ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

S. Mattia, M. Poss

EFFICIENT APPROACHES FOR THE ROBUST NETWORK LOADING PROBLEM

R. 4, 2014

Sara Mattia – Istituto di Analisi dei Sistemi ed Informatica, Consiglio Nazionale delle Ricerche, via dei Taurini 19, 00185 Roma, Italy. E-mail: sara.mattia@iasi.cnr.it.

Michael Poss – UMR CNRS 7253 Heudiasyc, Université de Technologie de Compiègne, Centre de Recherches de Royallieu, 60200 Compiègne, France. Email: michael.poss@hds.utc.fr.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

We consider the Robust Network Loading problem with splittable flows and demands that belong to the budgeted uncertainty set. We compare the optimal solution cost and computational cost of the problem when using static routing, volume routing, affine routing, and dynamic routing. For the first three routing types, we compare the compact formulation with a well-engineered Benders decomposition algorithm. For dynamic routing, we use a Benders-based reformulation to generate the extreme points of the uncertainty set on the fly within branch-and-cut algorithms. We test whether it is more efficient to incorporate the extreme points via Benders cuts or via flow variables and constraints, thus mimicking branch-and-cut-and-price algorithms. We test the effect of robust cutset inequalities for all models and algorithms. In addition, we introduce robust three-partition inequalities and show their effectiveness. Our computational experiments are realized on several realistic instances from SNDlib, including some instances based on historical traffic data.

1. Introduction

Given an undirected graph and a set of point-to-point commodities with known demands, 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 is able to route 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 [10] or traffic measurements [34]. 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 [2] and the references therein). 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 random variables compatible with the available data, falling into the framework of distributionally robust optimization (see, for instance, [14]).

In this work, we follow an even safer approach and consider that the uncertain demands are described through an uncertainty set, falling into the framework of robust optimization. Hence, the problem turns to designing a network able to route independently each demand in the uncertainty set. Although conservative, this approach has been used extensively in recent years to model demand uncertainty in telecommunications networks [1, 4, 19, 27] and transportation networks [24, 25]. Since it is extremely unlikely that all commodities reach their peak demands simultaneously, a natural way to model the forecast demands is to use the celebrated budgeted uncertainty set from [8]. 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 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. In 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. In the last years, intermediary routing schemes have been proposed with success, among which affine routing and volume routing, which restrict the flows to affine functions of the demands. In fact, affine routing simply applies to network optimization what has long been known as affine decision rules in adjustable robust optimization, e.g. [7]. Each of these routings has practical advantages and drawbacks, and the choice of the best routing heavily relies on the specific application one has in mind. For instance, static routing is more conservative than dynamic routing but easier to implement in practice than the latter. Notice that other intermediary routings have been considered, such as those based on dynamic partitions of the uncertainty sets [5, 32], but they lead to optimization problems that

are even harder to solve than the problem with dynamic routing (see the discussion in [30]).

The computational tractability of the robust network loading problem has been studied under static, affine, or dynamic routing. For the static routing, [1] propose a polyhedral investigation and numerical results for the problem assuming that the demand uncertainty polytope is the Hose model (independently proposed by [16] and [17]). Similar polyhedral and numerical studies have been carried out by [19] 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. The authors pursued their work in [12] where they propose a Benders decomposition algorithm for the problem. Prior to the work of [12], [20] had already proposed a Benders decomposition of the network loading problem with static routing and budgeted uncertainty. Notice, however, that [20] rely on cutting plane algorithms while [12] use branch-and-cut algorithms. This an an important aspect because recently published result on network design problems (e.g. [11, 18]) suggest that branch-and-cut implementations of Benders decomposition can be orders of magnitude faster than cutting plane implementations. Turning to dynamic routing, [23] studies the problem under the Hose model. She proposes an advanced branch-and-cut procedure related to bilevel optimization. Finally, [24] study the problem with affine routing and polyhedral or ellipsoidal uncertainty sets. They further consider a version of the problem with restrictions on the set of feasible paths and propose column generation algorithms. Other variants of the network loading problems have also been studied in the literature, for instance, by relaxing the integrality restrictions on integer variables (e.g. [31, 6]) or by limiting the number of paths that can be used by each commodity [4].

In this paper, we bridge the gap created by the recent works of [12] and [19] on static routing, [6] and [30] on volume routing, [31] on affine routing, and [23] on dynamic routing. We compare the numerical tractability and the solution cost of these routings on the same instances, with the same uncertainty set, and having spent comparable efforts to tune the decomposition algorithms. For volume and affine routings, our approach can essentially be seen as a generalization of the Benders decompositions proposed by [12] for static routing but using different types of valid inequalities. As [12], we separate robust cutset inequalities. However, we disregard envelope and tight metric inequalities introduced by [12] and [19] because they yield, at best, marginal improvements in the solution times. Instead, we generalize three-partition inequalities from [21] to the robust context. We separate heuristically the resulting robust three-partition inequalities, obtaining interesting speed-ups. For dynamic routing, the exponential numbers of variables and constraints in the formulations prevents us from using directly the classical Benders decomposition algorithm. Hence, we must draw from the more advanced tools developed recently for the robust network loading problem [23] and for more general two-stages robust optimization problems [33, 3].

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. In Section 5, we present our computational experiments, including the specifications of the implemented algorithms. The paper is concluded in Section 6. Detailed computational results can be found in the Appendix.

In what follows we say that mixed-integer linear program (MILP) is *compact* if its numbers of variables and constraints are polynomial functions of the numbers of nodes, edges, and commodities.

2. Models and formulations

Let $G(V, E)$ be an undirected graph without loops and parallel edges, K be the set of point-to-point commodities to be routed on the network, and $\mathcal{U} \subset \mathbb{R}_+^{|K|}$ be the uncertainty polytope introduced by [8] 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 demand vectors $d \in \mathcal{U}$ can be routed non simultaneously 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 simultaneous peak values 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 $\delta(i') = \{e = \{i, j\} \in E : i = i' \text{ or } j = i'\}$.

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

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

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

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

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

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 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 $|K|$. Namely, one readily sees that

$$|\operatorname{vert}(\mathcal{U})| = \sum_{l=1}^{\Gamma} \binom{|K|}{l}.$$

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, see [27, 28, 31]. 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 that are described in [31], among many others.

Proposition 2.1. *Let $\mathcal{U}$ be the polytope described in* (1)*. Then problem (RNL) together with constraints* (3) *can be reformulated as follows:*

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

where z, s, p, q, x, f^0 and y are optimization variables.

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.

Static routing. We enforce f to satisfy

$$f_{ij}^k(d) = y_{ij}^k d^k, \quad k \in K, (i,j) \in A. \tag{5}$$

Static routing is a well-known framework for the robust network loading problem. It has been studied long before affine routing was introduced, see [16, 17], and recent works [1, 19] have extended valid inequalities to network loading problems with static routing. When constraint (5) are satisfied, variables y are often called routing templates because they represent the fractional splittings of demands along the paths from $s(k)$ to $t(k)$ for each $k \in K$. Plugging constraints (5) into (RNL) and applying the techniques mentioned above yields the following reformulation for the problem:

$$\begin{aligned}
(SRNL) \quad \min \; & \sum_{e \in E} c_e x_e \\
\text{s.t.} \; & \sum_{j \in \delta(i)} (y_{ij}^k - y_{ji}^k) = \begin{cases} 1 & \text{if } i = t(k) \\ 0 & \text{otherwise} \end{cases} && i \in V \backslash \{s(k)\}, k \in K && \text{(6a)} \\
& \Gamma z_e + \sum_{k \in K} \left(p_e^k + (y_{ij}^{kk} + y_{ji}^{kk}) \bar{d}^k \right) \leq x_e && e = \{i,j\} \in E && \text{(6b)} \\
& z_e + p_e^k \geq (y_{ij}^{kk} + y_{ji}^{kk}) \hat{d}^k && e = \{i,j\} \in E, k \in K && \text{(6c)} \\
& z, p, y, x \geq 0. && && \text{(6d)} \\
& x \in \mathbb{Z}^{|E|} && && \text{(6e)}
\end{aligned}$$

Formulation (6) 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).

Volume routing. We enforce f to satisfy

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

Volume routing has been originally introduced by [6]. More precisely, [30] shows that the set of routings that satisfy (7) corresponds to the set of *General Volume-Oriented Routings* introduced by [6]. Plugging constraints (7) into (RNL) and applying the techniques mentioned above yields

the following reformulation for the problem:

$$
\begin{aligned}
(VRNL) \quad \min \ & \sum_{e\in E} c_e x_e \\
\text{s.t.} \ & \sum_{j\in\delta(i)} (y^k_{ij} - y^k_{ji}) = \begin{cases} 1 & \text{if } i = t(k) \\ 0 & \text{otherwise} \end{cases} && i \in V\backslash\{s(k)\}, k \in K && \text{(8a)} \\
& \sum_{j\in\delta(i)} (f^{0k}_{ij} - f^{0k}_{ji}) = 0 && i \in V\backslash\{s(k)\}, k \in K && \text{(8b)} \\
& \Gamma z_e + \sum_{k\in K} \left(p^k_e + f^{0k}_{ij} + f^{0k}_{ji} + (y^{kk}_{ij} + y^{kk}_{ji})\bar{d}^k \right) \leq x_e && e = \{i,j\} \in E && \text{(8c)} \\
& z_e + p^k_e \geq (y^{kk}_{ij} + y^{kk}_{ji})\hat{d}^k && e = \{i,j\} \in E, k \in K && \text{(8d)} \\
& f^{0k}_{ij} + y^{kk}_{ij}\bar{d}^k \geq 0 && (i,j) \in A, k \in K && \text{(8e)} \\
& f^{0k}_{ij} + y^{kk}_{ij}(\bar{d}^k + \hat{d}^k) \geq 0 && (i,j) \in A, k \in K && \text{(8f)} \\
& z, p, x \geq 0. && && \text{(8g)} \\
& x \in \mathbb{Z}^{|E|} && && \text{(8h)}
\end{aligned}
$$

Formulation (8) can be derived from formulation (4) by removing flow conservation constraints (4c), replacing robust non-negativity constraints (4f) and (4g) by (8e) and (8f), 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, see [4].

Other types of routings have been considered in the literature. However, they hardly lead to tractable optimization problems [30]. Let opt(P) denote the optimal solution of optimization problem (P). Then, from the above definitions we see immediately than the following ordering holds:

$$\text{opt}(RNL) \leq \text{opt}(ARNL) \leq \text{opt}(VRNL) \leq \text{opt}(SRNL). \tag{9}$$

3. Benders decomposition

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 reformulates the large (but compact) MILP as a MILP that contains only capacity variables. This avoids solving a large linear program at each node of the branch-and-bound tree. Instead, Benders cuts are generated only at the root and at integer nodes. The situation is different for (RNL) since the MILP contains exponentially many variables and constraints. Here the use of Benders decomposition avoids to consider explicitly all vectors in 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 more subtle approach used for problem (RNL).

3.1. Affine routing and simplifications

Benders decomposition is an 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. As usual in network loading problems, 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:

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

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

$$
\begin{aligned}
(FeasAff) \quad & \min \alpha \\
& \text{s.t.} \quad \sum_{j\in\delta(i)} (y^{kk}_{ij} - y^{kk}_{ji}) = 1 && i = t(k), k \in K && \text{(10a)} \\
& \sum_{j\in\delta(i)} (y^{kk}_{ij} - y^{kk}_{ji}) = 0 && i \in V\backslash\{s(k), t(k)\}, k \in K && \text{(10b)} \\
& \Gamma z_e + \sum_{k\in K} \left(p^k_e + f^{0k}_{ij} + f^{0k}_{ji} + \sum_{h\in K} (y^{kh}_{ij} + y^{kh}_{ji})\bar{d}^h \right) \leq x_e + \alpha \quad e = \{i,j\} \in E && && \text{(10c)} \\
& \text{(4b), (4c), (4e), (4f), (4g),}
\end{aligned}
$$

where α represents the maximum of the amounts 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 (10a) and (10c), 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 3.1. *Let $x \in \mathbb{R}^{|E|}$ be given. It holds that $x \in \mathcal{B}^{aff}$ if and only if*

$$-\sum_{e\in E} \mu_e x_e + \sum_{k\in K} \pi^k \leq 0, \tag{11}$$

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 3.1 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. An important property of constraints (11) 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

$$\mathcal{B}^{stat} \equiv \{x \in \mathbb{R}^{|E|} : \exists y \in \mathbb{R}_+^{|A|\times|K|} \text{ that satisfies (6a)} - \text{(6d)}\}.$$

and

$$\mathcal{B}^{vol} \equiv \{x \in \mathbb{R}^{|E|} : \exists f^0 \in \mathbb{R}^{|A|\times|K|} \text{ and } y \in \mathbb{R}^{|A|\times|K|} \text{ that satisfy (8a)} - \text{(8g)}\}.$$

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 3.1, and is therefore, omitted.

Lemma 3.2. *Let $x \in \mathbb{R}^{|E|}$ be given. The following holds:*

- *$x \in \mathcal{B}^{stat}$ if and only if x satisfies inequality (11) for each $(\pi, \mu) \in \text{vert}(\mathcal{D}^{stat})$,*
- *$x \in \mathcal{B}^{vol}$ if and only if x satisfies inequality (11) 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 $\text{vert}(\mathcal{U})$ 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

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

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^u_v, 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 pendant of uncertainty set $\mathcal{U}$ in the problem with aggregated commodities is $\mathcal{U}^{agg} \equiv$

$$\left\{ d \in \mathbb{R}^{|V|\times|V|} \;\middle|\; \exists \delta \in [0,1]^{|K|} \text{ s.t. } \begin{array}{ll} d^u_v = \bar{d}^{k(u,v)} + \delta^{k(u,v)} \hat{d}^{k(u,v)}, & u \neq v \in V \\ d^u_u = -\sum\limits_{v \in V \setminus \{u\}} (\bar{d}^{k(u,v)} + \delta^{k(u,v)} \hat{d}^{k(u,v)}), & u \in V \end{array} , \sum_{k \in K} \delta^k \leq \Gamma \right\}. \tag{13}$$

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

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

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 (e.g. [3, 33]) to (RNL).

Proposition 3.3. *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:

$$\max \; -\sum_{e\in E}\mu_e x_e + \sum_{u\neq v\in V}(\bar{d}^{k(u,v)}\pi_v^u + \hat{d}^{k(u,v)}\rho_v^u) \tag{15a}$$

$$\begin{aligned}
s.t. \quad & \rho_v^u \le \delta^{k(u,v)} && u\neq v\in V && (15b)\\
& \rho_v^u \le \pi_v^u && u\neq v\in V && (15c)\\
& \rho_v^u \ge \pi_v^u + \delta^{k(u,v)} - 1 && u\neq v\in V && (15d)\\
& \sum_{k\in K}\delta^k \le \Gamma && k\in K && (15e)\\
& \pi_i^u - \pi_j^u \le \mu_{\{i,j\}} && a=(i,j)\in A, u\in V && (15f)\\
& \sum_{e\in E}\mu_e \le 1 && && (15g)\\
& \delta\in\{0,1\}^{|K|} && && (15h)\\
& \mu,\pi\ge 0 && && (15i)\\
& \rho\ge 0. && && (15j)
\end{aligned}$$

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

$$\begin{aligned}
(Feas\text{-}d) \quad & \min \alpha\\
& \text{s.t.} \quad \sum_{j\in\delta(i)}(f_{ij}^u - f_{ji}^u) = d_v^u \qquad u\neq i\in V, && (16a)\\
& \qquad \sum_{u\in V}(f_{ij}^u + f_{ji}^u) \le x_e + \alpha \quad e=\{i,j\}\in E, && (16b)\\
& \qquad f,\alpha\ge 0. && (16c)
\end{aligned}$$

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

$$\max_{d\in\mathcal{U}^{agg}} \quad \begin{aligned}\min\ & \alpha\\ \text{s.t. } & \text{(16a)} - \text{(16c)}\end{aligned} \tag{17}$$

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 (17). Let π denote the dual variables of constraints (16a), μ denote the dual variables of constraints (16b). Dualizing the inner minmization problem of (17), we obtain the following bilinear program:

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

$$\begin{aligned}
\text{s.t.} \quad & \sum_{k\in K}\delta^k \le \Gamma && k\in K && (18a)\\
& \pi_i^u - \pi_j^u \le \mu_{\{i,j\}} && a=(i,j)\in A, u\in V && (18b)\\
& \sum_{e\in E}\mu_e \le 1 && && (18c)\\
& \delta\in[0,1]^{|K|} && && (18d)\\
& \mu,\pi\ge 0. && && (18e)
\end{aligned}$$

One sees easily that the optimal solution of problem (18) is reached at some binary vector δ, so that we can replace (18d) 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 $u \neq v \in V$. ∎

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. For other uncertainty polytopes, one is left with the reformulation of [23], based on complementary slackness conditions. However, the latter can hardly be solved for the instances considered in [23], while our numerical experiments show that the reformulation from Proposition 3.3 can be solved quite efficiently. 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 pendant of Lemma 3.1 for (RNL).

Lemma 3.4. *Let $x \in \mathbb{R}^{|E|}$ be given and let $\mathcal{D}$ be the polytope defined by constraints* (15f), (15g), *and* (15i). *It holds that $x \in \mathcal{B}$ if and only if*

$$-\sum_{e\in E} \mu_e x_e + \sum_{u\neq v\in V} d^{k(u,v)}\pi_v^u \leq 0, \tag{19}$$

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

4. Valid inequalities

Benders cuts (11) and (19) describe the projections on variables x of the feasibility sets of the linear relaxations of (RNL), $(ARNL)$, $(SRNL)$, and $(VRNL)$. However, these cuts 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 propose in what follows valid inequalities for $\mathcal{B}_I$, $\mathcal{B}_I^{aff}$, $\mathcal{B}_I^{stat}$, and $\mathcal{B}_I^{vol}$, which may not necessarily be valid for $\mathcal{B}$, $\mathcal{B}^{aff}$, $\mathcal{B}^{stat}$, and $\mathcal{B}^{vol}$. We present next a simple lemma, useful when describing valid inequalities for the aforementioned sets.

Lemma 4.1. *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 [23]. 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, comparing the gap closed for $\mathcal{B}_I^{stat}$ and $\mathcal{B}_I^{vol}$. From the theoretical point of view, an inequality can have a different facet-defining status according the considered polyhedron. This considerations are formalized in the corollary below. With a little abuse of notation, we say that routing scheme r_h is included in routing scheme r_k ($r_h \subseteq r_k$) if any solution that is feasible for r_h is also feasible for r_k. We also say that an inequality is valid (facet-defining) for r_h if is valid (facet-defining) for the polyhedron corresponding to routing r_h.

Corollary 4.2. *Let $\alpha^T x \leq \beta$ be a valid inequality for routing r_i. The following hold:*

1. *$\alpha^T x \leq \beta$ is a valid inequality for routing r_j, for any $r_j \subseteq r_i$;*
2. *if $\alpha^T x \leq \beta$ is not facet-defining for r_i, it is not facet-defining for r_j, for any $r_j \subseteq r_i$.*

4.1. Cutset inequalities

Consider a partition of V given by sets S and $\bar{S}$, and let E_S and K_S be the set of edges and commodities with extremities in different sets of the partition. The cutset inequality associated with partition $\{S, \bar{S}\}$ states that the amount of capacity installed on edges in E_S should be not less that the rounded up sum of the demands of commodities in K_S. In the deterministic network design problem where $\mathcal{U}$ is reduced to a singleton, the inequality can be written as

$$\sum_{e \in E_S} x_e \geq \left\lceil \sum_{k \in K_S} d^k \right\rceil. \tag{20}$$

One readily sees that the robust version of cutset inequality (20) is valid for $\mathcal{B}_I$:

$$\sum_{e \in E_S} x_e \geq \left\lceil \sum_{k \in K_S} \bar{d}^k + \max_{Q_S \subseteq K_S, |Q| \leq \Gamma} \sum_{k \in Q_S} \hat{d}^k \right\rceil. \tag{21}$$

Corollary 4.2 implies that cutset inequalities are also valid for $\mathcal{B}_I^{aff}$, $\mathcal{B}_I^{stat}$, and $\mathcal{B}_I^{vol}$. Moreover, it is possible to prove that they are facet defining for all the considered routing schemes (see [23] for the proof).

We separate cutset inequalities (21) with 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 MIP:

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

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.

4.2. Three-partition inequalities

Consider a partition of V given by sets S_1, S_2, and S_3, and let $\bar{S}_i = V \backslash S_i$ for $i = 1, 2, 3$. The robust three-partition inequality associated to the partition is obtained by summing the three cut inequalities associated to S_i and $\bar{S}_i$ 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 (21) with $S = S_i$ for $i = 1, 2, 3$, and let $E(S_1, S_2, S_3)$ be the set of edges with extremities in different sets of the partition. The robust three-partition inequality is

$$\sum_{e \in E(S_1, S_2, S_3)} x_e \geq \left\lceil \frac{1}{2}(rhs_1 + rhs_2 + rhs_3) \right\rceil . \tag{23}$$

Inequality (23) is valid for $\mathcal{B}_I$ since it is obtained by conic-combination and rounding of cutset inequalities, and Corollary 4.2 implies that it is also valid for $\mathcal{B}_I^{aff}$, $\mathcal{B}_I^{stat}$, and $\mathcal{B}_I^{vol}$. Unfortunately, it is not a facet for the dynamic problem and then, by Corollary 4.2, it is not a facet for any routing scheme. Consider three-node problem with with demands $\bar{d}_{12} = \bar{d}_{13} = \bar{d}_{23} = 0.2$, deviations $\delta_{12} = \delta_{13} = \delta_{23} = 0.1$ and $\Gamma = 1$. Then $rhs_1 = rhs_2 = rhs_3 = 1$ and the right-and-side of the three-partition inequality is $p = 2$. For the inequality to be a facet, there must exist four affinely independent integer feasible solutions satisfying it with equality. In this case, feasibility can be tested using only cutset inequalities, as this problem has the cut property (see [22]). Let the variables be ordered lexicographically, the only feasible capacity vectors satisfying the three-partition inequality with equality are $[1, 1, 0]$, $[1, 0, 1]$, $[0, 1, 1]$ and hence the three-partition inequality cannot be a facet. Interestingly, this is the first case of an inequality being facet-defining for the deterministic problem but not for the robust problem.

We separate the three-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.

4.3. Rounded Benders inequalities

We illustrate the rounding approach on Benders cut (19); one can readily adapt the following to Benders cut (11). Whenever μ is integer, Benders cut

$$\sum_{e \in E} \mu_e x_e \geq \sum_{u \neq v \in V} d^{k(u,v)} \pi_v^u \tag{24}$$

can be replaced by the stronger inequality

$$\sum_{e \in E} \mu_e x_e \geq \left\lceil \sum_{u \neq v \in V} d^{k(u,v)} \pi_v^u \right\rceil . \tag{25}$$

Inequality (25) can be separated by adding integrality restrictions on μ and a constraint similar to (22f) to the separation problem from Proposition 3.3. However, the approach turns out to

be numerically inefficient. Instead, we separate Benders inequalities in their non-rounded form, divide each side of the resulting inequality (24) by $min = \min_{e \in E} \mu_e$, and round the coefficients as follows,

$$\sum_{e \in E} \left\lceil \frac{\mu_e}{min} \right\rceil x_e \geq \left\lceil \sum_{u \neq v \in V} \frac{d^{k(u,v)} \pi_v^u}{min} \right\rceil, \tag{26}$$

following a procedure that dates back to [9]. In the unlikely situation where the rounded cut is not violated, we add instead original cut (24).

5. Computational experiments

The purpose of the computational experiments presented in this section is two-fold. First, and more importantly, we compare the solvability of (RNL), $(SRNL)$, and $(VRNL)$ using different algorithms and settings. Results for $(ARNL)$ are not presented because none of our instances could be solved within the time limit of 7200 seconds (and no upper bound was available). Second, we compare the optimal solution cost of (RNL), $(SRNL)$, and $(VRNL)$ on realistic instances.

5.1. Context

Configuration. Solution approaches for $(VRNL)$, and $(SRNL)$ have been coded in JAVA using Cplex Concert Technology, while solution approaches for (RNL) have been coded in C using Cplex Callable library. All computations were run on a computer equipped with an Intel(R) Xeon(R) CPU E5-2670 2.60GHz processor and 132 GB of RAM, using CPLEX 12.5 [13]. We allow 7200 seconds of computing time for each instance.

Instances. We test our models and algorithms on twelve realistic network instances available from SNDlib [26]. The main characteristics of these networks are reminded in Table 1. Networks *abilene*, *germany*, and *geant* come from [19], 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 [19] 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 done in recent papers (e.g. [31, 4]).

Protection. We choose the value of Γ according to the probabilistic bound introduced by [8]. Namely, we set three levels of probabilistic guaranty (denoted ϵ): $0.25, 0.10$, and 0.05. Then, for each value of ϵ, the bound from [8] 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 than demands exceeds the capacity installed on arc a is less than ϵ. Notice that the bound from [8] can be be over-conservative (see [29]). For that reason, and to avoid fractional values of Γ, we rounded down the resulting values of Γ, denoted by $\Gamma^{0.25}, \Gamma^{0.10}$, and $\Gamma^{0.05}$, respectively.

5.2. Algorithms

We describe next the algorithms that have been implemented to solve the problems.

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

Table 1: Instances description.

Affine routing and simplifications. Two approaches have been implemented for $(ARNL)$, $(VRNL)$, and $(SRNL)$. Affine routing is not mentioned in what follow because none of the approach to $(ARNL)$ would provide even a feasible solution within the time limit. The first approach solves compact formulations $(VRNL)$ and $(SRNL)$, enhanced by the separation of cutset and three-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 network. Then, rounded Benders inequalities are generated at the root node and at each integer solution. In the following, we refer shortly to these two approaches as Compact and Benders, respectively. We generate cutset and three-partition inequalities for both approaches, according to one of the following configurations:

0 No cutset or three-partition inequalities.

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

2 1+ heuristic separation of three-partition inequalities at the root node only.

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

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

Notice that for Compact, only the first three implementations are tested. For each of these configuration, the inequalities are separated in this order: 1) heuristic separation of cutset inequalities, 2) heuristic separation of three-partition inequalities, 3) exact separation of cutset inequalities, 4) separation of benders inequalities (only for Benders decomposition). As soon as an inequality is found, the other inequalities are skipped and the LP relaxation is solved again.

Dynamic routing. No compact formulation exists for (RNL) so that the problem is solved solely by decomposition algorithms, based either on the capacity formulation or on the flow formulation, which we denote shortly by Capacity or Flow, respectively. Capacity is almost identical to the Benders decomposition algorithm described above, the only difference being how Benders inequalities are separated. Flow starts from a relaxed version of formulation

(RNL) that contains only capacity and flow conservation constraints corresponding to a subset $\mathcal{U}^*$ of vert($\mathcal{U}$), possibly empty. The algorithm then looks for cutset, three-partition, and Benders inequalities similarly to Capacity. However, when a Benders inequality is found (corresponding to $\xi^* \in \text{vert}(\mathcal{U})$), the full set of flow variables and capacity and flow conservation constraints corresponding to ξ^* are added to the formulation. Otherwise, we only add the violated inequality. Strictly speaking, this approach falls into branch-and-cut-and-price algorithms. However, to use the efficient MIP solver of CPLEX, we implemented a branch-and-cut version of this algorithm. Hence, dummy variables are created in the initial problem, which are gradually changed to flow variables by adding the appropriate constraints. Whenever all dummy variables have been used, the algorithm continues by adding only Benders inequalities. Notice that in our numerical experiments, it never happens that the number of preallocated dummy variables is exceeded. In contrast with the approaches for affine routing and its simplifications, we test whether it is best to generate metric inequalities only at integer solutions (I) or at integer solutions and at the root node (A). Hence, we compare 20 approaches for dynamic routing: Capacity-Am, Capacity-Im, Flow-Am, and Flow-Im, for each $m \in \{0, 1, 2, 3, 4\}$.

5.3. Results

This section is organized as follows. We present first an overview of the results in Table 2, 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. 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 sensibility to the value of ϵ and study the gap closed by the valid inequalities.

Overview. We provide in Table 2 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} and red_{dyn} report the percental decrease in solution costs with volume 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 five columns provide insights into the computational difficulty of the optimization problems. Columns *Best solution time* provide the solution time in seconds of the best approach (reporting **T** if no algorithm could solve the instance due to the time limit) and columns *Best gap* provide the best gap at the of the execution if no approach could solve the problem within the time limit.

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 fair because we compare different algorithms for different types of routing and instances. A more rigorous 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 14

		Solution costs			Best solution time			Best gap	
name	ϵ	opt_{stat}	$red_{vol}(\%)$	$red_{dyn}(\%)$	*stat*	*vol*	*dyn*	*vol*(%)	*dyn*(%)
	0.25	3.100E+01	0.0	0.0	1	3	9	0	0
abilene1	0.1	3.200E+01	0.0	0.0	1	3	5	0	0
	0.05	3.300E+01	0.0	0.0	2	2	1	0	0
	0.25	2.000E+01	5.0	5.0	2	2	11	0	0
abilene2	0.1	2.200E+01	0.0	0.0	2	2	4	0	0
	0.05	2.200E+01	0.0	0.0	2	4	1	0	0
	0.25	3.500E+01	2.9	2.9	52	18	31	0	0
germany17	0.1	3.600E+01	0.0	0.0	24	29	132	0	0
	0.05	3.600E+01	0.0	0.0	14	37	14	0	0
	0.25	3.000E+01	0.0	≥ 0.0	106	455	**T**	0	28
geant1	0.1	3.100E+01	3.2	≥ 0.0	1650	287	**T**	0	29
	0.05	3.100E+01	3.2	≥ 0.0	115	403	**T**	0	29
	0.25	3.400E+01	2.9	2.9	478	248	344	0	0
geant2	0.1	3.400E+01	0.0	0.0	119	305	743	0	0
	0.05	3.400E+01	0.0	≥ 0.0	212	261	**T**	0	6
	0.25	5.241E+06	8.5	9.6	69	4	4	0	0
di-yuan	0.1	5.367E+06	1.9	3.7	8	111	7	0	0
	0.05	5.371E+06	1.1	2.8	6	23	5	0	0
	0.25	8.506E+05	4.8	6.5	600	186	5	0	0
pdh	0.1	8.523E+05	0.1	0.6	47	148	9	0	0
	0.05	8.526E+05	0.0	0.1	9	17	4	0	0
	0.25	2.612E+02	12.4	≥ 0.0	33	33	**T**	0	18
polska	0.1	2.874E+02	12.8	≥ 0.0	42	30	**T**	0	18
	0.05	2.935E+02	10.9	≥ 0.0	681	143	**T**	0	17
	0.25	2.949E+05	10.5	≥ 0.0	29	330	**T**	0	17
nobel-us	0.1	3.156E+05	9.2	≥ 0.0	33	928	**T**	0	17
	0.05	3.198E+05	7.9	≥ 0.0	137	294	**T**	0	15
	0.25	2.001E+05	4.7	5.4	25	150	43	0	0
atlanta	0.1	2.096E+05	3.4	3.6	45	104	572	0	0
	0.05	2.117E+05	2.7	2.7	188	42	3494	0	0
	0.25	9.852E+02	0.0	0.0	35	720	26	0	0
newyork	0.1	9.852E+02	0.0	0.0	42	450	33	0	0
	0.05	9.852E+02	0.0	0.0	46	458	30	0	0
	0.25	1.040E+01	7.7	≥ 0.0	5164	3321	**T**	0	17
france	0.1	1.100E+01	6.4	≥ 0.0	1259	2157	**T**	0	16
	0.05	1.120E+01	≥ 5.4	≥ 0.0	247	**T**	**T**	1	16

Table 2: Overview of the results.

instances out of 36, which ranges up to 12.8 %. In most cases, the reduction costs is higher when the protection level is high (and thus, Γ is small). Dynamic routing is, as expected, harder to solve than the two others routings and 13 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 to the time, the cost reductions could not be computed exactly in many cases. Nevertheless, the available solutions show that dynamic routing improves over volume routing

Formulation	Compact			Benders				
Valid inequalities	0	1	2	0	1	2	3	4
Arithmetic mean	2492	2170	1394	2536	2374	2127	1354	**1315**
Geometric mean	278	241	**131**	316	368	281	190	149
Unsolved	10	8	4	9	7	**3**	4	4

Table 3: General comparison of the approaches for static routing.

Figure 1: Performance profile comparing Compact-2, Benders-3, and Benders-4 for static routing.

Formulation	Compact			Benders				
Valid inequalities	0	1	2	0	1	2	3	4
Arithmetic mean	3610	3081	2361	2729	2651	2563	1291	**1203**
Geometric mean	1031	566	298	498	395	337	158	**140**
Unsolved	16	13	10	12	12	10	5	**3**

Table 4: General comparison of the approaches for volume routing.

Figure 2: Performance profile comparing Benders-3, and Benders-4 for volume routing.

by up to 1.8 additional percent (reached for *di-yuan*, $\epsilon = 0.1$).

Static routing. Table 3 presents an aggregated comparison of the solution times for the eight approaches. Solution times of unsolved instances are set to 7200 seconds. The table shows the significant improvement offered by the three-partition inequalities: Compact-2 is much faster than Compact-1 and leaves fewer instance unsolved, and the comparison of Benders-1 and Benders-2 yields similar comments. Benders-4 also behaves better than Benders-3, but the improvement is less important.. It is clear from that table that there is no absolute winner: Benders-4 has the best arithmetic mean, Compact-2 has the best geometric mean, and Benders-2 solves more instances than the others. We refine the comparison of the table by selecting the three fastest approaches that have at most 4 unsolved instances: Compact-2, Benders-3, and Benders-4. Then, Figure 1 compares these three approaches through a performance profile [15]. The profile confirms than Compact-2 and Benders-4 have comparable efficiencies. The profile also shows that Benders-3 seems a bit worse than the two other approaches. The full details of Compact-2 and Benders-4 are provided in tables 6 and 7 located in Appendix A.

Volume routing. Table 4 is the pendant of Table 3 for volume routing, presenting an aggregated comparison of the solution times for the eight approaches. The table shows again the constant improvement offered by the three-partition inequalities. In contrast with Table 3, we see from Table 4 that Benders-4 seems to be the best algorithm for volume routing for the three indicators, followed closely by Benders-3. Benders-3 and Benders-4 are further compared through the performance profile shown in Figure 2, which confirms the modest advantage of Benders-4 over Benders-3. The full details of Benders-4 are provided in Table 8 in Appendix A. 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 $(VRNL)$.

Formulation	Capacity									
Inequalities	A0	A1	A2	A3	A4	I0	I1	I2	I3	I4
Arithmetic mean	2985	2786	2517	2742	2578	2183	2056	**1765**	2129	2090
Geometric mean	350	313	194	318	205	129	123	**89**	142	98
Unsolved	17	16	16	16	16	14	14	**13**	14	14
Formulation	Flow									
Inequalities	A0	A1	A2	A3	A4	I0	I1	I2	I3	I4
Arithmetic mean	3125	2566	2486	2621	2503	3865	2452	2531	2455	2499
Geometric mean	413	227	189	240	201	574	170	174	181	175
Unsolved	20	19	18	19	18	26	18	18	17	18

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

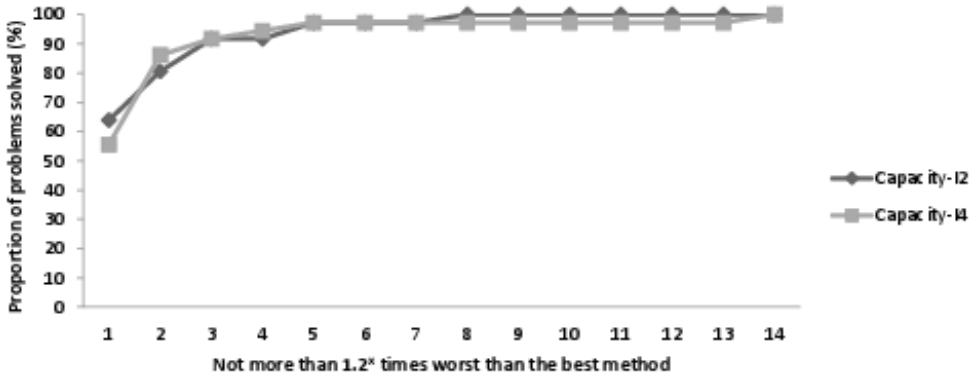


Figure 3: Performance profile comparing Capacity-I2 and Capacity-I4 for dynamic routing.

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-4 loses more time generating Benders cuts than Static-Benders-4. On the other hand, cut and three-partition inequalities are tighter for $(VRNL)$ than for $(SRNL)$, see the discussion below.

Dynamic routing. Table 5 is the pendant of tables 3 and 4 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 three-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 Capacity-I2, followed closely by Capacity-I4. Hence, differently from the previous approaches, it is better to generate cutset and three-partition inequalities at the root node only, while Benders inequalities are generated at integer solutions only. We purse our comparison between Capacity-I2 and Capacity-I4 through the performance profile shown in Figure 3, which confirms the very small advantage of Capacity-I2 over Capacity-I4. The full details of Capacity-I2 are provided in Table 9 in Appendix A.

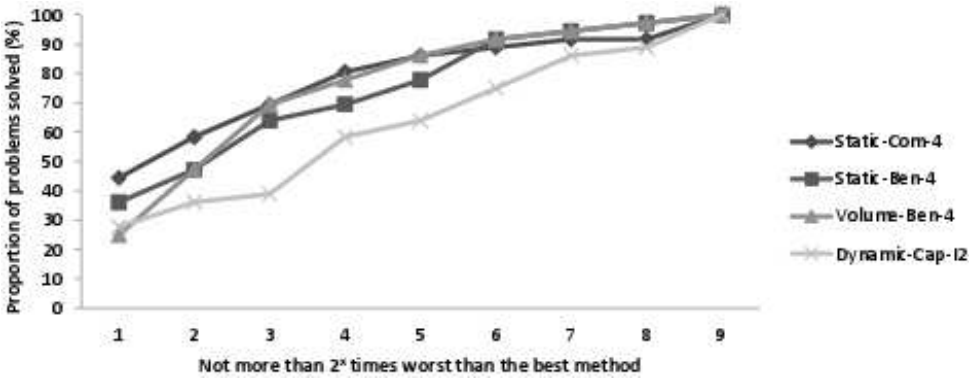


Figure 4: Performance profile comparing Static-Compact-2, Static-Benders-4, Volume-Benders-4, and Dynamic-Capacity-I2.

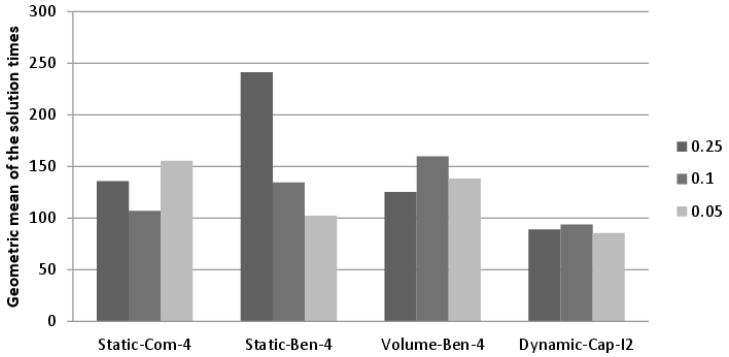
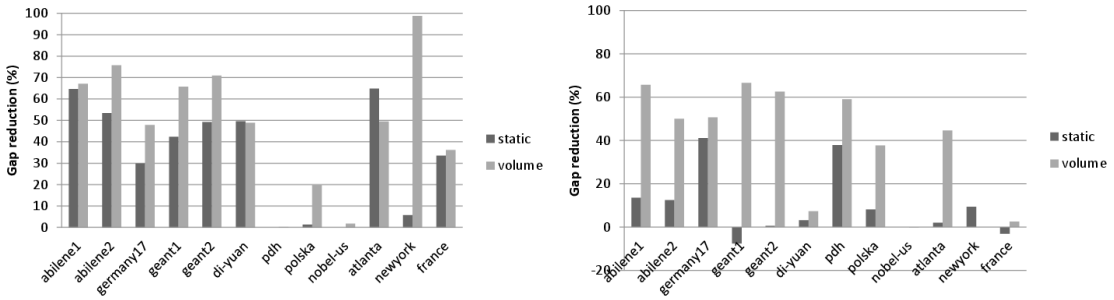


Figure 5: Sensibility of Static-Compact-2, Static-Benders-4, Volume-Benders-4, and Dynamic-Capacity-I2 to the variation of ϵ.

Comparing the different routings. We present in Figure 4 a performance profile that compares the best algorithms for the three routings. Figure 4 confirms than the efficiency of the approaches for static routing and volume routing can hardly by ordered. The plot also shows that Dynamic-Capacity-I2 is usually slower than the other algorithms.

Sensibility to the value of ϵ. We present in Figure 5 the sensibility of the four best algorithms to the variations of epsilon. The figure shows that the geometric means of the solution times for Static-Compact-2, Volume-Benders-4, and Dynamic-Capacity-I2 are not monotonically impacted by the value of ϵ. The results are different for Static-Benders-4, however, for which larger values of ϵ yields harder optimization problems.

Gap closed by the inequalities. We study next the effect of the robust cutset and three-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 three-partition inequalities, respectively. Then, we



(a) *RootGap* reduction by cutset inequalities.

(b) *CutsetGap* reduction by three-partition inequalities.

Figure 6: Arithmetic means of the gap closed by the valid inequalities.

compute the proportion of the gap closed by the cutset and three-partition inequalities as

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

respectively. We report in Figure 6 the arithmetic means of these gaps taken over the three different values of ϵ. For network *france* and volume routing, we disregard the value $\epsilon = 0.25$ because its optimal solution is unknown. Two important conclusions can be drawn from the figures. First, three-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 4.1. 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 negative numbers in figreffig:rootgap:3P: 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 6 on the compact formulations for static and volume routings, which do not involve rounded inequalities.

6. 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 as such, even using well engineered decomposition algorithms. 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 three-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.

As a side-product, we also show that two-stages robust optimization problems with first-stage integer variables may not benefit much from incorporating new extreme points via blocks of constraints and variables present in the original problem rather than Benders cuts. Indeed, our results show that solving (RNL) through the Capacity formulation is faster than through the Flow formulation. This observation contrasts with the recent findings of [33] who suggest, at the contrary, that row-and-column generations algorithms are order of magnitude faster than pure row generation algorithms. We think that the difference between our results is mainly due to two reasons. First, [33] solve MILP master problems at each iteration while we use branch-and-cut algorithms. Second, and more importantly, our approach relies heavily on strong valid inequalities (cutset and three-partition inequalities), many of them being separated heuristically. The heuristical separation of these cuts avoids the difficult separation of Benders inequalities, known to be the bottleneck of both approaches.

References

[1] 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 Computing*, vol. 23, no. 1, pp. 75–89, 2011.

[2] R. Andrade, A. Lisser, and N. Maculan, "Multi-service multi-facility network design under uncertainty," *Annals of Operations Research*, vol. 199, no. 1, pp. 157–178, 2012.

[3] J. Ayoub and M. Poss, "Decomposition for adjustable robust linear optimization subject to uncertainty polytope," 2013. Submitted.

[4] F. Babonneau, J.-P. Vial, O. Klopfenstein, and A. Ouorou, "Robust capacity assignment solutions for telecommunications networks with uncertain demands," *Networks*, vol. 62, no. 4, pp. 255–272, 2013.

[5] W. Ben-Ameur, "Between fully dynamic routing and robust stable routing," in *6th International Workshop on Design and Reliable Communication Networks, 2007. DRCN 2007*, 2007.

[6] W. Ben-Ameur and M. Zotkiewicz, "Volume oriented routing," in *14th International Telecommunications Network Strategy and Planning Symposium (NETWORKS)*, pp. 1–7, 2010.

[7] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski, "Adjustable robust solutions of uncertain linear programs," *Math. Program.*, vol. 99, no. 2, pp. 351–376, 2004.

[8] D. Bertsimas and M. Sim, "The price of robustness," *Oper. Res.*, vol. 52, no. 1, pp. 35–53, 2004.

[9] D. Bienstock, S. Chopra, O. Günlük, and C.-Y. Tsai, "Minimum cost capacity installation for multicommodity network flows," *Math. Program.*, vol. 81, pp. 177–199, 1998.

[10] 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*, pp. 107–116, 2008.

[11] Q. Botton, B. Fortz, L. Gouveia, and M. Poss, "Benders decomposition for the hop-constrained survivable network design problem," *INFORMS Journal on Computing*, vol. 25, no. 1, pp. 13–26, 2013.

[12] G. Classen, A. Koster, M. Kutschka, and I. Tahiri, "Robust metric inequalities for the gamma-robust network loading problem." Available at Optimization Online, 2013.

[13] CPLEX, *IBM ILOG CPLEX 12.5 Reference Manual*. ILOG CPLEX Division, Gentilly, France, 2013.

[14] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Operations Research*, vol. 58, no. 3, pp. 595–612, 2010.

[15] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

[16] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merive, "A flexible model for resource management in virtual private networks," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 4, pp. 95–108, 1999.

[17] J. A. Fingerhut, S. Suri, and J. S. Turner, "Designing least-cost nonblocking broadband networks," *Journal of Algorithms*, vol. 24, no. 2, pp. 287 – 309, 1997.

[18] B. Fortz and M. Poss, "An improved benders decomposition applied to a multi-layer network design problem," *Oper. Res. Lett.*, vol. 37, no. 5, pp. 359–364, 2009.

[19] A. Koster, M. Kutschka, and C. Raack, "Robust network design: Formulations, valid inequalities, and computations," *Networks*, vol. 61, no. 2, pp. 128–149, 2013.

[20] C. Lee, K. Lee, and S. Park, "Benders decomposition approach for the robust network design problem with flow bifurcations," *Networks*, vol. 62, no. 1, pp. 1–16, 2013.

[21] T. L. Magnanti, P. Mirchandani, and R. Vachani, "The convex hull of two core capacitated network design problems," *Math. Program.*, vol. 60, pp. 233–250, 1993.

[22] S. Mattia, "The cut condition for robust network design," in *Proc. of INOC 2013*, vol. 41 of *ENDM*, pp. 303–310, 2013.

[23] S. Mattia, "The robust network loading problem with dynamic routing," *Comput. Optim. Appl.*, vol. 54, no. 3, pp. 619–643, 2013.

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

[25] F. Ordóñez and J. Zhao, "Robust capacity expansion of network flows," *Networks*, vol. 50, no. 2, pp. 136–145, 2007.

[26] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.

[27] A. Ouorou, "Tractable approximations to a robust capacity assignment model in telecommunications under demand uncertainty," *Computers & OR*, vol. 40, no. 1, pp. 318–327, 2013.

[28] A. Ouorou and J.-P. Vial, "A model for robust capacity planning for telecommunications networks under demand uncertainty," in *Proceedings of the 6th International Workshop on Design and Reliable Communication Networks (DRCN 2007)*, pp. 1–4, 2007.

[29] M. Poss, "Robust combinatorial optimization with variable budgeted uncertainty," *4OR*, vol. 11, no. 1, pp. 75–92, 2013.

[30] M. Poss, "A comparison of routing sets for robust network design," *Optimization Letters*, vol. 8, no. 5, pp. 1619–1635, 2014.

[31] M. Poss and C. Raack, "Affine recourse for the robust network design problem: Between static and dynamic routing," *Networks*, vol. 61, no. 2, pp. 180–198, 2013.

[32] M. G. Scutellà, "On improving optimal oblivious routing," *Operations Research Letters*, vol. 37, no. 3, pp. 197–200, 2009.

[33] B. Zeng and L. Zhao, "Solving two-stage robust optimization problems using a column-and-constraint generation method," *Oper. Res. Lett.*, vol. 41, no. 5, pp. 457–461, 2013.

[34] 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*, pp. 206–217, 2003.

A. Detailed results

We describe next tables 6–9. Columns *TTime*, *CTime*, *3PTile*, and *BTime* report the total solution times and the time to generate, respectively, cut inequalities, three-partition inequalities, and Benders inequalities. Columns *CCuts*, *3PCuts*, and *BCuts* report the number of cutting planes generated, respectively, cut inequalities, three-partition inequalities, and Benders inequalities; *C3PICuts* further reports the number of cut and three-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 make next some comments about the computational times presented in the tables. We see that separating three-partition inequalities can be done in a negligible amount of time. Cut inequalities take more time to separate, even heuristically. We further see that *CTime* is much larger in tables 7 and 8 than in Table 6. This is due to the time taken by the exact

separation of cutset inequalities. Indeed, since cutset inequalities are implied by the compact formulation and since we separate them in their non-rounded form, the exact separation is never used for the compact formulation while it is used in Benders decomposition algorithms. For all benders decomposition algorithms, we see that separating Benders cuts takes a large amount 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.

name	ϵ	TTime	CTime	3PTime	CCuts	3PCuts	endGap	nodes
	0.25	1	0	0	0	7	0	0
abilene1	0.1	2	1	0	1	6	0	0
	0.05	2	1	0	1	7	0	0
	0.25	5	3	0	3	8	0	31
abilene2	0.1	2	1	0	2	13	0	0
	0.05	6	4	0	4	7	0	0
	0.25	58	11	0	6	32	0	354
germany17	0.1	42	10	0	6	36	0	72
	0.05	38	13	0	5	39	0	40
	0.25	1762	28	0	8	43	0	2866
geant1	0.1	**T**	27	0	6	60	4.1	6466
	0.05	5823	30	0	7	53	0	2256
	0.25	1484	18	0	6	59	0	2456
geant2	0.1	1034	11	0	5	86	0	589
	0.05	1539	20	0	6	57	0	612
	0.25	71	0	0	0	2	0	17165
di-yuan	0.1	11	0	0	1	5	0	1812
	0.05	6	0	0	1	5	0	703
	0.25	645	1	0	0	3	0	117038
pdh	0.1	181	1	0	0	26	0	10770
	0.05	130	1	0	0	31	0	4867
	0.25	35	3	0	0	0	0	6615
polska	0.1	56	3	0	0	2	0	10738
	0.05	1303	3	0	0	1	0	203776
	0.25	35	7	0	0	0	0	4074
nobel-us	0.1	40	7	0	0	0	0	3664
	0.05	146	8	0	0	0	0	11085
	0.25	25	11	0	4	3	0	696
atlanta	0.1	45	9	0	2	3	0	1454
	0.05	188	10	0	2	4	0	6695
	0.25	**T**	5	0	0	42	9.5	9400
newyork	0.1	**T**	7	0	0	55	11	6394
	0.05	**T**	8	0	1	55	9.3	10951
	0.25	5164	77	0	2	14	0	28951
france	0.1	1259	34	0	1	10	0	4129
	0.05	247	44	0	1	12	0	379
Arithmetic mean		1394	12	0	2	22	1	13253

Table 6: Details of Compact-2 for static routing.

name	ϵ	TTime	CTime	3PTime	BTime	CCuts	3PCuts	C3PICuts	BCuts	endGap	n
	0.25	3	2	0	0	19	3	9	0	0	
abilene1	0.1	5	4	0	0	19	3	11	0	0	
	0.05	2	2	0	0	25	3	16	1	0	
	0.25	7	6	0	0	22	3	15	11	0	
abilene2	0.1	4	3	0	0	22	7	9	2	0	
	0.05	3	2	0	0	28	5	7	0	0	
	0.25	52	43	0	8	80	22	8	39	0	
germany17	0.1	26	22	0	3	59	33	7	10	0	
	0.05	21	17	0	3	61	26	12	3	0	
	0.25	126	99	0	23	243	90	155	8	0	3
geant1	0.1	1650	288	0	1345	457	51	374	383	0	2
	0.05	115	59	0	54	185	70	76	16	0	2
	0.25	478	184	0	286	368	56	181	149	0	5
geant2	0.1	119	52	0	64	232	60	121	18	0	1
	0.05	212	75	0	134	227	72	119	38	0	1
	0.25	94	9	0	7	33	13	6	136	0	31
di-yuan	0.1	13	5	0	5	31	17	5	53	0	8
	0.05	13	4	0	4	35	6	12	41	0	1
	0.25	2530	261	0	48	46	6	9	537	0	31
pdh	0.1	96	73	0	16	52	15	18	127	0	1
	0.05	29	24	0	4	72	44	21	28	0	2
	0.25	2846	1412	0	50	51	1	48	1383	0	17
polska	0.1	480	397	0	14	47	11	10	375	0	21
	0.05	933	380	0	20	50	8	3	363	0	19
	0.25	**T**	6836	0	172	111	8	110	1774	0.4	17
nobel-us	0.1	2425	2224	0	48	82	13	1	596	0	28
	0.05	3929	3589	0	106	87	11	3	976	0	39
	0.25	483	439	0	30	54	11	7	313	0	3
atlanta	0.1	785	670	0	76	39	14	9	461	0	8
	0.05	935	757	0	115	47	21	8	536	0	12
	0.25	35	13	0	8	291	69	286	0	0	3
newyork	0.1	42	16	0	5	356	54	340	0	0	5
	0.05	46	19	0	12	343	86	328	0	0	2
	0.25	**T**	4357	0	2820	148	9	138	539	6.7	1
france	0.1	**T**	7162	0	44	138	7	7	2	none	
	0.05	**T**	4662	0	2521	143	36	3	553	none	
Arithmetic mean		1315	949	0	223	120	27	69	263	0	24

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

name	ϵ	TTime	CTime	3PTime	BTime	CCuts	3PCuts	C3PICuts	BCuts	endGap	nodes
	0.25	4	2	0	1	22	1	12	0	0	3
abilene1	0.1	4	3	0	1	23	3	12	0	0	2
	0.05	4	2	0	1	19	7	10	0	0	5
	0.25	3	2	0	1	24	4	6	0	0	2
abilene2	0.1	2	1	0	1	21	2	9	0	0	0
	0.05	4	2	0	1	17	4	12	0	0	2
	0.25	18	6	0	11	57	21	6	0	0	77
germany17	0.1	30	9	0	21	57	35	14	0	0	101
	0.05	37	12	0	24	61	20	6	0	0	29
	0.25	455	50	0	401	261	75	167	0	0	7199
geant1	0.1	287	11	0	273	244	26	252	0	0	6425
	0.05	403	23	0	379	121	57	37	1	0	468
	0.25	248	24	0	223	156	49	70	0	0	601
geant2	0.1	305	23	0	278	323	22	307	0	0	7466
	0.05	261	15	0	244	200	44	114	0	0	1743
	0.25	11	2	0	9	27	10	9	24	0	333
di-yuan	0.1	148	11	0	67	29	8	4	174	0	262859
	0.05	40	5	0	29	34	14	10	62	0	23551
	0.25	255	99	0	61	40	17	7	222	0	275208
pdh	0.1	196	100	0	88	61	26	19	189	0	19714
	0.05	30	17	0	12	67	40	27	18	0	1576
	0.25	78	63	0	13	37	4	5	63	0	8150
polska	0.1	152	127	0	22	40	8	2	119	0	18486
	0.05	431	310	0	93	44	2	43	311	0	124533
	0.25	2457	1906	0	283	64	7	5	545	0	761840
nobel-us	0.1	6290	3836	0	689	71	3	5	1167	0	2401578
	0.05	7153	5245	0	848	69	5	4	1484	0	1299635
	0.25	158	86	0	71	46	4	11	60	0	1686
atlanta	0.1	148	76	0	71	34	4	16	48	0	608
	0.05	49	23	0	25	44	6	11	11	0	65
	0.25	993	18	0	958	385	77	377	0	0	41846
newyork	0.1	450	21	0	422	257	69	235	0	0	18688
	0.05	608	14	0	583	305	61	293	0	0	28068
	0.25	**T**	815	0	6383	123	11	5	76	3.7	127
france	0.1	**T**	3029	0	4174	128	13	4	21	none	45
	0.05	**T**	275	0	6928	124	16	4	20	none	40
Arithmetic mean		1203	452	0	658	101	22	59	128	0	147577

Table 8: Details of Benders-4 for volume routing.

name	ϵ	TTime	CTime	3PTime	BTime	CCuts	3PCuts	BCuts	endGap	nodes
	0.25	10	1	0	9	27	9	1	0	7
abilene1	0.1	9	3	0	6	31	3	0	0	10
	0.05	3	2	0	0	31	2	5	0	16
	0.25	11	2	0	9	21	10	0	0	5
abilene2	0.1	15	2	0	13	26	7	3	0	7
	0.05	1	1	0	0	28	6	2	0	6
	0.25	74	7	0	66	60	14	23	0	624
germany17	0.1	132	17	0	114	76	22	14	0	421
	0.05	388	8	0	380	70	20	36	0	505
	0.25	**T**	47	0	7141	78	26	242	31	6535
geant1	0.1	**T**	27	0	7165	81	37	170	31	2100
	0.05	**T**	88	0	7100	88	16	180	30	5916
	0.25	1564	55	0	1502	108	16	189	0	4362
geant2	0.1	743	19	0	715	67	18	334	0	7139
	0.05	**T**	26	0	7172	89	13	155	30	2129
	0.25	5	1	0	4	54	13	5	0	79
di-yuan	0.1	7	3	0	4	65	17	1	0	163
	0.05	5	4	0	1	65	18	0	0	30
	0.25	5	1	0	4	43	12	0	0	89
pdh	0.1	15	0	0	14	59	15	6	0	247
	0.05	4	2	0	1	67	27	4	0	443
	0.25	**T**	8	0	7192	57	12	3	18	14
polska	0.1	**T**	8	0	7192	68	4	5	19	20
	0.05	**T**	13	0	7187	59	9	0	18	5
	0.25	**T**	47	0	7153	84	9	2	17	20
nobel-us	0.1	**T**	51	0	7149	88	7	1	17	17
	0.05	**T**	49	0	7150	80	9	3	17	22
	0.25	43	18	0	25	67	10	1	0	16
atlanta	0.1	626	16	0	610	56	7	2	0	33
	0.05	6151	18	0	6133	61	12	4	0	55
	0.25	34	2	0	8	14	14	229	0	25531
newyork	0.1	33	2	0	7	14	13	213	0	27107
	0.05	36	2	0	8	13	15	234	0	26283
	0.25	**T**	238	0	6954	155	22	3	18	128
france	0.1	**T**	326	0	6856	140	32	17	17	1165
	0.05	**T**	300	0	6883	167	23	4	17	1847
Arithmetic mean		2883	39	0	2831	65	14	58	8	3142

Table 9: Details of Capacity-I2 for dynamic routing.