Efficient approaches for the robust network loading problem

Sara Mattia
Istituto di Analisi dei Sistemi ed Informatica
Consiglio Nazionale delle Ricerche
via dei Taurini 19, 00185 Roma (Italy)
sara.mattia@iasi.cnr.it

Michael Poss
UMR CNRS 5506 LIRMM
Université de Montpellier
rue Ada 161, 34095 Montpellier cedex 5, France
michael.poss@lirmm.fr

Abstract

We consider the Robust Network Loading problem with splittable flows and demands that belong to the budgeted uncertainty set. Four routing schemes are investigated: static, volume, affine and dynamic. First, following what done for static and dynamic routing, we derive capacity formulation for affine and volume routing. This is done using a Benders reformulation approach. Then, we introduce the robust 3-partition inequalities and discuss their facet-defining status and the one of the well-known robust cutset inequalities. Finally, we test the effectiveness, from a computational viewpoint, of robust cutset, 3-partition and Benders-like inequalities. We compare the results obtained considering all routing schemes, using both the flow formulation and the Benders-like formulation on the capacity variables. We report computational experiments realized on several realistic instances from SNDlib, including instances based on historical traffic data.

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 matrices), 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 limit, for instance, the type of flow admissible on the edges or the type of capacities that can be installed on the edges, as well as a large variety of technical considerations such as ensuring a given level of survivability in case of link failures or limiting the length of the paths used. Herein, we study the so-called *Network Loading Problem* where capacities can be installed by integer multiples of some capacity module, flows between the end-nodes of commodities can be split arbitrarily among different paths, and without additional technical constraints.

An important aspect of network design problems is related to the knowledge of demands. In a large number of applications, these demands are not available at the time we decide of the capacity installation. The best one can do is to rely on demand forecasts, based, for instance, on population statistics [16] or traffic measurements [43]. If these statistical studies are accurate enough, one can come up with a stochastic model that considers demands as known random 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. The best one can reasonably do is to define sets of admissible values of random variables compatible with the available data, falling into the framework of distributionally robust optimization.

In this work, we follow an even safer approach and consider that the uncertain demands are described through an uncertainty set [10], falling into the framework of robust optimization [12]. 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, 24, 30, 32, 33, 40]. A choice to model the forecast demands is to use the budgeted uncertainty set from [14]. The latter supposes that demands fluctuate between their nominal values and peak values and that at most Γ of them reach their peak values simultaneously. The model is motivated by probabilistic guarantees [14] and has been used in numerous papers on robust network design problems [8, 24, 40].

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 known as routing in the literature on network optimization. Different routings have been studied in the past, each with its own flexibility and computational issues. At one extreme, we find static routing, which imposes that the fractional splitting of the commodities among a fixed set of paths stay constant for all realizations of the demands. In the other extreme, dynamic routing allows the flows to be defined by arbitrary functions of the demands. 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 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 less hard than the dynamic one. Affine routing [36] 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 [13]. Affine routing has been used in several papers on robust network design, see for instance [8, 26, 35, 40], including a variation of the problem where it is the capacity, rather than the demand, which is uncertain [37]. Volume routing [11] 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 routings since the routing for a commodity only depends on the demand value for that commodity. Other intermediate routings have also been proposed in the literature, such as those based on dynamic partitions of the uncertainty sets [9, 41]. However, they lead to optimization problems that are even harder to solve than the problem with dynamic routing, see the discussion in [39]. 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 include the problem without uncertainty, and then the Steiner tree problem, as special case. However, 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, 11, 40]. In contrast, the problem with dynamic routing was proved to be $\mathcal{NP}$-hard, both in general and for the budgeted uncertainty set [17, 23, 31]. 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. This affects the computational performance of the problem with integer capacities. For the 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 [20, 21]. Similar numerical studies have been carried out in [24] for the problem with budgeted uncertainty. The authors study different classes of valid inequalities among which robust cutset inequalities stand out for their numerical efficiency. In [25] a Benders decomposition approach for the network loading problem with static routing and budgeted uncertainty is derived and studied. The same formulation is used in [18], where a computational analysis is also reported. In [30] the author studies the problem with dynamic routing under the Hose model, proposing a branch-and-cut procedure related to bilevel optimization. Finally, in [32] the problem with 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 bring together, study, test and compare four different routing schemes: static, affine, volume and dynamic. To the best of our knowledge this is the first time that such a comprehensive study is carried out. A first contribution of the paper is to derive a capacity formulation for the problem with volume and affine routing, thus generalizing what was done so far for static routing [25], using a Benders decomposition approach. For dynamic routing, the exponential numbers 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 porposed in the recent years for adjustable robust optimization, see [30, 7, 42]. We refer to the formulations including only capacity variables as *Benders formulations* in the rest of the paper, despite the differences in the techniques used to obtain such formulations. We investigate then the different routing schemes from the theoretical point of view. In particular, we present a new class of valid inequalities: the robust 3-partition inequalities. They generalize the 3-partition inequalities used for the problem without uncertainty [2]. We prove that they are facet-defining for the problem with dynamic routing. We also present examples showing that they are not facet-defining, under the same conditions, for the other routing schemes. On the contrary, the robust cutset inequalities, that were already known to be facet-defining for some of the routing schemes considered here [24, 30], can be proved to be facet-defining for all of them, independently of the uncertainty set and of the flows (splittable/unsplittable). Finally, we study the considered routing schemes from a computational perspective, investigating the trade-off between flexibility and computational efficiency. We compare the Benders formulations using different cutting plane approaches including robust cutset, 3-partition and Benders-like (also known as metric) inequalities. 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, if the flow formulation is strengthened using the above mentioned inequalities, its performance can be significantly improved.

The paper is structured as follows. The next section presents the formulations of the robust network loading problem for each of the aforementioned routings. Section 3 presents then Benders decompositions of these formulations. We present separately Benders decompositions for the problems with static/volume/affine routing (Section 3.1) and dynamic routing (Section 3.2). We describe the valid inequalities in Section 4, introducing the 3-partition inequalities and proving their facet-defining status and the one of the cutset inequalities. In Sections 5 and 6 we present our computational experiments, including the specifications of the implemented algorithms. The paper is concluded in Section 7.

2 Models and 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 let $\mathcal{U} \subset \mathbb{R}_+^{|K|}$ be the uncertainty polytope introduced by [14] and defined below. Each commodity $k \in K$ is defined by its endnodes $s(k)$ and $t(k)$ and its demand value d^k for any $d \in \mathcal{U}$. 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 network loading problem studied herein aims at installing the cheapest capacities x on the edges of the graph, such that all realizations of the demand vectors $d \in \mathcal{U}$ can be routed on the resulting network. In this paper, we suppose that the uncertainty set has a special structure, often used in the literature: each demand value d^k varies between its nominal value $\bar{d}^k$ and its peak demand value $\bar{d}^k + \hat{d}^k$ and that the number of deviations from the nominal value is bounded by integer Γ. It will be useful in the following to formulate $\mathcal{U}$ through the extended formulation below

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

The problem can be formulated mathematically as follows. Integer variable x_e represents the capacity allocation on edge $e \in E$ and real variable $f_{ij}^k(d)$ describes the amount of flow for demand d and commodity k routed on each arc $(i,j) \in A$. We also define the star of node $i' \in N$ as $\delta(i') = \{e = \{i,j\} \in E : i = i' \text{ or } j = i'\}$.

$$
\begin{aligned}
(RNL) \quad \min \quad & \sum_{e \in E} c_e x_e \\
\text{s.t.} \quad & \sum_{j \in \delta(i)} (f_{ji}^k(d) - f_{ij}^k(d)) = \begin{cases} d^k & \text{if } i = t(k) \\ 0 & \text{otherwise} \end{cases} \qquad i \in V \backslash \{s(k)\}, k \in K, d \in \mathcal{U} && \text{(2a)} \\
& \sum_{k \in K} (f_{ij}^k(d) + f_{ji}^k(d)) \leq x_e \qquad e = \{i,j\} \in E, d \in \mathcal{U} && \text{(2b)} \\
& f, x \geq 0 && \text{(2c)} \\
& x \in \mathbb{Z}^{|E|} && \text{(2d)}
\end{aligned}
$$

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 denote the flow function f as the *routing*. We use the term *dynamic routing* when no particular assumption is made on admissible functions, as in (RNL).

Problem (RNL) is a MILP with an infinite number of variables and constraints. However, we see easily that the problem can be discretized by considering the extreme points of $\mathcal{U}$, denoted $\text{vert}(\mathcal{U})$. Hence, we can replace $\mathcal{U}$ by $\text{vert}(\mathcal{U})$ in (RNL), yielding a finite mixed-integer linear formulation for the problem. However, the resulting formulation is extremely large since the number of extreme points of $\mathcal{U}$ grows exponentially with $\Gamma \leq |K|$. Namely, one readily sees that

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

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 perform such a decomposition. Alternatively, we could restrict the routing to simple functions of d. Rather than letting f be an arbitrary function of d, we enforce that the following restrictions be satisfied

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

Constraints (3) yields what is known as affine decision rules or *affine routing* in the literature [35, 36, 40]. The constraints limit function f to affine functions of d. Plugging constraints (3) into (RNL), then dualizing inequality constraints (2b) and identifying the terms of equality constraints (2a), we obtain the polynomial reformulation for the problem described in the following result. The full details of the reformulation are omitted since they rely on applying well-known techniques from robust optimization, see [12].

Lemma 1. *Let $\mathcal{U}$ be the polytope described in* (1)*. Then problem* (RNL) *together with constraints* (3) *can be reformulated as follows:*

$$
\begin{aligned}
(ARNL) \quad \min \quad & \sum_{e\in E} c_e x_e \\
s.t. \quad & \sum_{j\in\delta(i)} (y^k_{ij} - y^k_{ji}) = \begin{cases} 1 & \textit{if } i = t(k) \\ 0 & \textit{otherwise} \end{cases} && i \in V\backslash\{s(k)\}, k \in K && \text{(4a)} \\
& \sum_{j\in\delta(i)} (y^{kh}_{ji} - y^{kh}_{ij}) = 0 && i \in V\backslash\{s(k)\}, k \neq h \in K, && \text{(4b)} \\
& \sum_{j\in\delta(i)} (f^{0k}_{ji} - f^{0k}_{ij}) = 0 && i \in V\backslash\{s(k)\}, k \in K && \text{(4c)} \\
& \Gamma z_e + \sum_{k\in K} \left(p^k_e + f^{0k}_{ij} + f^{0k}_{ji} + \sum_{h\in K} (y^{kh}_{ij} + y^{kh}_{ji}) \bar{d}^h \right) \leq x_e && e = \{i,j\} \in E && \text{(4d)} \\
& z_e + p^k_e \geq \sum_{k\in K} (y^{kh}_{ij} + y^{kh}_{ji}) \hat{d}^h && e = \{i,j\} \in E, h \in K && \text{(4e)} \\
& \Gamma s_{ij} - f^{0k}_{ij} + \sum_{h\in K} \left(q^{kh}_{ij} - y^{kh}_{ij} \bar{d}^h \right) \leq 0 && (i,j) \in A, k \in K && \text{(4f)} \\
& s_{ij} + q^{kh}_{ij} \geq y^{kh}_{ij} \hat{d}^h && (i,j) \in A, k \in K, h \in K && \text{(4g)} \\
& z, s, p, q, x \geq 0, && && \text{(4h)} \\
& x \in \mathbb{Z}^{|E|} && && \text{(4i)}
\end{aligned}
$$

where z, s, p, q and f^0 and are the optimization variables that come from dualizing the robust constraints.

Although compact, formulation $(ARNL)$ is quadratic in $|K|$, which makes the problem significantly harder to solve than its deterministic counterpart. To avoid this quadratic dependency on $|K|$, we can consider special cases of affine routing that contain less degrees of freedom than (3). We present below two special cases of affine routing previously studied in the literature.

2.1 Volume routing

We enforce f to satisfy

$$f^k_{ij}(d) = f^{0k}_{ij} + y^k_{ij} d^k, \quad k \in K, (i,j) \in A. \tag{5}$$

Roughly speaking, (5) means that the flow is defined by set of paths from $s(k)$ to $t(k)$ (defined by y), which is modified by the circulation described by f^0, see [40]. Volume routing has been originally introduced in [11]. Plugging constraints (5) into (RNL) and applying the techniques mentioned above yields the following reformulation for the problem:

$$
\begin{aligned}
(VRNL) \quad \min \quad & \sum_{e\in E} c_e x_e \\
\text{s.t.} \quad & \sum_{j\in\delta(i)} (y^k_{ji} - y^k_{ij}) = \begin{cases} 1 & \text{if } i = t(k) \\ 0 & \text{otherwise} \end{cases} && i \in V\backslash\{s(k)\}, k \in K && \text{(6a)}
\end{aligned}
$$

$$\sum_{j\in\delta(i)} (f_{ji}^{0k} - f_{ij}^{0k}) = 0 \qquad i \in V\backslash\{s(k)\}, k \in K \quad (6b)$$

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

$$z_e + p_e^k \geq (y_{ij}^{kk} + y_{ji}^{kk})\hat{d}^k \qquad e = \{i,j\} \in E, k \in K \quad (6d)$$

$$f_{ij}^{0k} + y_{ij}^{kk}\bar{d}^k \geq 0 \qquad (i,j) \in A, k \in K \quad (6e)$$

$$f_{ij}^{0k} + y_{ij}^{kk}(\bar{d}^k + \hat{d}^k) \geq 0 \qquad (i,j) \in A, k \in K \quad (6f)$$

$$z, p, x \geq 0. \quad (6g)$$

$$x \in \mathbb{Z}^{|E|} \quad (6h)$$

Formulation (6) can be derived from formulation (4) by removing flow conservation constraints (4c), replacing robust non-negativity constraints (4f) and (4g) by (6e) and (6f), and removing the terms corresponding to y^{kh} for $h \neq k$ in robust capacity constraints (4d) and (4e). Notice that other authors have proposed different restrictions of affine routing to reduce the size of the reformulations [8].

2.2 Static routing

We enforce f to satisfy

$$f_{ij}^k(d) = y_{ij}^k d^k, \quad k \in K, (i,j) \in A. \quad (7)$$

Static routing is a well-known framework for the robust network loading problem. It has been studied long before affine routing was introduced, see [20, 21], and recent works [4, 24] have extended valid inequalities to network loading problems with static routing. When constraints (7) are satisfied, variables 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 (7) into (RNL) and applying the techniques mentioned above yields the following reformulation for the problem:

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

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

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

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

$$z, p, y, x \geq 0. \quad (8d)$$

$$x \in \mathbb{Z}^{|E|} \quad (8e)$$

Formulation (8) can be derived from formulation (4) by removing flow conservation constraints (4b) and (4c), replacing robust non-negativity constraints (4f) and (4g) by $y \geq 0$, and removing the terms corresponding to f^0 and y^{kh} for $h \neq k$ in robust capacity constraints (4d) and (4e).

In the following, we sometimes commit an abuse of language and refer to $(SRNL)$, $(VRNL)$, $(ARNL)$, and (RNL) as the static problem, volume problem, affine problem, and dynamic problem, respectively.

3 Benders decomposition

Benders decomposition is a technique that projects out part of the continuous variables of a MILP, 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. A standard choice in solving networks design problems consists of projecting out the flow variables [1, 6, 25, 30, 29] working on a master problem which includes only the capacity variables. This corresponds to the practical decomposition of the decision process: in fact, capacity variables correspond to long term decisions, whereas flow variables correspond to decisions taken at the operational level (short term decisions).

We explain in this section how to apply Benders decomposition to (RNL) and $(ARNL)$. We point out that the purposes of using Benders decomposition are different for (RNL) and $(ARNL)$. For $(ARNL)$ (and its simplifications $(SRNL)$ and $(VRNL)$), Benders decomposition avoids solving a large linear program at each node of the branch-and-bound tree. The situation is different for (RNL) since the MILP 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 for (RNL) avoids to consider explicitly all vectors in $\text{vert}(\mathcal{U})$. Instead, needed extreme points are generated on the fly by solving a separation problem. We start in Section 3.1 with problem $(ARNL)$ for which a classical Benders reformulation is presented. We present then in Section 3.2 the approach used for problem (RNL).

3.1 Affine routing and simplifications

We project the flow variables out of formulation (4) and let $\mathcal{B}^{aff}$ be the projection of the set defined by constraints (4a)–(4h), formally:

$$\mathcal{B}^{aff} \equiv \{x \in \mathbb{R}^{|E|} : \exists f^0 \in \mathbb{R}^{|A|\times|K|} \text{ and } y \in \mathbb{R}^{|A|\times|K|\times|K|} \text{ that satisfy } (4a) - (4h)\}.$$

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

$$\begin{aligned}
(FeasAff) \quad & \min \alpha \\
\text{s.t.} \quad & \sum_{j\in\delta(i)} (y^k_{ji} - y^k_{ij}) = 1 && i = t(k), k \in K \qquad (9a)\\
& \Gamma z_e + \sum_{k\in K} \left(p^k_e + f^{0k}_{ij} + f^{0k}_{ji} + \sum_{h\in K} (y^{kh}_{ij} + y^{kh}_{ji}) \bar{d}^h \right) \le x_e + \alpha && e = \{i,j\} \in E \qquad (9b)\\
& (4b), (4c), (4e), (4f), (4g),
\end{aligned}$$

where α represents the amount of capacity required to route all demands on the network. One readily sees that $x \in \mathcal{B}^{aff}$ if and only if the optimal solution cost of $(FeasAff)$ is non-positive. Let $\mathcal{D}^{aff}$ be the feasibility polyhedron of the dual of problem $(FeasAff)$, and let π and μ denote the dual variables of constraints (9a) and (9b), 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: $(\pi, \mu) \in \text{vert}(\mathcal{D}^{aff})$ means that there exists a vector λ of appropriate dimension such that $(\pi, \mu, \lambda) \in \text{vert}(\mathcal{D}^{aff})$. The Benders reformulation of $(ARNL)$ is formalized in the next lemma.

Lemma 2. *Let $x \in \mathbb{R}^{|E|}$ be given. It holds that $x \in \mathcal{B}^{aff}$ if and only if*

$$-\sum_{e\in E} \mu_e x_e + \sum_{k\in K} \pi^k \le 0, \qquad (10)$$

for each $(\pi, \mu) \in \text{vert}(\mathcal{D}^{aff})$.

Proof. Recall that $x \in \mathcal{B}$ if and only if the optimal solution cost of $(FeasAff)$ is non-positive. Problem $(FeasAff)$ is always feasible and bounded so that its optimal solution cost is equal to the optimal solution cost of its dual. The optimal solution of the dual is always obtained at an extreme point of $\mathcal{D}^{aff}$, which yields the result. □

Lemma 2 provides a description of $\mathcal{B}^{aff}$ that may contain an exponential number of constraints. 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 (10) is that they can be separated "easily" by solving compact linear program $(FeasAff)$ or its dual.
The Benders reformulations of $(SRNL)$ and $(VRNL)$ are obtained similarly. Namely, we define

$$\mathcal{B}^{stat} \equiv \{x \in \mathbb{R}^{|E|} : \exists y \in \mathbb{R}_+^{|A|\times|K|} \text{ that satisfies (8a)} - \text{(8d)}\}.$$

and

$$\mathcal{B}^{vol} \equiv \{x \in \mathbb{R}^{|E|} : \exists f^0 \in \mathbb{R}^{|A|\times|K|} \text{ and } y \in \mathbb{R}^{|A|\times|K|} \text{ that satisfy (6a)} - \text{(6g)}\}.$$

To check whether a given vector x belongs to $\mathcal{B}^{stat}$ or $\mathcal{B}^{vol}$, we can introduce feasibility problems $(FeasStat)$ and $(FeasVol)$ as before. We omit the formulations of these problems since they are obtained from $(FeasAff)$ in the same way $(SRNL)$ and $(VRNL)$ are obtained from $(ARNL)$. We denote the feasibility polyhedrons of the duals of $(FeasStat)$ and $(FeasVol)$ by $\mathcal{D}^{stat}$ and $\mathcal{D}^{vol}$, respectively. The Benders reformulation of problems $(SRNL)$ and $(VRNL)$ is formalized in the next lemma. Its proof is similar to the proof of Lemma 2, and is therefore, omitted.

Lemma 3. *Let $x \in \mathbb{R}^{|E|}$ be given. The following holds:*

- $x \in \mathcal{B}^{stat}$ *if and only if x satisfies inequality* (10) *for each* $(\pi, \mu) \in \text{vert}(\mathcal{D}^{stat})$,
- $x \in \mathcal{B}^{vol}$ *if and only if x satisfies inequality* (10) *for each* $(\pi, \mu) \in \text{vert}(\mathcal{D}^{vol})$.

3.2 Dynamic routing

We explain in this section how to apply Benders decomposition to problem (RNL). Notice that the approach presented next is more intricate than the one presented in the previous section, because the variables and constraints of (RNL) are indexed by the (exponential) number of extreme points of $\mathcal{U}$ (recall that we can replace $\mathcal{U}$ by vert$(\mathcal{U})$ in (RNL)). As in the previous section, we project the flow variables out of formulation (2) and let $\mathcal{B}$ be the projection of the set defined by constraints (2a) and (2b):

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

One readily sees that a given $x \in \mathbb{R}^{|E|}$ belongs to $\mathcal{B}$ if and only the optimal solution cost of the feasibility problem below is non-positive

$$\begin{aligned}
&\min \alpha \\
&\text{s.t.} \sum_{j\in\delta(i)} (f_{ji}^k(d) - f_{ij}^k(d)) = d^k && i = t(k), k \in K, d \in \text{vert}(\mathcal{U}) \\
&\quad\;\; \sum_{j\in\delta(i)} (f_{ji}^k(d) - f_{ij}^k(d)) = 0 && i \in V\backslash\{s(k), t(k)\}, k \in K, d \in \text{vert}(\mathcal{U}) \\
&\quad\;\; \sum_{k\in K} (f_{ij}^k(d) + f_{ji}^k(d)) \le x_e + \alpha && e = \{i, j\} \in E, d \in \text{vert}(\mathcal{U}) \\
&\quad\;\; f, \alpha \ge 0.
\end{aligned}$$

We can reformulate the above feasibility problem by aggregating commodities with common source as it is often done for the deterministic network loading problem. To keep simple notations, we assume without loss of generality that set K contains one commodity for each (directed) pair of nodes in V. If this is not true, we can always add dummy commodities with demand value equal to 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 u in the new set of commodities V can be identified by its source node u and contains $|V| - 1$ sink nodes. For each $u, v \in V$, we denote the demand at node v for commodity u by d_v^u, which is equal to $-\sum_{v\in V\setminus\{u\}} d^{k(u,v)}$ if $u = v$ or $d^{k(u,v)}$ if $u \neq v$. One readily sees that the counterpart of uncertainty set $\mathcal{U}$ in the problem with aggregated commodities is $\mathcal{U}^{agg} \equiv$

$$\left\{ d \in \mathbb{R}^{|V|\times|V|} \;\middle|\; \exists \delta \in [0,1]^{|K|} \text{ s.t. } \begin{array}{ll} d_v^u = \bar{d}^{k(u,v)} + \delta^{k(u,v)}\hat{d}^{k(u,v)}, & u \neq v \in V \\ d_u^u = -\sum\limits_{v\in V\setminus\{u\}} (\bar{d}^{k(u,v)} + \delta^{k(u,v)}\hat{d}^{k(u,v)}), & u \in V \end{array}, \sum_{k\in K} \delta^k \leq \Gamma \right\}. \tag{12}$$

With the set of aggregated commodities, the above problem can be rewritten as

$$\begin{aligned} (Feas) \quad & \min \alpha \\ \text{s.t.} \quad & \sum_{j\in\delta(i)} (f_{ji}^u(d) - f_{ij}^u(d)) = d_i^u && u \neq i \in V, d \in \text{vert}(\mathcal{U}^{agg}) \\ & \sum_{u\in V} (f_{ij}^u(d) + f_{ji}^u(d)) \leq x_e + \alpha && e = \{i,j\} \in E, d \in \text{vert}(\mathcal{U}^{agg}) \\ & f, \alpha \geq 0. \end{aligned}$$

Even with a reduced set of commodities, $(Feas)$ can be very hard to solve since it contains exponential numbers of variables and constraints. The purpose of the Benders reformulation presented below is to replace the linear program $(Feas)$ that contains exponentially many variables and constraints by a compact MILP. We recall that this situation is in sharp contrast with $(FeasAff)$ which is a compact linear program. The proof of the next result applies recent techniques for adjustable robust optimization [42] to (RNL).

Lemma 4. *Let $x \in \mathbb{R}_+^{|E|}$ be given. It holds that $x \in \mathcal{B}$ if and only if the solution of the following MILP is non-positive:*

$$\max \; -\sum_{e\in E} \mu_e x_e + \sum_{u\neq v\in V} (\bar{d}^{k(u,v)}\pi_v^u + \hat{d}^{k(u,v)}\rho_v^u) \tag{14a}$$

$$\begin{aligned} s.t. \quad & \rho_v^u \leq \delta^{k(u,v)} && u \neq v \in V && (14b) \\ & \rho_v^u \leq \pi_v^u && u \neq v \in V && (14c) \\ & \rho_v^u \geq \pi_v^u + \delta^{k(u,v)} - 1 && u \neq v \in V && (14d) \\ & \sum_{k\in K} \delta^k \leq \Gamma && k \in K && (14e) \\ & \pi_i^u - \pi_j^u \leq \mu_{\{i,j\}} && a = (i,j) \in A, u \in V && (14f) \\ & \sum_{e\in E} \mu_e \leq 1 && && (14g) \\ & \pi_v^v = 0 && v \in V && (14h) \\ & \delta \in \{0,1\}^{|K|} && && (14i) \\ & \mu, \pi \geq 0 && && (14j) \\ & \rho \geq 0. && && (14k) \end{aligned}$$

Proof. The first step of the proof replaces $(Feas)$ by one feasibility problem for each $d \in \mathcal{U}^{agg}$:

$$\begin{aligned} (Feas\text{-}d) \quad & \min \alpha \\ \text{s.t.} \quad & \sum_{j\in\delta(i)} (f_{ji}^u - f_{ij}^u) = d_i^u && u \neq i \in V, && (15a) \\ & \sum_{u\in V} (f_{ij}^u + f_{ji}^u) \leq x_e + \alpha && e = \{i,j\} \in E, && (15b) \end{aligned}$$

$$f, \alpha \geq 0. \tag{15c}$$

We see easily that the optimal solution cost of $(Feas)$ is non-positive if and only if

$$\max_{d \in \mathcal{U}^{agg}} \quad \begin{aligned} & \min \ \alpha \\ & \text{s.t. } (15a) - (15c) \end{aligned} \tag{16}$$

is non-positive. This is the crucial step of our reformulation, where the exponential numbers of variables and constraints are replaced by a maximization over $\text{vert}(\mathcal{U}^{agg})$. The next two steps amount to provide a mixed-integer reformulation of problem (16). Let π denote the dual variables of constraints (15a), μ denote the dual variables of constraints (15b). Dualizing the inner minmization problem of (16), we obtain the following bilinear program:

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

$$\text{s.t.} \quad \sum_{k \in K} \delta^k \leq \Gamma \qquad k \in K \tag{17a}$$

$$\pi_i^u - \pi_j^u \leq \mu_{\{i,j\}} \qquad a = (i,j) \in A, u \in V \tag{17b}$$

$$\sum_{e \in E} \mu_e \leq 1 \tag{17c}$$

$$\pi_v^v = 0 \qquad v \in V \tag{17d}$$

$$\delta \in [0,1]^{|K|} \tag{17e}$$

$$\mu, \pi \geq 0. \tag{17f}$$

One sees easily that the optimal solution of problem (17) is reached at some binary vector δ, so that we can replace (17e) by $\delta \in \{0,1\}^{|K|}$. The result finally follows from representing product $\delta^{k(u,v)} \pi_v^u$ by auxiliary variable ρ_v^u for each $u \neq v \in V$, which can be done without introducing big-M coefficients because any solution of (17) satisfies $\pi_v^u \leq 1$ for each $u, v \in V$. □

The crucial step in the proof above enforces δ to be binary. This allows us to use classical techniques for linearizing the product of a binary variable and a bounded continuous variable. Notice that not all variables ρ are needed in the formulation above: if $\hat{d}^{k(u,v)} = 0$ for some $u \neq v \in V$ then ρ_v^u can be removed from the formulation.

We finish the section by providing without proof the counterpart of Lemma 2 for (RNL).

Lemma 5. *Let $x \in \mathbb{R}^{|E|}$ be given and let $\mathcal{D}$ be the polytope defined by constraints* (14f)*,* (14g)*, and* (14j)*. It holds that $x \in \mathcal{B}$ if and only if*

$$-\sum_{e \in E} \mu_e x_e + \sum_{u \neq v \in V} d^{k(u,v)} \pi_v^u \leq 0, \tag{18}$$

for each $(\pi, \mu) \in \text{vert}(\mathcal{D})$ and $d \in \text{vert}(\mathcal{U})$.

4 Valid inequalities

Benders cuts (10) and (18) describe the projections on variables x of the feasibility sets of the linear relaxations of (RNL), $(ARNL)$, $(SRNL)$, and $(VRNL)$. However, these cuts in their simple forms do not take into account the fact that these problems look for *integer* x. Restricting set $\mathcal{B}$ to integer values of x we obtain the set of capacities that are feasible for (RNL): $\mathcal{B}_I = \mathcal{B} \cap \mathbb{Z}^{|E|}$; we define similarly $\mathcal{B}_I^{aff}$, $\mathcal{B}_I^{stat}$, and $\mathcal{B}_I^{vol}$. We present next a simple lemma, useful when describing valid inequalities for the problem with integer variables.

Lemma 6. *It holds that $\mathcal{B}_I^{stat} \subseteq \mathcal{B}_I^{vol} \subseteq \mathcal{B}_I^{aff} \subseteq \mathcal{B}_I$.*

Proof. The proof follows from analyzing the flexibility of the different routing schemes involved. Namely, the feasibility problem for $(ARNL)$ (before dualization and substitution of the robust constraints) is similar to the feasibility problem for (RNL), but with additional constraints (3). Hence, $\mathcal{B}^{aff} \subseteq \mathcal{B}$, which in turn implies $\mathcal{B}_I^{aff} \subseteq \mathcal{B}_I$. The other inclusions are obtained from the fact that $(VRNL)$ (resp. $(SRNL)$) is obtained from $(ARNL)$ (resp. $(VRNL)$) by removing some of the variables that appear in (3). □

The lemma implies that any valid inequality for $\mathcal{B}_I$ is also a valid inequality for $\mathcal{B}_I^{stat}$, $\mathcal{B}_I^{vol}$, and $\mathcal{B}_I^{aff}$, as already mentioned in [30]. In the same way, inequalities that are valid for affine (but possibly not for dynamic) routing are also valid for volume and static and so on. However, although an inequality may be valid for more than one routing scheme, it may in principle have a different strength according to the considered polyhedron. We illustrate it from the computational point of view in our experiments. From the theoretical point of view, an inequality can have a different facet-defining status according to the considered polyhedron. These considerations are formalized in the corollary below. We denote by $P^r(\mathcal{U})$ the problem with uncertainty set $\mathcal{U}$ under routing policy r and by $\mathcal{B}_I^r(\mathcal{U})$ the convex hull of integer feasible solutions of the Benders formulation of $P^r(\mathcal{U})$. Given two routing policies r and r', with a little abuse of notation we say that r' includes r ($r' \supseteq r$ or, equivalently, $r \subseteq r'$), if $\mathcal{B}_I^r(\mathcal{U}) \subseteq \mathcal{B}_I^{r'}(\mathcal{U})$, that is, any feasible solution that is feasible for r is also feasible for r'. It is easy to see that the definition is independent of $\mathcal{U}$ so that we remove the explicit dependency on $\mathcal{U}$ in what follows. The following lemma formalizes simple properties of valid inequalities of $\mathcal{B}_I^r$.

Lemma 7. *Let $\alpha^T x \leq \beta$ be a valid inequality for $\mathcal{B}_I^r$. Independently of $\mathcal{U}$, the following holds:*

1. *$\alpha^T x \leq \beta$ is a valid inequality for $\mathcal{B}_I^{r'}$, for any $r' \subseteq r$;*
2. *if $\alpha^T x \leq \beta$ is not facet-defining for $\mathcal{B}_I^r$, then it is not facet-defining for $\mathcal{B}_I^{r'}$, for any $r' \subseteq r$;*
3. *if $\alpha^T x \leq \beta$ is facet-defining for $\mathcal{B}_I^r$, then it is facet-defining for $\mathcal{B}_I^{r'}$, for any $r' \supseteq r$.*

As proved in [30], for the considered Benders formulations, all facet-defining inequalities $a^T x \geq b$ are tight metric inequalities, that is, the left-hand-side coefficients $a \geq 0$ must define a metric on G (for any $e \in E$, a_e is the shortest path length between the endpoints of e according to weights a) and b must be the value of the optimal solution of the problem where a is used as objective. In what follows, we concentrate on two classes of valid inequalities for $\mathcal{B}_I$, $\mathcal{B}_I^{aff}$, $\mathcal{B}_I^{stat}$, and $\mathcal{B}_I^{vol}$: the well-known robust cutset inequalities and a new class of inequalities, namely, the robust 3-partition inequalities. We investigate their facet status for all the considered polyhedra. We remember that all those sets are full-dimensional [30], then, for an inequality to be a facet, we must produce $|E|$ affinely independent feasible capacity vectors satisfying it with equality.

4.1 Cutset inequalities

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 edges and commodities with endpoints in different sets of the partition. The 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 be not less than the rounded up sum of the demands of commodities in $K(S_1, S_2)$. In the deterministic network design problem, where $\mathcal{U}$ is reduced to a singleton, the inequality can be written as

$$\sum_{e \in E(S_1,S_2)} x_e \geq \left\lceil \sum_{k \in K(S_1,S_2)} d^k \right\rceil . \tag{19}$$

One readily sees that the robust version of cutset inequality (19) is valid for $\mathcal{B}_I$:

$$\sum_{e \in E(S_1,S_2)} x_e \geq \left\lceil \max_{d \in \mathcal{U}} \sum_{k \in K(S_1,S_2)} d^k \right\rceil . \tag{20}$$

Lemma 7 implies that cutset inequalities are also valid for $\mathcal{B}_I^{aff}$, $\mathcal{B}_I^{stat}$, and $\mathcal{B}_I^{vol}$. They were already proved to be valid and facet-defining for some routing schemes [24, 30]. Since they are known to be facet-defining for splittable flows under static/dynamic routing and budgeted-uncertainty, Lemma 7 already implies that they are facet-defining for splittable flows and budgeted uncertainty also for volume/affine routing. In [30] it is already discussed that they are actually facet-defining for all the considered routing schemes, independently of the uncertainty set. Here we give a formal proof of that result. It holds for both splittable and unsplittable flows (commodities are required to follow a single path in the network), for both undirected and for directed (the graph and the demands are directed) problems.

Consider problem P^r and let the k-node problem $P^r(V_l)$ be the problem obtained from a l-partition $V_l = [S_1 : \ldots : S_l]$ of the nodes by shrinking each subset of the partition into a single node (possibly merging the parallel edges, if any) and by projecting the demand polyhedron $\mathcal{U}$ in the corresponding reduced space. We say that a partition V_l is connected if S_i is connected for any $S_i \in V_l$. The following result holds.

Theorem 8. *[30] Let $a^T x \geq b$ be a facet defining inequality for $\mathcal{B}_I^r(V_l)$ for any $l \geq 2$ and let V_l be connected, then the inequality is facet defining for $\mathcal{B}_I^r$).*

Using the above result, we can prove what follows.

Theorem 9. *If V_2 is connected, the cutset inequality (20) is facet defining for $\mathcal{B}_I^r$, for any r and $\mathcal{U}$.*

Proof. If we consider a two-node problem, then all routing schemes collapse into static routing, as there is only one path in the network for every commodity. For the same reason, splittable and unsplittable flow coincide. It is also trivial to see that the cutset inequality corresponding to the unique non trivial cut $[\{v_1\} : \{v_2\}]$ is facet-defining for $\mathcal{B}_I^r(V_2)$. The same is true for directed networks considering strong connection (each undirected cut is replaced by the outgoing and the incoming cut). Therefore, by Theorem 8, cutset inequalities are facet defining for $\mathcal{B}_I^r$. □

4.2 Robust 3-partition inequalities

Consider a 3-partition of the nodes $V_3 = [S_1, S_2, S_3]$ and let $\bar{S}_i = V \backslash S_i$ for $i = 1, 2, 3$. The robust 3-partition inequality associated with the partition is obtained by summing the three cutset inequalities associated with partition $V_2 = [S_i : \bar{S}_i]$ 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 inequality (20) 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

$$\sum_{e \in E(S_1, S_2, S_3)} x_e \geq \left\lceil \frac{1}{2}(rhs_1 + rhs_2 + rhs_3) \right\rceil . \tag{21}$$

This is the robust version of the 3-partition inequalities in [2, 27]. In the following we prove under which conditions (21) is facet-defining for (RNL). We also provide an example showing that the result does not extend to $(SRNL)$, $(VRNL)$, and $(ARNL)$.

Theorem 10. *Inequality* (21) *is facet-defining for the $\mathcal{B}_I(V_3)$ if and only if the following conditions are satisfied:*

1. *$rhs_1 + rhs_2 + rhs_3$ is odd;*
2. *$rhs_i > 0$ for each $i = 1, \ldots, 3$;*
3. *$p = \lceil \frac{rhs_1 + rhs_2 + rhs_3}{2} \rceil > rhs_i$ for each $i = 1, \ldots, 3$.*

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

Table 1: Affinely independent vectors for Theorem 10.

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

Necessity. Suppose that condition 1 is not satisfied. Then the 3-partition inequality is dominated by the sum of the cutset inequalities. Therefore it cannot be facet-defining. Suppose that condition 2 is not satisfied. Then there exists cutset inequality i such that $rhs_i = 0$. Let us suppose that $i = 1$. This means that the problem reduces to a single commodity problem with commodity between S_2 and S_3 and then $rhs_2 = rhs_3$. Therefore $rhs_1 + rhs_2 + rhs_3$ cannot be odd and then, by the proof of condition 1, the 3-partition inequality cannot be facet-defining. Suppose that condition 3 is not satisfied. Then there exists cutset inequality i such that $rhs_i \geq p$. Hence the 3-partition inequality is dominated by the cutset inequality i.

Sufficiency. We suppose without loss of generality that $rhs_1 \geq rhs_2 \geq rhs_3$. According to the values of p and rhs_i for $i = 1, \ldots 3$ we distinguish four case: (i) $p \geq rhs_1 + rhs_2 \geq rhs_1 + rhs_3 \geq rhs_2 + rhs_3$; (ii) $rhs_1 + rhs_2 > p \geq rhs_1 + rhs_3 \geq rhs_2 + rhs_3$; (iii) $rhs_1 + rhs_2 \geq rhs_1 + rhs_3 > p \geq rhs_2 + rhs_3$; (iv) $rhs_1 + rhs_2 \geq rhs_1 + rhs_3 \geq rhs_2 + rhs_3 > p$. For each case we provide in Table 4.2 three affinely independent vectors that are feasible for the problem and satisfy the 3-partition inequality with equality. Since the dynamic problem on 3 nodes has the robust cut property [28], the feasibility of a solution can be tested checking the cutset inequalities only. It is easy to see that for each case the listed vectors satisfy the cutset inequalities. Moreover, one readily verifies that Conditions 1–3 imply that the four 3×3 matrices from Table 4.2 are non-singular. □

Therefore, we can prove what follows.

Corollary 11. *Inequalities* (21) *corresponding to connected partitions* $[S_1 : S_2 : S_3]$ *that satisfy the conditions of Theorem 10 are facet-defining for* $B_{dyn}(\mathcal{U})$*.*

Proof. The result follows from Theorems 8 and 10. □

We point out that the above results are independent of the uncertainty set $\mathcal{U}$ considered. We provide next an example of 3-partition inequality that satisfies the conditions of Theorem 10, but it is not facet-defining for $(SRNL)$, not even for the budgeted uncertainty set.

Example 12. *Consider a complete undirected graph with 3 nodes and let* $\mathcal{U}$ *be the budgeted uncertainty set. 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 cutset inequalities and the corresponding 3-partition inequality are reported below.*

$$c_1 : \quad x_{12} + x_{13} \geq 3$$

$$c_2 : \quad x_{12} + x_{23} \geq 2$$

$$\begin{aligned} c_3 : \quad & x_{13} + x_{23} \geq 2 \\ 3p : \quad & x_{12} + x_{13} + x_{23} \geq 4 \end{aligned}$$

The conditions of Theorem 10 are satisfied, but it is not possible to find three affinely independent integer vectors that are feasible for $(SRNL)$ and satisfy $3p$ with equality. In fact, the only capacity allocation x that is feasible for $(SRNL)$ and satisfies $3p$ with equality is $x_{12} = x_{13} = 2$, $x_{23} = 0$.

One can construct similar examples for $(VRNL)$ and $(ARNL)$, using the property that affine routing reduces to static routing whenever the uncertainty set contains the corner of $\mathbb{R}_+^{|K|}$ [40] Interestingly, this is the first case of an inequality being facet-defining for the deterministic problem but not for the robust problem for some routing template, at least not under the conditions that hold for dynamic routing.

5 Test-bed and implementation

The purpose of the computational experiments presented in this section is two-fold. First we compare the solvability of (RNL), $(SRNL)$, and $(VRNL)$ using different formulations algorithms and settings. Second, we compare the optimal solution cost of (RNL), $(SRNL)$, and $(VRNL)$ on realistic instances. Let opt(P) denote the optimal solution of optimization problem (P). Then, we see immediately than the following ordering holds:

$$\text{opt}(RNL) \leq \text{opt}(ARNL) \leq \text{opt}(VRNL) \leq \text{opt}(SRNL).$$

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

5.1 The test-bed

We test our models and algorithms on twelve realistic network instances available from SNDlib [34]. The main characteristics of these networks are reminded in Table 2. Networks *abilene*, *germany*, and *geant* come from [24], 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 [24], 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, 40].

We choose the value of Γ according to the probabilistic bound introduced by [14] (the improved bounds from [38] can hardly be used in this paper because flows are fractionary). 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 bound from [14] prescribes a value Γ^ϵ such that all feasible solutions to (RNL) and $(ARNL)$ 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.

5.2 Implementation

Solution approaches have been coded in JAVA using Cplex Concert Technology. 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 [19]. We allow 7200 seconds of computing time for each instance.

For the affine and the dynamic problems only the Benders formulation is considered. For the static and the volume problems, two approaches are used. The first approach solves compact formulations $(VRNL)$ and $(SRNL)$, enhanced by the separation of robust cutset and 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 cutset inequality for each node of the

name	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert K\rvert$	$\Gamma^{0.25}$	$\Gamma^{0.10}$	$\Gamma^{0.05}$	$\Gamma^{0.01}$
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: Instances description.

network. Then, rounded Benders inequalities are generated at each integer solution and possibly at the root node. In the following, we refer shortly to these two approaches as Compact and Benders, respectively. We generate robust cutset and 3-partition inequalities for both approaches, 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 integer solutions.

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

Notice that for Compact, 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 Compact and for Benders. That is, instead of solving the Benders master problem to integrality at every iteration as in [25], we solve the linear relaxation, as in a traditional branch-and-cut framework. This approach is known to yield much faster algorithms, e.g. [22]. As soon as an inequality is found, the other inequalities are skipped and the LP relaxation is solved again. Another important characterisic of the Benders decomposition algorithm is the primal heuristic provided at the root node. Namely, the LP relaxation is first solved for the static problem using the compact formulation and the resulting capacities are rounded up to the nearest integer values. Altough very small, the solution time of this heuristic is included in the total solution time.

Separation of robust cutset inequalities When the budgeted uncertainty set is used, 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.

$$
\begin{aligned}
\max\ & -\sum_{e\in E} x_e\mu_e + \beta \\
\text{s.t.}\ & \mu_e \geq \max\{r_i - r_j, r_j - r_i\} && e \in E && \text{(22a)}\\
& \mu_e \leq \min\{r_i + r_j, 2 - r_j - r_i\} && e \in E && \text{(22b)}\\
& \nu_k \geq \max\{r_{s(k)} - r_{t(k)}, r_{t(k)} - r_{s(k)}\} && k \in K && \text{(22c)}\\
& \nu_k \leq \min\{r_{s(k)} + r_{t(k)}, 2 - r_{s(k)} - r_{t(k)}\} && k \in K && \text{(22d)}\\
& w_k \leq \min\{\gamma_k, \nu_k\} && k \in K && \text{(22e)}\\
& \beta \leq \sum_{k\in K} \bar{d}_k\nu_k + \sum_{k\in K} \hat{d}^k w_k + 1 - \epsilon && && \text{(22f)}\\
& \sum_{k\in K} \gamma_k = \Gamma \\
& \gamma, w \in \{0,1\}^{|K|}, \mu \in \{0,1\}^{|E|}, \nu \in \{0,1\}^{|K|} \\
& \beta \in \mathbb{Z}, r \in \{0,1\}^{|V|}.
\end{aligned}
$$

Variable r_i is one if node i belongs to set S of the partition and zero otherwise. Variable w_k and constraints (22e) represent product $\gamma_k\mu_k$. Constraints (22a)–(22d) 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.

Separation of robust 3-partition inequalities We separate the 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 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. Is is also possible to develop an exact algorithm based on a generalization of formulation (22). 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 cut inequalities. However, a preliminary testing proved this formulation to be quite slow in practice, therefore we rely on the heuristic approach, as usually usually done in the literature for partition-based inequalities, even for the problem without uncertainty [1].

Separation of Benders-like inequalities Benders inequalities are separated exactly by solving problems (9) and (17) depending on the routing scheme. After finding a violated inequality $\mu^t x \geq b$, it is strengthened by using the following rounding approach [15]. It consists of replacing it by

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

where m is the smallest positive entry of μ. In the unlikely situation where the rounded cut is not violated, we add instead original cut. We note that one could directly separate inequalities with integer μ 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 the strengthening them, is computationally more efficient than solving the integer separation problem.

6 Computational results

This section is organized as follows. We present first an overview of the results in Table 3, paying a particular attention to the cost reductions offered by volume and dynamic routings. Then, we compare the efficiency of the different algorithms for solving each type of routing, but the affine

name	$1-\epsilon$	Solution costs opt_{stat}	$red_{vol}(\%)$	$red_{aff}(\%)$	$red_{dyn}(\%)$	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 3: Overview of the results.

routing because it could only solve one third of the instances. This step is carried out by comparing arithmetic and geometric means of the solution times and the numbers of unsolved instances. It is worth recalling that arithmetic mean gives more weight to hard instances while geometric mean considers equally all instances, regardless of their difficulty. Notice also that this approach hides part of the difficulty of the unsolved instances since their solution times count for 7200 seconds in all computations. Hence, we report in reality lower bounds for the true (unknown) means. This comment is particularly important for dynamic routing which is unable to solve many of the instances. 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 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 A global overview

We provide in Table 3 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 percental 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 $\geq$ since better solutions may exist. 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 hit happened, 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). 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. Due to the time limit, the cost reductions could not be computed exactly in many cases. Nevertheless, the available solutions show that 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. 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 Static routing

Table 4 presents an aggregated comparison of the solution times for the thirteen approaches. Solution times of unsolved instances are set to 7200 seconds. The table shows the significant improvement offered by the 3-partition inequalities: Compact-2 is much faster than Compact-1 and leaves fewer instance unsolved, and 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,

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 4: General comparison of the approaches for static routing.

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 5: General comparison of the approaches for volume routing.

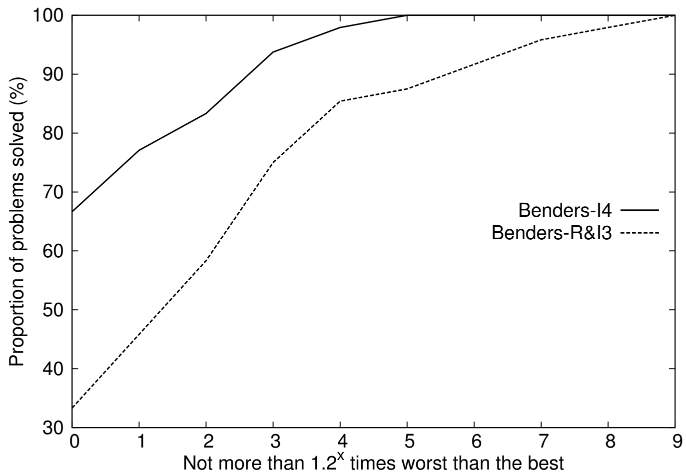


Figure 1: Performance profile comparing Benders-R&I3 and Benders-I4 for volume routing.

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 of that solution algorithm (number of cuts, etc ...) are provided in Table 7 situated in Section 6.8.

6.3 Volume routing

Table 5 is the counterpart of Table 4 for volume routing, presenting an aggregated comparison of the solution times for the thirteen approaches. The table shows again the improvement offered by the 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. Contrasting with the situation observed in Table 4, there is no absolute winner for volume routing. Namely, Benders-R&I3 has 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 are further compared through the performance profile shown in Figure 1, which confirms the modest advantage of Benders-I4 over Benders-R&I3. The full details of Benders-I4 are provided in Table 8 in Section 6.8. Interestingly, compact formulations are less efficient for $(VRNL)$ than they are for $(SRNL)$, which is probably due to the larger numbers of variables and constraints present in $(VRNL)$. Nevertheless, Benders decomposition algorithms seem to perform comparably well for both types of routing. Analyzing tables 7 and 8, 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 a larger with volume routing than with static routing. On the other hand, cut and 3-partition inequalities are tighter for $(VRNL)$ than for $(SRNL)$, see

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 6: General comparison of the approaches for dynamic routing.

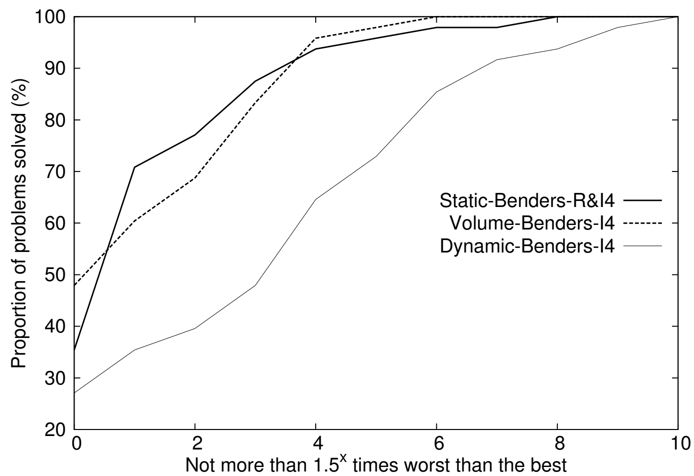


Figure 2: Performance profile comparing Static-Benders-R&I4, Volume-Benders-I4, and Dynamic-Benders-I4.

the discussion in Section 6.7.

6.4 Dynamic routing

Table 6 is the counterpart of tables 4 and 5 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 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 9 situated in Section 6.8.

6.5 Comparing the different routings

We present in Figure 2 a performance profile that compares the best algorithms for the three routings. Figure 2 confirms than the efficiency of the approaches for static routing and volume routing can hardly by ordered. The plot also shows that Dynamic-Benders-I4 is usually slower than the other algorithms.

6.6 Sensitivity to the value of ϵ

We present in Figure 3 the sensitivity of the three best algorithms to the variations of epsilon. 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&4, however, for which larger values of ϵ consistantly yields harder optimization problems.

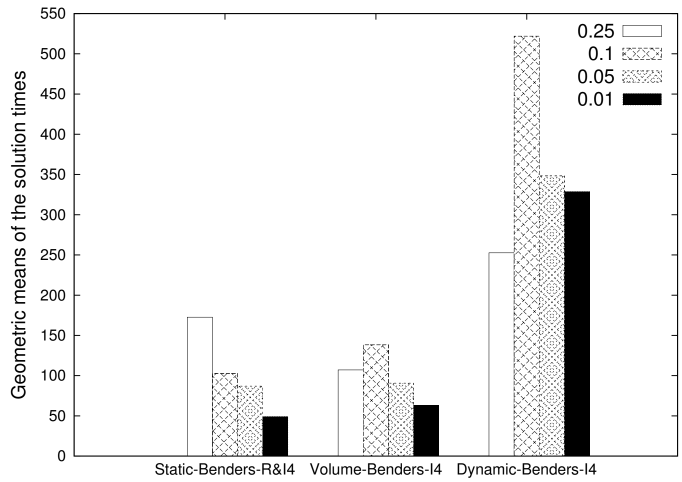
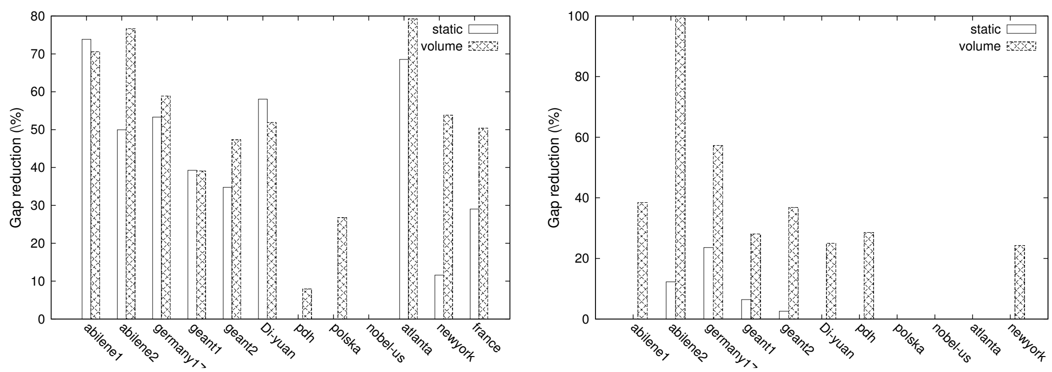


Figure 3: Sensitivity of Static-Benders-R&I4, Volume-Benders-I4, and Dynamic-Benders-I4 to the variations of ϵ.



(a) *RootGap* reduction by cutset inequalities. (b) *CutsetGap* reduction by 3-partition inequalities.

Figure 4: Arithmetic means of the gap closed by the valid inequalities.

6.7 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 $(SRNL)$ or $(VRNL)$, respectively, and let *CutsetGap* and *ThreePartitionGap* be the root gaps obtained after having separated the cutset inequalities and the 3-partition inequalities, respectively. Then, we compute the proportion of the gap closed by the cutset and 3-partition inequalities as

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

We report in Figure 4 the arithmetic means of these gaps taken over the four different values of ϵ. For Figure 4(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, 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 the discussion below Lemma 6. In fact, the only reason for which we witness a few reductions more important for static routing than for volume routing is because these numbers are extracted from our experiments. Hence, since some cutset inequalities are generated heuristically, one may luckily find better inequalities for static routing than for volume routing. A similar argument explains the presence of the negative number set to zero for Figure 4(a): the 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 because all Benders inequalities are rounded, thus biasing the gaps. In contrast, we based Figure 4 on the compact formulations for static and volume routings, which do not involve rounded inequalities.

6.8 Detailed results

Tables 7–9 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, cut inequalities, 3-partition inequalities, and Benders inequalities. Columns *CCuts*, *3PCuts*, and *BCuts* report the number of cutting planes generated, respectively, cut inequalities, 3-partition inequalities, and Benders inequalities; *C3PICuts* further reports the number of cut and 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 3-partition inequalities can be done in a negligible amount of time. Cut 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 9 shows that almost all computational time is spent in the separation of Benders inequalities.

7 Conclusion

The contributions of this paper are essentially numerical, providing interesting insights into the computational tractability of the Robust Network Loading problem with different routing schemes. This study has shed light on the following issues. First, affine routing is hardly tractable, even using decomposition algorithms. More specifically, dynamic routing could solve more instances than affine routing. 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. 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 compact formulations. Third, we show that dynamic routing can be solved for many instances, while others still require very long computational times. Interestingly, our results suggest than dynamic routing is simpler computationally than affine routing. Finally, we confirm the efficiency of robust cutset inequalities and show that the generalization of 3-partition inequalities to the robust context further reduces root gaps and computational times. In particular, we show that, as expected, the gap reductions are more marked for volume routing than for static routing. To conclude, the main message of this paper is that 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.

References

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

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	7200	6647	0	554	170	23	2	47	0	14.7
	0.1	7200	652	0	6530	148	35	2	611	0	14.4
	0.05	7200	7166	0	41	122	5	4	3	0	19.6
	0.01	7200	722	0	6417	129	27	3	609	0	12.7
Arithmetic mean		930	384	0	379	118	27	63	192	313363	1

Table 7: Details of Benders-R&4 for static routing.

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

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

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	7201	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	7200	7122	0	83	114	2	4	2	0	19.9
	0.05	7200	316	0	6881	138	20	3	174	0	19.9
	0.01	7200	312	0	6887	123	27	2	167	0	16.5
Arithmetic mean		1003	215	0	580	103	25	59	123	264513	1

Table 8: Details of Benders-I4 for volume routing.

[4] A. Altin, H. Yaman, and M. Ç. Pinar. The robust network loading problem under hose demand uncertainty: Formulation, polyhedral analysis, and computations. *INFORMS Journal on Computing*, 23(1):75–89, 2011.

[5] R. Andrade, A. Lisser, and N. Maculan. Multi-service multi-facility network design under

name	ϵ	TTime	CTime	3PTime	BTime	CCuts	3PCuts	C3PICuts	BCuts	nodes	endGap
abilene1	0.25	21	3	0	18	55	4	21	0	8	0
	0.1	5	4	0	1	46	8	17	0	3	0
	0.05	5	3	0	1	41	10	11	0	7	0
	0.01	39	2	0	37	49	10	17	0	3	0
abilene2	0.25	117	2	0	115	56	8	10	0	6	0
	0.1	3	2	0	0	50	3	9	0	3	0
	0.05	3	2	0	0	53	1	18	0	9	0
	0.01	54	15	0	38	144	17	20	0	155	0
germany17	0.25	124	14	0	110	123	27	16	0	56	0
	0.1	1673	9	0	1664	123	33	11	0	31	0
	0.05	7183	8	0	7174	119	34	11	0	21	3.6
	0.01	2551	35	0	2512	354	84	165	0	6205	0
geant1	0.25	7188	22	0	7164	209	77	10	0	28	35.4
	0.1	7158	14	0	7142	215	79	29	0	69	32.6
	0.05	7138	24	0	7113	214	52	25	0	38	22.2
	0.01	837	30	0	804	315	67	63	0	736	0
geant2	0.25	7177	20	0	7155	290	79	35	0	84	36.6
	0.1	7185	20	0	7164	246	73	15	0	48	36.2
	0.05	7131	9	0	7120	255	83	21	0	37	36.5
	0.01	4	0	0	3	49	9	15	0	24	0
di-yuan	0.25	19	0	0	18	55	11	9	0	138	0
	0.1	9	0	0	8	56	5	17	0	140	0
	0.05	1	0	0	0	51	8	7	0	61	0
	0.01	12	5	0	7	95	9	8	0	29	0
pdh	0.25	15	3	0	11	95	13	9	0	1583	0
	0.1	3	2	0	0	94	31	12	0	902	0
	0.05	4	3	0	0	106	63	17	0	770	0
	0.01	7200	5	0	7194	61	2	18	1	48	17.9
polska	0.25	7192	3	0	7188	55	0	9	0	2	19.3
	0.1	7193	3	0	7190	47	2	11	0	2	17
	0.05	7193	3	0	7189	54	0	10	0	5	11.5
	0.01	7198	10	0	7187	82	3	7	0	14	17.7
nobel-us	0.25	7200	14	0	7185	72	0	8	0	8	16.3
	0.1	7200	10	0	7189	90	1	12	0	2	15.3
	0.05	7196	8	0	7187	83	5	9	0	11	11.9
	0.01	5228	12	0	5214	94	2	12	0	22	0
atlanta	0.25	7200	4	0	7196	90	5	14	0	7	7.8
	0.1	7196	5	0	7191	87	5	14	0	12	7.8
	0.05	7190	4	0	7185	83	3	11	0	5	6
	0.01	22	10	0	0	373	96	255	0	22435	0
newyork	0.25	32	19	0	0	364	85	257	0	23269	0
	0.1	34	18	0	1	383	101	266	0	27825	0
	0.05	20	8	0	1	401	73	273	0	22244	0
	0.01	7189	73	0	7113	272	22	10	0	54	20
france	0.25	7197	0	0	0	0	0	0	0	0	10
	0.1	7200	0	0	0	0	0	0	0	0	10
	0.05	7200	0	0	0	0	0	0	0	0	10
	0.01	0	0	0	0	0	0	0	0	0	0
Arithmetic mean		3520	10	0	3058	130	27	38	0	2232	8

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

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. *Discrete Optimization*, 4:103–114, 2007.

[7] J. Ayoub and M. Poss. Decomposition for adjustable robust linear optimization subject to

uncertainty polytope, 2013. Minor revision submitted to *Computational Management Science*, optimization-online.org/DB_HTML/2015/11/5207.html.

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

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

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

[12] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

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

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

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

[16] 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.

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

[18] G. Classen, A. Koster, M. Kutschka, and I. Tahiri. Robust metric inequalities for network loading under demand uncertainty. *Asia-Pacific Journal of Operational Research*, 2015. In press.

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

[20] 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.

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

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

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

[28] S. Mattia. The cut property under demand uncertainty. *Networks*. To appear.

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

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

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

[32] 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.

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

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

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

[36] 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.

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

[38] M. Poss. Robust combinatorial optimization with variable budgeted uncertainty. *4OR*, 11(1):75–92, 2013.

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

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

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

[42] 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.

[43] 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.