183	8	0	7174	119	34	11	0	21	3.6
geant1	0.25	2551	35	0	2512	354	84	165	0	6205	0
	0.1	TL	22	0	TL	209	77	10	0	28	35.4
	0.05	TL	14	0	TL	215	79	29	0	69	32.6
	0.01	TL	24	0	TL	214	52	25	0	38	22.2
geant2	0.25	837	30	0	804	315	67	63	0	736	0
	0.1	TL	20	0	TL	290	79	35	0	84	36.6
	0.05	TL	20	0	TL	246	73	15	0	48	36.2
	0.01	TL	9	0	TL	255	83	21	0	37	36.5
di-yuan	0.25	4	0	0	3	49	9	15	0	24	0
	0.1	19	0	0	18	55	11	9	0	138	0
	0.05	9	0	0	8	56	5	17	0	140	0
	0.01	1	0	0	0	51	8	7	0	61	0
pdh	0.25	12	5	0	7	95	9	8	0	29	0
	0.1	15	3	0	11	95	13	9	0	1583	0
	0.05	3	2	0	0	94	31	12	0	902	0
	0.01	4	3	0	0	106	63	17	0	770	0
polska	0.25	TL	5	0	TL	61	2	18	1	48	17.9
	0.1	TL	3	0	TL	55	0	9	0	2	19.3
	0.05	TL	3	0	TL	47	2	11	0	2	17
	0.01	TL	3	0	TL	54	0	10	0	5	11.5
nobel-us	0.25	TL	10	0	TL	82	3	7	0	14	17.7
	0.1	TL	14	0	TL	72	0	8	0	8	16.3
	0.05	TL	10	0	TL	90	1	12	0	2	15.3
	0.01	TL	8	0	TL	83	5	9	0	11	11.9
atlanta	0.25	5228	12	0	5214	94	2	12	0	22	0
	0.1	TL	4	0	TL	90	5	14	0	7	7.8
	0.05	TL	5	0	TL	87	5	14	0	12	7.8
	0.01	TL	4	0	TL	83	3	11	0	5	6
newyork	0.25	22	10	0	0	373	96	255	0	22435	0
	0.1	32	19	0	0	364	85	257	0	23269	0
	0.05	34	18	0	1	383	101	266	0	27825	0
	0.01	20	8	0	1	401	73	273	0	22244	0
france	0.25	TL	73	0	TL	272	22	10	0	54	20
	0.1	TL	134	0	TL	289	0	14	0	57	21.2
	0.05	TL	0	0	TL	3	0	0	2	0	73.6
	0.01	TL	94	0	TL	272	0	15	0	54	18.8
Arithmetic mean		3520	14	0	3503	143	27	39	0	2235	10

Table 10: Details of Benders-I4 for dynamic routing.

Discrete Optimization, 4:103–114, 2007.

[7] J. Ayoub and M. Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Comput. Manag. Science*, 13(2):219–239, 2016.

[8] F. Babonneau, J.-P. Vial, O. Klopfenstein, and A. Ouorou. Robust capacity assignment solutions

for telecommunications networks with uncertain demands. *Networks*, 62(4):255–272, 2013.

[9] F. Barahona. Network design using cut inequalities. *SIAM Journal on Optimization*, 6:823–834, 1996.

[10] W. Ben-Ameur. Between fully dynamic routing and robust stable routing. In *Proceedings of DRCN2007*, pages 1–6, 2007.

[11] W. Ben-Ameur and H. Kerivin. Routing of uncertain traffic demands. *Optimization and Engineering*, 6:283–313, 2005.

[12] W. Ben-Ameur and M. Zotkiewicz. Volume oriented routing. In *Proceedings of NETWORKS2010)*, pages 1–7, 2010.

[13] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[14] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

[15] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[16] D. Bienstock, S. Chopra, O. Günlük, and C.-Y. Tsai. Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming*, 81:177–199, 1998.

[17] D. Bienstock and S. Mattia. Using mixed-integer programming to solve power grid blackout problems. *Discrete Optimization*, 4:115–141, 2007.

[18] A. Bley, R. Klaehne, U. Menne, C. Raack, and R. Wessaely. Multi-layer network design – A model-based optimization approach. In *Proceedings of the PGTS 2008, Berlin, Germany*, pages 107–116, 2008.

[19] V. Cacchiani, M. Jünger, F. Liers, A. Lodi, and D. R. Schmidt. Single-commodity robust network design with finite and hose demand sets. *Mathematical Programming*, 157(1):297–342, 2016.

[20] C. Chekuri, G. Oriolo, M. Scutellà, and F. Shepherd. Hardness of robust network design. *Networks*, 50(1):50–154, 2007.

[21] CPLEX. *IBM ILOG CPLEX 12.6 Reference Manual*. ILOG CPLEX Division, Gentilly, France, 2013.

[22] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, and J. van der Merive. A flexible model for resource management in virtual private networks. *SIGCOMM Computer Communication Review*, 29(4):95–108, 1999.

[23] J. Fingerhut, S. Suri, and J. Turner. Designing least-cost nonblocking broadband networks. *Journal of Algorithms*, 24(2):287 – 309, 1997.

[24] B. Fortz and M. Poss. An improved benders decomposition applied to a multi-layer network design problem. *Oper. Res. Lett.*, 37(5):359–364, 2009.

[25] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: a network design problem for multicommodity flows. In *Proceedings of ACMSTOC 2001*, pages 389–398, 2001.

[26] A. Koster, M. Kutschka, and C. Raack. Robust network design: Formulations, valid inequalities, and computations. *Networks*, 61(2):128–149, 2013.

[27] C. Lee, K. Lee, and S. Park. Benders decomposition approach for the robust network design problem with flow bifurcations. *Networks*, 62(1):1–16, 2013.

[28] C. Lemaréchal, A. Ouorou, and G. Petrou. Robust network design in telecommunications under polytope demand uncertainty. *European Journal of Operational Research*, 206(3):634–641, 2010.

[29] T. Magnanti, P. Mirchandani, and R. Vachani. The convex hull of two core capacitated network design problems. *Mathematical Programming*, 60:233–250, 1993.

[30] S. Mattia. Separating tight metric inequalities by bilevel programming. *Operations Research Letters*, 40(6):568–572, 2012.

[31] S. Mattia. Solving survivable two-layer network design problems by metric inequalities. *Computational Optimization and Applications*, 51(2):809–834, 2012.

[32] S. Mattia. A polyhedral study of the capacity formulation of the multilayer network design problem. *Networks*, 62(1):17–26, 2013.

[33] S. Mattia. The robust network loading problem with dynamic routing. *Computational Optimization and Applications*, 54(3):619–643, 2013.

[34] S. Mattia. The cut property under demand uncertainty. *Networks*, 66(2):159–168, 2015.

[35] S. Mattia, F. Rossi, M. Servilio, and S. Smriglio. Staffing and scheduling flexible call centers by two-stage robust optimization. *Omega*, in press.

[36] M. Minoux. Robust network optimization under polyhedral demand uncertainty is NP-hard. *Discrete Applied Mathematics*, 158(5):597–603, 2010.

[37] S. Mudchanatongsuk, F. Ordonez, and J. Liu. Robust solutions for network design under transportation cost and demand uncertainty. *Journal of the Operations Research Society*, 59:552–562, 2008.

[38] F. Ordóñez and J. Zhao. Robust capacity expansion of network flows. *Networks*, 50(2):136–145, 2007.

[39] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0–Survivable Network Design Library. *Networks*, 55(3):276–286, 2010.

[40] A. Ouorou. Tractable approximations to a robust capacity assignment model in telecommunications under demand uncertainty. *Computers & OR*, 40(1):318–327, 2013.

[41] A. Ouorou and J.-P. Vial. A model for robust capacity planning for telecommunications networks under demand uncertainty. In *Proceedings of DRCN 2007*, pages 1–4, 2007.

[42] M. Pioro, D. Nace, M. Poss, and Y. Fouquet. An optimization model for multicommodity flow networks with partial multiple link failures. *Operations Research*. Accepted.

[43] M. Poss. A comparison of routing sets for robust network design. *Optimization Letters*, 8(5):1619–1635, 2014.

[44] M. Poss and C. Raack. Affine recourse for the robust network design problem: Between static and dynamic routing. *Networks*, 61(2):180–198, 2013.

[45] M. Scutellà. On improving optimal oblivious routing. *Operations Research Letters*, 37(3):197–200, 2009.

[46] W. Wiesemann, D. Kuhn, and M. Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

[47] B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.

[48] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *Proceedings of ACM SIGMETRICS*, pages 206–217, 2003.