Robust network design with uncertain outsourcing cost

Artur Alves Pessoa
Production Engineering Department, Fluminense Federal University, Rua Passo da Pátria 156, 24210-240 Niteroí, RJ, Brazil, artur@producao.uff.br

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

The expansion of a telecommunications network faces two sources of uncertainty, which are the demand for traffic that shall transit through the expanded network and the outsourcing cost that the network operator will have to pay to handle the traffic that exceeds the capacity of its network. The latter is determined by the future cost of telecommunications services, whose negative correlation with the total demand is empirically measured in the literature through the price elasticity of demand . Unlike previous robust optimization works on the subject, we consider in this paper both sources of uncertainty and the correlation between them. The resulting mathematical model is a linear program that exhibits a constraint with quadratic dependency on the uncertainties. To solve the model, we propose a decomposition approach that avoids considering the constraint for all scenarios. Instead, we use a cutting plane algorithm that generates required scenarios on the fly by solving linear multiplicative programs. Computational experiments realized on the networks from SNDlib show that our approach is orders of magnitude faster than the classical semidefinite programming reformulation for such problems.

Key words: Robust optimization; Network design; Random recourse; Linear multiplicative programming; Convex optimization

1. Introduction

Given a directed graph and a set of point-to-point commodities with known demand values, the classical network design problem looks for the cheapest capacity installation able to route all demands simultaneously. However, the hypothesis that all demand values are known at the time at which the network expansion is planned is not realistic. In practice, future demands are estimated by using, for instance, traffic measurements or population statistics. Using single forecasted values for the demands leads to overestimation of the traffic and unnecessary installation costs. Rather, one should construct a large set of demand vector values that contain most of plausible outcomes for these demands. The network design problem turns then to design a network able to route non-simultaneously each demand vector in the above set.

Many research papers have been published on network design problems under demand uncertainty. Part of these papers considers that demand vectors are described by precisely characterized random variables, see for instance Andrade et al. (2006). However, many authors have pointed out that this assumption is not realistic in the context of telecommunications network design because the historical data necessary to compute the probabilities is not available. For this reason, most of the recent papers on the topic rather consider the framework of robust optimization, where future demand vectors are only supposed to belong to predetermined bounded and convex sets, often called uncertainty sets. Such uncertainty set is usually assumed to be a polytope or the intersection of a box and an ellipsoid. We refer the interested reader to Ben-Tal et al. (2009); Bertsimas et al. (2011) for detailed surveys of robust optimization.

The main novelty of this paper lies in the use of uncertain outsourcing cost for unmet demands. In the literature, authors have commonly studied problems involving two conflicting objectives. The first objective considers the cost of expanding the capacity of the existing network, which involves the price of the equipment as well as the installation costs, such as digging conducts into the ground. The second objective is related to the amount of unmet demands. These objectives are conflicting since a network expanded with a large amount of capacity is likely to cope with a larger proportion of demands than a network expanded with a small amount of capacity. Hence, the first network would have a higher capacity cost and a lower demand outsourcing cost than the second network. Research papers on the topic diverge on how to model the combination of the two objectives. We are aware of two approches to address this modeling issue. In the first approach , the authors consider separately the two objectives, that is, one objective is turned into a hard constraint by assuming that a fixed budget is available for expanding the network (Lemaréchal et al., 2010) , or by limiting the amount of unmet demands (Ouorou, 2013) . In the second approach, the authors rather consider one objective function made up by summing the two costs (Andrade et al., 2006; Sen et al., 1994) . In any case, one needs to introduce an explicit outsourcing cost for unmet demands. The model considered in this paper differs from the aforementioned models by a crucial detail: we suppose that the outsourcing cost is not known with precision and can take any value in a predetermined uncertainty set. This assumption complicates significantly the resulting optimization models, adding to the objective function a term that is quadratic in the uncertain parameters. Nevertheless, modeling outsourcing costs as uncertain is an important practical consideration because of the price elasticity of demand observed in the telecommunications market (e.g. Agiakloglou and Yannelis (2006)). Namely, a decrease in prices tends to lead to an increase in demand. Such a price decrease is often imposed by regulatory agencies to avoid the use of market power by companies to the customer's detriment (Garbacz and Thompson Jr (2007)). In this case, it likely that both outsourcing and end-user costs are equally affected to avoid price distortions that could favor some segments of the industry over others, as pointed out by Garbacz and Thompson Jr (2007).

More generally, for many optimization problems it is not realistic to model the cost of recourse variables by known parameters. In fact, the consideration of uncertain penalty cost had been motivated a long time ago in the context of stochastic programming in Garstka (1973):

> Randomness in the second-stage (recourse) costs can arise quite naturally in two ways.

> Firstly, and most naturally, the decision maker may be unable to specify the recourse costs with complete certainty. Secondly, and perhaps more interesting, random recourse costs may reflect uncertainty in the length of the first stage. At any point in time, recourse costs may be able to be specified with certainty. However, if these costs change with time, then knowing only probabilistically when the first stage will end is equivalent to specifying it in a probabilistic manner.

The revival of affine decision rules in multistage robust optimization by Ben-Tal et al. (2004) has led to a considerable interest in these adjustable models, and many authors have seeked to extend them to more complex decision rules and have applied them to practical decision problems. Robust network design models in particular have witnessed a particularly high interest in affine decision rules, called affine routing in that context, see Ouorou and Vial (2007); Poss and Raack (2013). Surprisingly enough, very little attention has been devoted to the situation where recourse costs are uncertain.

It is well known that adjustable robust linear optimization problems are $\mathcal{NP}$-hard in general (Ben-Tal et al., 2004) and problems involving uncertain recourse costs are particularly difficult. When all coefficients of the adjustable variables are fixed (fixed recourse), the classical approach relies on using affine decision rules. Whenever the uncertainty set is defined by conic constraints, such as second-order cone inequalities or linear inequalities, the resulting optimization problems can be reformulated as second-order cone programs and linear programs, respectively, by using duality theory for convex optimization. The situation is more complex when the fixed recourse assumption is not fulfilled because this leads to constraints with quadratic dependency on the uncertainty. In particular, the above dualization techniques are no longer applicable. The only known convex reformulation to such problems is due to Ben-Tal et al. (2002) and relies on the S-lemma (Pólik and Terlaky, 2007) : the robust constraints with quadratic dependency on the uncertain vectors are reformulated as linear matrix inequalities (LMI). The S-lemma approach suffers from the following two drawbacks. First, it is an exact approach only when the uncertain set is an ellipsoid. To handle uncertainty sets defined by intersections of ellipsoids, such as polytopes (when the ellipsoids are degenerated), the procedure only provides an approximated solution, and its number of variables increases linearly with the number of ellipsoids. Second, and more importantly, LMIs lead to semidefinite programming which cannot easily handle large-scale optimization problems. This is particularly important in this paper since network design problems involve large numbers of flow variables and flow conservation constraints.

In this paper, we introduce an alternative approach based on cutting plane algorithms. Instead of considering all demand vectors simultaneously in the robust model, we decompose the latter into a master problem that contains the cuts associated to some particular vectors in the uncertainty set, and a set of separation problems that check if more cuts should be added to the master problem. Such decomposition procedures have been proposed in the recent years for robust optimization, see for instance Fischetti and Monaci (2012); Monaci et al. (2013). The results in these papers show that, for some problems, decomposition can outperform by far the classical dualization. The choice between dualization and decomposition has also been studied from a machine learning perspective in the work of Bertsimas et al. (2014). Still, these techniques have not yet been applied to robust optimization problems that involve constraints with quadratic dependency on the uncertainties. As we show in this paper, the resulting separation problems become linear multiplicative programs, which are non-convex optimization problems that need to be handled adequately. We focus in this paper on problems with fractional capacities to be able to compare our approach to the classical SDP reformulation. If the capacities or the flow needed to be modeled through integer variables, our algorithms would involve mixed-integer linear programs (instead of a mixed-integer SDP). Hence, our algorithms would be more likely to cope efficiently with integer variables than approaches based on non-linear formulations. This is yet another advantage of our approach since branch-and-cut algorithms based on linear formulations have more scientific maturity than the emergent branch-and-bound algorithms based on SDP formulations.

We highlight the following main contributions of this paper:

- We introduce a new model for telecommunications network design under demand uncertainty where the uncertainty of outsourcing cost is considered since these costs are not known with precision at the time the planning decisions must be made.
- We propose cutting-plane algorithms to solve the resulting robust optimization problems. Our algorithms are not restricted to ellipsoidal uncertainty and outperform the SDP reformulation by orders of magnitude.
- We prove that our optimization problem, where the uncertain part of the outsourcing cost is the same for each commodity, is polynomially solvable. However, the problem would become strongly $co\mathcal{NP}$-hard if the outsourcing cost of distinct commodities were allowed to depend on different uncertainties.

We outline next the structure of the paper. The next section presents the mathematical formulation of the problem and situates the work within the robust network design literature. The following two sections describe two different approaches for solving the problem. Section 3 applies the tools from the robust optimization literature to provide conic reformulations for the robust constraints with linear dependency on the uncertainties. Section 4 proposes decomposition algorithms to handle resource constraints with linear and quadratic dependency on the uncertainties. The section concludes by proving that the problem can be solved in polynomial time. In Section 5, we present a generalization of the problem where the uncertain part of the outsourcing costs can be different for each commodity and we prove that the resulting problem is strongly $co\mathcal{NP}$-hard. In Section 6, we present extensive numerical experiments that compare SDP reformulations with decomposition algorithms for the problem on realistic networks. Finally, a small conclusion is provided in Section 7.

2. Mathematical formulation

We provide in this section a mathematical programming formulation for the problem studied in the paper. We denote the directed graph by $G = (V, A)$ and a the set of point-to-point commodities by K. For each node $v \in V$, let $\delta^+(v)$ and $\delta^-(v)$ denote the sets of outgoing arcs and incoming arcs at node v, respectively. For each $a \in A$, let $c_a \in \mathbb{R}_+$ denote the unitary capacity cost and $u_a \in \mathbb{R}_+$ denote the initial capacity. For each commodity $k \in K$, we denote its source and destination by $s(k) \in V$ and $t(k) \in V$, respectively. Each demand value d^k varies between its nominal value $\overline{d}^k > 0$ and its peak value $\overline{d}^k + \hat{d}^k$ to which we subtract $\alpha\xi^0$ where ξ^0 is the (uncertain) normalized deviation of telecommunication services price and $\alpha > 0$ represents the price dependency of the demand. More precisely, we suppose for each $k \in K$ that

$$d^k(\xi) = \overline{d}^k(1 - \alpha\xi^0) + \hat{d}^k\xi^k, \tag{1}$$

where ξ can take any value in set Ξ, that is closed, convex and bounded. Such a set Ξ is called an uncertainty set in the following.

The problem is modeled with the help of three types of optimization variables. For each $a \in A$, a continuous capacity variable $x_a \geq 0$ represents the amount of capacity installed on a. For each $a \in A$ and $k \in K$, a continuous flow variable $f_a^k \geq 0$ represents the fraction of commodity k routed on arc a. Finally, for each $k \in K$, outsourcing variable g^k represents the fraction of commodity k that is not served by the network. Rejection variables are redundant since they satisfy $g^k = 1 - \sum_{a\in\delta^-(t(k))} f_a^k + \sum_{a\in\delta^+(t(k))} f_a^k$; we use them to simplify the presentation of the model.

The main contribution of this work lies in the consideration of uncertain outsourcing cost r. We suppose that each component $r^k(\xi)$ is defined by the affine function $\overline{r}^k(1+\beta\xi^0)$, where ξ^0 is the normalized deviation of services price as before, $\beta > 0$ represents the relation between the outsourcing cost and the price, and $\overline{r}^k > 0$ is a constant factor that depends on commodity k which can be defined, for instance, from the distance between $s(k)$ and $t(k)$. Hence, the price elasticity of demand is equal to $-\alpha/\beta$. The objective function of our problem aims thus at minimizing the capacity installation cost plus the worst-case cost of outsourcing the unmet demands:

$$\min_{x,f,g} \sum_{a\in A} c_a x_a + \max_{\xi\in\Xi} \sum_{k\in K} r^k(\xi) d^k(\xi) g^k. \tag{2}$$

Notice that for arbitrary uncertainty set Ξ, nothing prevents the second term from Eq. (2) to reach its maximum at $\xi^* \in \Xi$ such that $r^k(\xi^*) < 0$ and $d^k(\xi^*) < 0$ for some $k \in K$, which would be unrealistic. To make sure this unrealistic situation never happens, we assume in the following that Ξ is symmetric: $\xi \in \Xi$ implies that $-\xi \in \Xi$ for any $\xi \in \mathbb{R}^K$. One readily sees that, under this assumption, the maximum is always reached at $\xi^* \in \Xi$ such that $r^k(\xi^*) > 0$ and $d^k(\xi^*) > 0$ for each $k \in K$.

In the formulation below, we reformulate Eq. (2) through an epigraph reformulation by introducing an auxiliary variable θ that represents the outsourcing cost.

$$(RND) \quad \begin{array}{lll} \min & \displaystyle\sum_{a \in A} c_a x_a + \theta & \\ \text{s.t.} & \displaystyle\theta \geq \sum_{k \in K} r^k(\xi) d^k(\xi) g^k & \xi \in \Xi \qquad (3) \\ & \displaystyle\sum_{a \in \delta^-(v)} f_a^k - \sum_{a \in \delta^+(v)} f_a^k = 0, & k \in K, v \in V \backslash \{s(k), t(k)\} \qquad (4) \\ & \displaystyle\sum_{k \in K} d^k(\xi) f_a^k \leq u_a + x_a & a \in A, \xi \in \Xi \qquad (5) \\ & \displaystyle g^k = 1 - \sum_{a \in \delta^-(t(k))} f_a^k + \sum_{a \in \delta^+(t(k))} f_a^k & k \in K \qquad (6) \\ & f, x, g \geq 0. & \qquad (7) \end{array}$$

In the above formulation, constraints (3) define the outsourcing cost, constraints (4) are the flow conservation constraints, constraints (5) impose that the total flow on each arc must not exceed the available capacity, and constraints (6) define the outsourcing variables. Notice that term $\sum_{a \in \delta^+(t(k))} f_a^k$ of constraints (6) is necessary to prevent the apparition of unrealistic cycle-flows. Formulation (RND) is a semi-infinite linear programming formulation since there are infinite numbers of constraints (3) and (5). We propose in Sections 3 and 4 two approaches to solve (RND) exactly. The first consists in applying known robust optimization results to reformulate constraints (3) and (5) as finite number of conic constraints. The second relaxes these constraints and proposes cutting plane algorithms to recover the solution feasibility.

In this paper, we follow the literature on robust network design and suppose that the uncertainty set is one of the following: a symmetric polytope, the intersection of an ellipsoid and a box, or an ellipsoid. The polytope model supposes that Ξ is described by a given system of linear inequalities. Different polytopes have been used in the literature to model uncertainty on the demand for robust network design problems. The most common ones are the budget uncertainty set (Poss and Raack, 2013; Ouorou, 2013) and the Hose model see Altin et al. (2011) and the references therein. The second uncertainty set we consider has been used by Babonneau et al. (2013), among others, and defines Ξ as

$$\{\xi \mid \|\xi\|_\infty \leq 1, \|\xi\|_2 \leq \kappa_2\},$$

where κ_2 is a given constant. In order to compare our cutting plane algorithm to the semidefinite programming reformulation from the literature (Ben-Tal et al., 2002), we will also use the purely ellipsoidal model

$$\{\xi \mid \|\xi\|_2 \leq \kappa_2\}.$$

We finish this section by situating (RND) in the vast literature on robust network design. We use the so-called static routing in (RND) where the flow on $a \in A$ for $k \in K$ is given by $d^k(\xi)f_a^k$, which is a linear function of d^k. This routing framework is largely used in the literature, see Altin et al. (2011); Koster et al. (2013) and the references therein. At the opposite of static routing stands dynamic routing which assumes that there is no relation between the routing used for two different demand vectors d_1 and d_2, that is, the flow for each $a \in A$ and $k \in K$ becomes an arbitrary function of d (Mattia, 2013). Intermediary routing schemes have also been considered, see Poss (2013a) and the references therein, among which affine routing, introduced by Ouorou and Vial (2007), stands out for offering a good balance between computational complexity and flexibility. In this paper, we stick to static routing rather than affine routing to keep the presentation as simple as possible and better focus on the implications of considering uncertain outsourcing costs. Nevertheless, one can easily see that all ideas developed in the following can be extended to affine routing, and the computational complexity of the resulting optimization problems is the same as in the case of static routing.

Other models require to introduce integer variables. The aforementioned routing frameworks suppose that the flow for each commodity can be split among an arbitrary number of paths, which is not always a realistic assumption. Alternative models bound the number of paths that can be used simultaneously to route a demand by a small number (Babonneau et al., 2013), which could be equal to one when bifurcations of the flow is not allowed (Lee et al., 2012). The capacity on each link of the network could also be modeled by a finite set of discrete modules, each having different characteristics (Altin et al., 2011; Koster et al., 2013). Each of these technicalities leads to integer programming formulations, which are significantly harder to solve than their fractional counterparts. In theory, the different solution approaches presented in this paper could be applied to refinements of (RND) with integer variables, by using the appropriated branch-and-bound and branch-and-cut algorithms. Namely, the approaches based on conic reformulations could be embedded into a branch-and-bound algorithm with non-linear relaxation, while the decomposition approach would lead to a branch-and-cut algorithm with linear relaxation. In practice however, the decomposition approach is likely to cope more easily with integer variables because branch-and-cut solvers based on liner formulations have more scientific maturity than the emerging branch-and-bound solvers based on conic formulations.

3. Conic reformulation

We apply in this section classical reformulation results of robust optimization (Ben-Tal et al., 2009) to provide a conic reformulation for (RND). Hence, the results of this section are presented without proof. The main success of robust optimization is largely due to the fact that the infinite number of constraints that appear in robust problems can often be reformulated as finite numbers of conic constraints, or even linear constraints when Ξ is a polyhedron. When the constraints of the original problem are linear in the problem variables and uncertain parameters, these reformulations are essentially based on the duality theory for conic optimization. However, if the original constraints involve non-linearities, one must use more subtle techniques.

Constraint (3) has a quadratic dependency on Ξ preventing us from applying the duality-based reformulations. Nevertheless, a different approach can be used when Ξ is an ellipsoid. Rather than using duality theory, the approach is based on another result from convex programming: the S-lemma. We provide in this section conic reformulations for constraints (3) and constraints (5) as conic constraints.

3.1 Outsourcing cost

We study first how to reformulate constraint (3). To highlight the quadratic nature of the constraint, it is useful to rewrite the latter by grouping its terms according to their degree in ξ:

$$\begin{aligned} & \theta \geq \sum_{k\in K} \overline{r}^k g^k (1+\beta\xi^0)(\overline{d}^k - \alpha\overline{d}^k\xi^0 + \hat{d}^k\xi^k) && \xi\in\Xi \\ \Leftrightarrow\quad & 0 \leq \alpha(g) + 2\xi^T\beta(g) + \xi^T\Gamma(g)\xi && \xi\in\Xi, \end{aligned} \tag{8}$$

where

- $\alpha(g) = -\sum_{k\in K} \overline{r}^k\overline{d}^k g^k + \theta$,
- $\beta^k(g) = -\frac{\overline{r}^k\hat{d}^k g^k}{2}$ for each $k\in K$, $\beta^0(g) = -\frac{\beta-\alpha}{2}\sum_{k\in K}\overline{r}^k\overline{d}^k g^k$,
- $\Gamma^{kh}(g) = 0$ for each $k,h\in K$, $\Gamma^{0k}(g) = -\frac{\beta\overline{r}^k\hat{d}^k g^k}{2}$ for each $k\in K$, and $\Gamma^{00}(g) = \alpha\beta\sum_{k\in K}\overline{r}^k\overline{d}^k g^k$.

Problem (RND) contains an infinite number of constraint (8). A natural approach to handle them would seek to separate the constraints and rely on cutting plane algorithms. This approach is described in Section 4. In this section, we would like to consider a compact reformulation for constraint (8). Unfortunately, such a reformulation is known only when Ξ is an ellipsoid, see Ben-Tal et al. (2002) for details.

Theorem 1. *Robust constraint*

$$0 \leq \alpha(g) + 2\xi^T\beta(g) + \xi^T\Gamma(g)\xi \textit{ for all } \xi\in\Xi = \{\xi \mid \|\xi\|_2 \leq \kappa_2\}$$

is equivalent to the following LMIs in g, and v

$$\begin{pmatrix} \Gamma(g) + \kappa_2^{-2} v \,\mathrm{Id} & \beta(g) \\ \beta^T(g) & \alpha(g) - v \end{pmatrix} \succeq 0 \qquad (9)$$
$$v \geq 0,$$

where Id *is the* $(|K|+1) \times (|K|+1)$ *identity matrix.*

3.2 Capacity constraints

In this section, we consider constraints (5). These constraints involve only affine functions of ξ, so that they can be rewritten as follows:

$$\begin{aligned} & \sum_{k\in K} \hat{d}^k f_a^k \xi^k + \sum_{k\in K} \overline{d}^k f_a^k - \alpha \xi^0 \sum_{k\in K} \overline{d}^k f_a^k \leq u_a + x_a && a \in A, \xi \in \Xi \\ \Leftrightarrow \quad & \xi^T \mu_a(f) \leq \nu_a(f,x) && a \in A, \xi \in \Xi, \qquad (10) \end{aligned}$$

where

- $\mu_a^k(f) = \hat{d}^k f_a^k$ for each $a \in A, k \in K$, $\mu_a^0(f) = -\alpha \sum_{k\in K} \overline{d}^k f_a^k$ for each $a \in A$, and
- $\nu_a(x,f) = u_a + x_a - \sum_{k\in K} \overline{d}^k f_a^k$ for each $a \in A$.

It is well known in robust optimization that duality theory for convex optimization yields a direct reformulation for constraints (10) when Ξ is described by conic constraints, see (Ben-Tal et al., 2009, Theorem 1.3.4) . Next, we recall the specific reformulation that we obtain for the different types of uncertainty sets considered in this paper.

Theorem 2. *Let $a \in A$ be fixed and Ξ be an uncertainty set, and consider robust constraint*

$$\xi^T \mu_a(f) \leq \nu_a(f,x) \text{ for all } \xi \in \Xi. \qquad (11)$$

The following holds:

1. *If $\Xi = \{\xi \mid \|\xi\|_2 \leq \kappa_2\}$, then constraint* (11) *is equivalent to the following constraints in x and f:*

$$\kappa_2 \|\mu_a(f)\|_2 \leq \nu_a(f,x). \qquad (12)$$

2. *If $\Xi = \{\xi \mid \|\xi\|_\infty \leq 1, \|\xi\|_p \leq \kappa_p\}$ with $p = 1$ or $p = 2$, then constraint* (11) *is equivalent to the following constraint in x, f, and w:*

$$\kappa_p \|\mu_a(f) - w\|_q + \|w\|_1 \leq \nu_a(f,x), \qquad (13)$$

with $q = \infty$ if $p = 1$ and $q = 2$ if $p = 2$.

3. If $\Xi = \{\xi \mid B\xi \leq b\}$, then constraint (11) *is equivalent to the following constraints in x, f, and w:*

$$\begin{array}{l} b^T w \leq \nu_a(f,x) \\ B^T w = \mu_a(f) \\ w \geq 0 \end{array} \tag{14}$$

Theorem 2 enables us to solve (RND) with mathematical programming solvers for the cases where constraint (3) can be reformulated conveniently. As we have seen in Section 3.1, the latter constraint can be reformulated as a LMI when Ξ is an ellipsoid. Hence, to obtain a pure semidefinite programming reformulation for (RND) in the case of ellipsoidal uncertainty, we should also reformulate constraints (12) as LMIs. This can be done using the following basic property of semidefinite positiveness.

Lemma 1. *For any vector $\lambda \in \mathbb{R}^{|K|}$ and positive scalar $\lambda^0 \in \mathbb{R}_+$, the following holds:*

$$\|\lambda\|_2 \leq \lambda_0 \qquad \Leftrightarrow \qquad \begin{pmatrix} \lambda_0 \, \mathrm{Id} & \lambda \\ \lambda^T & \lambda_0 \end{pmatrix} \succeq 0.$$

Using the above result and Theorem 2, we can reformulate constraints (10) as LMIs.

Corollary 1. *Robust constraints*

$$\mu_a^T(f)\xi \leq \nu_a(f,x) \text{ for all } \xi \in \Xi = \{\xi \mid \|\xi\|_2 \leq \kappa_2\} \text{ and } a \in A$$

are equivalent to constraints in x and f

$$\begin{pmatrix} \nu_a(f,x)\,\mathrm{Id} & \kappa_2\mu_a(f) \\ \kappa_2\mu_a^T(f) & \nu_a(f,x) \end{pmatrix} \succeq 0 \qquad a \in A, \tag{15}$$

where Id *is the $(|K|+1) \times (|K|+1)$ identity matrix.*

4. Decomposition

The reformulation from the previous section handles only pure ellipsoidal uncertainty sets. These sets are however not used in the robust network design literature because they tolerate very high individual demand perturbations, which does not happen in practice. This situation does not occur with the alternative sets described in Section 2. Therefore, our first motivation in this section is to provide a solution methodology for solving (RND) exactly that is also able to cope with an uncertainty set that is a polytope or the intersection of a box and an ellipsoid. To this end, we propose the cutting plane algorithm described in what follows. Then, we report numerical experiments that show for ellipsoidal uncertainty sets that our method outperforms the semidefinite programming reformulation by orders of magnitude. For the remaining uncertainty sets, we prove in this section that (RND) is also polynomially solvable.

We specify in the next subsection our cutting plane algorithm. The separation of robust constraints (5) and (3) is then discussed in Subsections 4.2 and 4.3, respectively.

4.1 Cutting plane algorithm

Problem (RND) is a linear program that contains an infinite number of constraints. A solution approach for the problem could seek to solve it by separating the required constraints on the fly. Let Ξ_0^* and Ξ_a^*, $a \in A$, denote finite subsets of Ξ. The algorithm we propose in this section relies on a master problem

$$
(MP) \qquad
\begin{array}{lll}
\min & \displaystyle\sum_{a\in A} c_a x_a + \theta & \\
\text{s.t.} & \displaystyle\theta \geq \sum_{k\in K} r^k(\xi) d^k(\xi) g^k & \xi \in \Xi_0^* \qquad (16) \\
& \displaystyle\sum_{k\in K} d^k(\xi) f_a^k \leq u_a + x_a & a \in A, \xi \in \Xi_a^* \qquad (17) \\
& (4), (6), (7), &
\end{array}
$$

derived from (RND) by relaxing all constraints (3) and (5) corresponding to vectors ξ not belonging to Ξ_0^* and Ξ_a^*, respectively. Problem (MP) is a finite linear program so that it can be solved by efficient solvers such as the simplex algorithm from CPLEX (2013). Then, if the optimal solution to (MP) violates constraints (3) and (5) corresponding to vectors ξ that do not belong to Ξ_0^* or Ξ_a^*, the corresponding vectors are appended to Ξ_0^* or Ξ_a^* and (MP) is solved again. This iterative procedure continues until the optimal solution to (MP) does not violate any constraint in Ξ. For completeness, this cutting-plane algorithm is formally described in Algorithm 1.

Algorithm 1: Cutting-plane algorithm

```
repeat
    solve $(MP)$;
    let $(\theta^*, x^*, f^*, g^*)$ be an optimal solution;
    foreach $a \in A$ do
        solve $\max_{\xi\in\Xi} \sum_{k\in K} d^k(\xi) f_a^{*k} - u_a + x_a^*$;        // Separating capacity constraints
        let $\xi^*$ be an optimal solution and $z_a^*$ be the optimal solution cost;
        if $z_a^* > 0$ then add $\xi^*$ to $\Xi_a^*$;
    solve $\max_{\xi\in\Xi} \sum_{k\in K} r^k(\xi) d^k(\xi) g^{*k} - \theta^*$;        // Separating cost constraints
    let $\xi^*$ be an optimal solution and $z_0^*$ be the optimal solution cost;
    if $z_0^* > 0$ then add $\xi^*$ to $\Xi_0^*$;
until $z_a^* \leq 0$ for all $a \in A$ and $z_0^* \leq 0$;
```

The crucial step of Algorithm 1 lies in the efficient separation of capacity and cost constraints. This is detailed in the next subsections.

4.2 Capacity constraints

Let $a \in A$ be fixed and recall the notations $\mu_a(f)$ and $\nu_a(f,x)$ introduced in subsection 3.2. In what follows, we take a closer look at the separation problem

$$\max_{\xi \in \Xi} \xi^T \mu_a(f) - \nu_a(f,x), \tag{18}$$

which amounts to maximize a linear function of ξ over a convex uncertainty set Ξ, which is polynomially solvable. Thus, the remaining of this subsection aims at reducing the time complexity for the special cases considered in this paper. Clearly, the complexity of problem (18) depends on the structure of Ξ. If Ξ is a polyhedron such as the hose model or the budgeted polytope, then problem (18) turns to a linear program. If Ξ is the intersection of an ellipsoid and a box or an ellipsoid, problem (18) turns to convex optimization problems with one second-order cone constraint with and without bounds on ξ, respectively. In fact, these two problems can be solved much faster than second-order cone programs by using the special structure of the uncertainty set. If Ξ is an ellipsoid of radius κ_2, then it is well-known that the optimal solution ξ^* to problem (18) is obtained when ξ^* is collinear to $\mu_a(f)$, that is, for $\xi^* = \kappa_2 \frac{\mu_a(f)}{\|\mu_a(f)\|_2}$. Finally, if Ξ is the intersection of a box and an ellipsoid, Kreinovich et al. (2008) showed that the problem can be solved essentially by ordering the coefficients $\mu_a(f)$ for $a \in A$. Our proof below is simpler and more direct than the general result from Kreinovich et al. (2008). To simplify notations, $\mu_a(f)$ is denoted by μ in the result below.

Lemma 2. *There exists a set $K^+ \subseteq K$ containing all the indices k having μ^k strictly larger than a given threshold μ^* such that the optimal solution to optimization problem*

$$\max_{\xi} \{\xi^T \mu : \|\xi\|_\infty \le 1, \|\xi\|_2 \le \kappa_2\} \tag{19}$$

is attained at $\xi^{*k} = 1$ for each $k \in K^+$ and $\xi^{*k} = \dfrac{\sqrt{\kappa_2^2 - |K^+|}\mu^k}{\sqrt{\sum_{k \in K \setminus K^+} (\mu^k)^2}}$ for each $k \in K \setminus K^+$.

Proof. Assume without loss of generality that $\mu^k \ge 0$ for each $k \in K$, so that $\|\xi\|_\infty \le 1$ can be replaced by

$$\xi^k \le 1 \text{ for each } k \in K. \tag{20}$$

For $\lambda \ge 0$, we define below the lagrangean relaxation of (19) by dualizing constraints (20):

$$\begin{aligned} L(\lambda) \quad = \quad \max \quad & \sum_{k \in K} \mu^k \xi^k - \sum_{k \in K} \lambda^k (\xi^k - 1) = \sum_{k \in K} (\mu^k - \lambda^k) \xi^k + \sum_{k \in K} \lambda^k \\ \text{s.t.} \quad & \sum_{k \in K} (\xi^k)^2 \le \kappa_2^2. \end{aligned} \tag{21}$$

For any $\lambda \ge 0$, the optimal solution of $L(\lambda)$ is attained at

$$\xi^{*k} = \frac{\kappa_2 (\mu^k - \lambda^k)}{\sqrt{\sum_{k \in K} (\mu^k - \lambda^k)^2}} \text{ for each } k \in K. \tag{22}$$

Let λ^* be the optimal solution of $\min_\lambda L(\lambda)$ and let ξ^* be the associated primal solution defined by equations (22). Because $\kappa_2 > 0$, Slater's condition holds, implying the complementarity slackness conditions $(1 - \xi^{*k})\lambda^{*k} = 0$ for each $k \in K$.

Let $K^+ = \{k \in K | \lambda^{*k} > 0\}$. For each $k \in K^+$, the complementarity slackness conditions impose that $\xi^{*k} = 1$, and Eq. (22) for k becomes

$$\mu^k - \lambda^{*k} = \frac{\sqrt{\sum_{h\in K}(\mu^h - \lambda^{*h})^2}}{\kappa_2} \equiv \mu^*. \tag{23}$$

Because ξ^* is feasible for problem (19), we have for each $k \in K\backslash K^+$ that

$$\xi^{*k} = \frac{\kappa_2 \mu^k}{\sqrt{\sum_{h\in K}(\mu^h - \lambda^{*h})^2}} \leq 1 = \frac{\kappa_2 \mu^*}{\sqrt{\sum_{h\in K}(\mu^h - \lambda^{*h})^2}}.$$

Hence, $\mu^k \leq \mu^* < \min\limits_{h\in K^+} \mu^h$ for each $k \in K\backslash K^+$, where the strict inequality follows from Eq. (23). Therefore, we have proven that there exists a set $K^+ \subseteq K$ containing all the indices k having $\mu^k > \mu^*$ such that $\xi^{*k} = 1$ for each $k \in K^+$, and the result follows from optimizing the remaining values of ξ^* over the ellipsoid of radius $\sqrt{\kappa_2^2 - |K^+|}$. □

Lemma 2 enables us to solve problem (19) in polynomial time in a two-steps algorithm. First, we sort the elements of μ. Then, we search for the optimal K^+ through a linear search on $|K^+|$ where both $\sqrt{\kappa_2^2 - |K^+|}$ and $\sqrt{\sum_{k\in K\backslash K^+}(\mu^k)^2}$ are updated in $O(1)$ time at each step.

4.3 Outsourcing costs

In what follows, we comment on how to solve the separation problem

$$\max_{\xi\in\Xi} \sum_{k\in K} r^k(\xi) d^k(\xi) g^k. \tag{24}$$

We can assume that $g \neq 0$ since otherwise the maximum can be trivially computed. The objective function of problem (24) is a bi-affine function. Denoting affine functions $1 + \beta\xi^0$ and $\sum_{k\in K} \overline{r}^k d^k(\xi) g^k$ by $a(\xi)$ and $b(\xi)$, respectively , problem (24) can be reformulated as

$$\max_{\xi\in\Xi} a(\xi) b(\xi). \tag{25}$$

Problem (25) is often denoted as a linear multiplicative program in the literature and is known (Matsui, 1996) to be $\mathcal{NP}$-hard when no assumption is made on the affine functions and Ξ is a general polyhedron. Recalling the discussion that follows Eq. (2), problem (25) can be reformulated as

$$\sup_{\xi\in\Xi, a(\xi)>0, b(\xi)>0} a(\xi) b(\xi). \tag{26}$$

Hence, the optimal solution ξ^* to problem (25) is the same as the optimal solution to

$$\sup_{\xi \in \Xi, a(\xi)>0, b(\xi)>0} \ln(a(\xi)) + \ln(b(\xi)), \tag{27}$$

which is a convex optimization problem that can be solved in polynomial time for any convex set Ξ that has a polynomial time separation oracle. Among the many optimization algorithms that exist to solve convex optimizations problems such as problem (27), we opted for a cutting-plane algorithm inspired by Kelley (1960), which relies on approximating the objective function via tangent cutting planes corresponding to elements in a finite subset $\Xi' \subset \Xi$:

$$\begin{array}{llll} & \sup & \gamma & \\ (SLA) & \text{s.t.} & \gamma \leq (\xi - \xi')^T \nabla(\ln(a) + \ln(b))(\xi') & \xi' \in \Xi' \\ & & \xi \in \Xi & \\ & & a(\xi) > 0,\ b(\xi) > 0. & \end{array}$$

For completeness, the algorithm is described in Algorithm 2. An interesting property of our algorithm is that the solution of (SLA) is implemented with the efficient simplex solver of CPLEX.

Algorithm 2: Kelley's cutting-plane algorithm

```
repeat
    solve $(SLA)$;
    let $(\xi^*, \gamma^*)$ be an optimal solution;
    define $z^* = \gamma^* - \ln(a(\xi^*)) - \ln(b(\xi^*))$;
    if $z^* > 0$ then add $\xi^*$ to $\Xi'$;
until $z^* \leq 0$;
```

4.4 Complexity of (RND)

We have seen in this section that constraints (3) and (5) can be separated in polynomial time whenever Ξ is a convex uncertainty set. Using the equivalence between separation and optimization (Grötschel et al., 1993), it follows that (RND) can be solved in polynomial time.

Theorem 3. *Consider problem (RND) and let Ξ be a convex uncertainty set that has a polynomial time separation oracle. Then, problem (RND) can be solved in polynomial time.*

5. Commodity dependent outsourcing costs

We suppose in this section that the rejection cost for each commodity is an arbitrary affine function of ξ denoted by $\rho^k(\xi)$. Hence, the worst-case cost of outsourcing demands becomes

$$\max_{\xi \in \Xi} \sum_{k \in K} \rho^k(\xi) d^k(\xi) g^k. \tag{28}$$

One readily sees that the previous approach does not lead to polynomial time algorithm in this case, because problem (28) is now a *sum of* linear multiplicative programs. Namely, replacing the objective function of problem (28) by its logarithm does not make it concave. In fact, we show next that the problem turns out to be hard when rejection costs can be different. More precisely, let $(RND(\rho^k))$ be the problem of minimizing $c(f,x,g) \equiv \sum_{a \in A} c_a x_a + \max_{\xi \in \Xi}\{\sum_{k \in K} \rho^k(\xi) d^k(\xi) g^k\}$ subject to (4),(5),(6), and (7). The decision version of $(RND(\rho^k))$, referred to as $(RND(\rho^k, \eta))$, is then defined by the following question.

Question 1. *Given a threshold robust cost η, is there a feasible solution (f,x,g) to $(RND(\rho^k))$ such that $c(f,x,g) \leq \eta$?*

We prove that $(RND(\rho^k, \eta))$ is co$\mathcal{NP}$-complete in the strong sense for general polytopes Ξ.

Theorem 4. *If Ξ is a general polytope, the $(RND(\rho^k, \eta))$ is strongly co$\mathcal{NP}$-complete.*

Proof. First, we show that $(RND(\rho^k, \eta))$ belongs to co$\mathcal{NP}$. To see this, let $c(f,x,g,\xi) \equiv \sum_{a \in A} c_a x_a + \sum_{k \in K} \rho^k(\xi) d^k(\xi) g^k$. A positive answer to question 1 indicates that there exists (f,x,g) that satisfies (4),(5),(6), (7), and $c(f,x,g,\xi) \leq \eta$ for all $\xi \in \Xi$. Hence, to answer "no" to this question, one must prove that the previous semi-infinite linear program is infeasible. Since only $|A| \times |K| + |A| + |K| + 1$ linear constraints are always sufficient to ensure that a linear program with $|A| \times |K| + |A| + |K|$ variables is infeasible, there always exists a certificate for the answer "no" that can be verified in polynomial time.

Now, we present a reduction from the σ-stable set problem defined as follows: given an undirected graph $G' = (V', E')$ and an integer σ, find a subset S of V' with cardinality σ such that, for every pair of vertices $i, j \notin S$, $(i,j) \in E'$. Let us refer to the aforementioned subset S as a σ-stable set for G'. In our reduction, it exists if and only if the answer to question 1 for the corresponding instance of $(RND(\rho^k, \eta))$ is "no", which leads to this theorem.

Let us assume without loss of generality that $\sigma \leq |V'|/2$. If this is not the case, one can obtain an equivalent instance of the σ-stable set problem that satisfies the previous condition by adding vertices to G' connected to all other vertices.

We build the following instance of $(RND(\rho^k, \eta))$: $V = \{1, 2\}$, $A = \{(1,2)\}$, $K = \{1, \dots, 2|V'|\}$, $d^k(\xi) = \rho^k(\xi) = \xi^k$, for each $k \in K$, $u_{(1,2)} = 0$, $c_{(1,2)} = 2$, and $\eta = |V'| - \epsilon$, where ϵ will be defined later. Moreover, we define the following polytope as an uncertainty set:

$$\Xi = \{\xi \in [0,1]^{|K|} \,|\, (\forall i \in V', \xi^{2i-1} + \xi^{2i} \leq 1), (\forall (i,j) \in E', \xi^{2i} + \xi^{2j} \leq 1), \sum_{i \in V'} \xi^{2i} \geq \sigma\}.$$

Notice that Ξ is never empty because $\xi^k = 0.5$ for all $k \in K$ always belong to it. The last constraint that defines Ξ is satisfied because of the assumption that $\sigma \leq |V'|/2$. Then, the corresponding instance of $(RND(\rho^k, \eta))$ asks for a solution (f, x, g) such that

$$(RND(\rho^k,\eta)) \text{ for } (\sigma, V', E') \qquad \begin{aligned} & c_{(1,2)}x_{(1,2)} + \sum_{k \in K} (\xi^k)^2 g^k \leq \eta && \xi \in \Xi && (29) \\ & \sum_{k \in K} \xi^k f^k_{(1,2)} \leq x_{(1,2)} && \xi \in \Xi && (30) \\ & g^k = 1 - f^k_{(1,2)} && k \in K && (31) \\ & f, x, g \geq 0. && && (32) \end{aligned}$$

Since we always have $(\xi^k)^2 \leq \xi^k$, it is always worth rejecting all the demand to minimize the left-hand side of (29). It remains to decide whether there exists $\xi \in \Xi$ such that $\sum_{k \in K} (\xi^k)^2 > \eta$. For that, we first prove that $\sum_{k \in K} (\xi^k)^2 \geq |V'|$ is possible if and only if there exists a σ-stable set for G'. This is true because, by the definition of Ξ, $\sum_{k \in K} (\xi^k)^2 \geq |V'|$ if and only ξ is an integer vector, in which case the set $\{i \in V' \,|\, \xi^{2i} = 1\}$ is a σ-stable set for G'.

It only remains to give a polynomial value for ϵ such that $\sum_{k \in K} (\xi^k)^2 > |V'| - \epsilon$ suffices to ensure the existence of a σ-stable set for G'. If the answer to the stable set instance is "no", then any integer ξ violates at least one of the constraints that define Ξ by at least one unit. As a result, any $\xi \in \Xi$ must be at least "one unit apart" from any integer solution. More formally, we must have

$$\sum_{k \in K} \min\{\xi^k, 1 - \xi^k\} \geq 1. \tag{33}$$

Since the function $F(\xi) = \sum_{k \in K} (\xi^k)^2$ is convex, this function is maximized subject to (33) and $\xi^{2i-1} + \xi^{2i} \leq 1$ for all $i \in V'$ when ξ^k is integer for all $k \in K$ except two, for which $\xi^k = 0.5$. Hence, we must have

$$F(\xi) \leq |V'| - 1 + 2(0.5)^2 = |V'| - 0.5$$

for any $\xi \in \Xi$. Thus, any $\epsilon \leq 0.5$ suffices. □

6. Computational experiments

We present in this section our computational study carried out on the real networks from SNDlib. The main objective of this section is to prove that cutting plane algorithms are more appropriated to handle constraints (3) than the classical SDP reformulation. As a byproduct of this approach, we compare the computational complexity and solution costs provided by the three uncertainty sets introduced in Section 2 solved by two variants of cutting plane algorithms. Finally, we also compare our new model with the classical robust network design model where no demand outsourcing is permitted,and to the deterministic models where the uncertainty sets are reduced to the nominal value.

6.1 Test problems

6.1.1 Instances

We test our models and algorithms on fifteen realistic network instances available from SNDlib (Orlowski et al., 2010). The main characteristics of these instances are reminded in Table 1. The networks being undirected, we chose to duplicate each arc to be sure that each commodity can be served by at least one directed path. For each network, we consider that the price elasticity of demand $-\alpha/\beta$ can take each value in $\{-0.2, -0.5, -0.8\}$, which are reasonable values according to the literature (Agiakloglou and Yannelis, 2006). Then, the value of β is set to 0.5 and α is computed accordingly.

	$\|V\|$	$\|A\|$	$\|K\|$
di-yuan	11	84	22
pdh	11	68	24
polska	12	36	66
nobel-us	14	42	91
atlanta	15	44	210
newyork	16	98	240
france	25	90	300
india35	35	160	595
cost266	37	114	1332
abilene1	12	30	66
abilene2	12	30	65
germany17	17	52	106
geant1	22	72	181
geant2	22	72	170
germany50	50	352	662

Table 1: Instances description.

6.1.2 Uncertainty

We describe in the following how uncertainty has been characterized in our numerical experiments. Recent published results (Babonneau et al., 2013; Poss and Raack, 2013) state that forecast errors on demand are of the order of 50% of the nominal value of the demand. Hence, for the first nine networks, we fix $\hat{d}^k = 0.5\overline{d}^k$ for each $k \in K$. For the last six networks, our values for $\hat{d}^k$ and $\overline{d}^k$ are those proposed by Koster et al.

(2013), which are based on historical data. Deciding for realistic values of $\overline{r}$ is harder because previous literature on the problem decide for somewhat arbitrarily high values for $\overline{r}$, and suppose that the parameters are fixed, see Andrade et al. (2006). We considered two sets of instances which generate $\overline{r}$ in two distinct ways. In the first set, the value of $\overline{r}^k$ for each $k \in K$ is equal to $\widetilde{r}$, which is comprised between 104% and 106% of the capacity cost per unit of nominal demand, computed as follows. Let z be the optimal solution cost for the deterministic version of the problem where the demand takes its nominal value $\overline{d}$. The nominal outsourcing cost is defined as $\widetilde{r} = \frac{\Delta z}{\sum_{k \in K} \overline{d}^k}$, where Δ can take any value in $\{1.04, 1.05, 1.06\}$. In the second set of instances, the value of $\overline{r}^k$ for each $k \in K$ is given by $\overline{r}^k = \frac{SP^k}{SP}\widetilde{r}$, where SP^k is the value of the shortest path from $s(k)$ to $t(k)$ using costs c, and $SP = \sum_{k \in K} \frac{SP^k}{|K|}$. Next, we recall how to choose the size of uncertainty sets to ensure a probabilistic protection of the capacity constraints when no precise knowledge about the actual probability distributions for demands is available.

We consider constraint-wise uncertainty in this paper so that we can recall the probabilistic results for a single robust constraint. Let $\eta(\xi)$ denote a vector of uncertain coefficients $\eta_i(\xi)$ for $i \in I$ which depend affinely on the uncertainty, that is, $\eta = \eta^0 + \sum_{k \in K} \eta^k \xi^k$ for given vectors η^0 and η^k in $\mathbb{R}^{|I|}$. Let us also denote by γ a constant scalar and by y a vector of optimization variables. Given uncertainty set Ξ, we consider robust constraint

$$\sum_{i \in I} \eta_i(\xi) y_i \leq \gamma \text{ for all } \xi \in \Xi. \tag{34}$$

We can associate an ambiguous chance constraint to robust constraint (34) by introducing arbitrary random variables independently and symmetrically distributed in $[-1, 1]$, denoted by $\tilde{\xi}^k$ for each $k \in K$, and defining the random variable associated to η as $\tilde{\eta} = \eta^0 + \sum_{k \in K} \eta^k \tilde{\xi}^k$. We recall below two well-known bounds for the ambiguous chance constraint associated with robust constraint (34). The proofs are omitted and can be found, for instance, in Bertsimas and Sim (2004) and Ben-Tal et al. (2009).

Theorem 5. *Let y^* be a vector that satisfies the robust constraint* (34). *It holds that*

$$P\left(\sum_{i \in I} \tilde{\eta}_i y_i^* > \gamma\right) \leq \epsilon,$$

where ϵ is defined as follows:

1. *If $\Xi = \{\xi \mid \|\xi\|_2 \leq \kappa_2\}$ or $\Xi = \{\xi \mid \|\xi\|_\infty \leq 1, \|\xi\|_2 \leq \kappa_2\}$, then $\epsilon = \exp(-\kappa_2^2/2)$.*
2. *If $\Xi = \{\xi \mid \|\xi\|_\infty \leq 1, \|\xi\|_1 \leq \kappa_1\}$, then $\epsilon = \frac{1}{2^{|K|}}\left((1-\mu)\binom{|K|}{\lfloor \nu \rfloor} + \sum_{l=\lfloor \nu \rfloor + 1}^{|K|} \binom{|K|}{l}\right)$, where $\nu = (\kappa_1 + |K|)/2$, $\mu = \nu - \lfloor \nu \rfloor$.*

Notice that Poss (2013b, 2014) has recently pointed out the conservatism of the second probabilistic bound from Theorem 5. Poss (2013b) proposes an alternative approach based on using multifunctions instead of the classical budgeted uncertainty set, which is tractable for problems involving only binary variables. For optimization problems involving real variables as in this paper, the approach of Poss (2013b) is $\mathcal{NP}$-hard so that we stick to the classical budgeted uncertainty set in the following.

6.2 Experiments

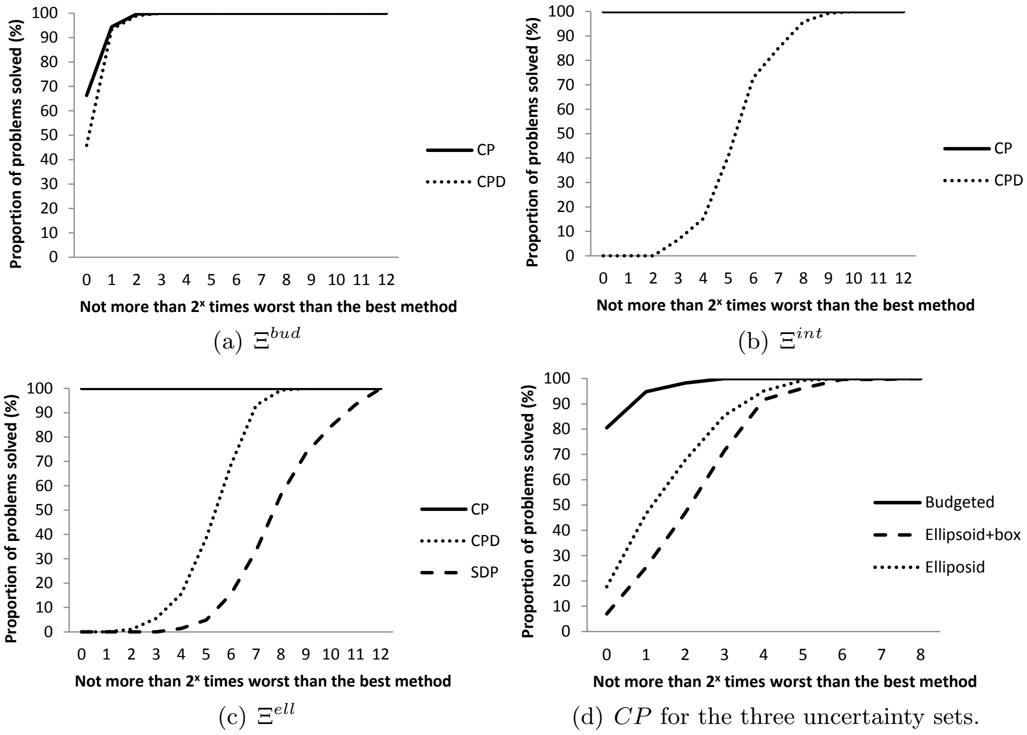
Our model has been solved for the three uncertainty sets recalled in Theorem 5, denoted Ξ^{ell} (Ellipsoid), Ξ^{int} (Ellipsoid+box), and Ξ^{bud} (Budgeted) in what follows. We have considered two solution algorithms calling the linear and conic quadratic solvers of CPLEX 12.4 (CPLEX, 2013). The first algorithm, denoted by CP, is a pure cutting plane algorithm that follows Algorithm 1. The second algorithm, denoted by $CP{+}D$, only separates cost constraints; capacity constraints are dualized according to Theorem 2 and included in the master problem. For the pure ellipsoidal uncertainty set and model (RND), we have also solved the SDP reformulation obtained by combining Theorem 1 and Corollary 1 by csdp (Borchers, 1999), denoted by SDP. The experiments are carried on a computer equipped with a processor Intel Core i7 at 2.90 GHz and 8 GB of RAM memory. Table 2 provides the number of additional variables (coming from the dualization of robust constraints) and the number of constraints present in the conic reformulations (before cutting planes are added) for each type of uncertainty set; notice that in the ellipsoidal case, we consider separately algorithms $CP{+}D$ and SDP because the former contains linear and second-order cone constraints (LC and SOCC, respectively) while the latter contains only LMIs.

	Budgeted	Ellipsoid + box	Ellipsoid	Ellipsoid
	$CP{+}D$	$CP{+}D$	$CP{+}D$	SDP
Flow conservation	$\lvert K\rvert(\lvert V\rvert-1)$	$\lvert K\rvert(\lvert V\rvert-1)$	$\lvert K\rvert(\lvert V\rvert-1)$	$\lvert K\rvert(\lvert V\rvert-1)$
(4),(6)	LC	LC	LC	LMIs of size 1
Capacity	$\lvert A\rvert(\lvert K\rvert+2)$ LC	$\lvert A\rvert$ SOCC	$\lvert A\rvert$ SOCC	$\lvert A\rvert$ LMIs of size $\lvert K\rvert+1$
(5)	$\lvert A\rvert(\lvert K\rvert+2)$ variables	$2\lvert A\rvert(\lvert K\rvert+1)$ variables		
Outsourcing cost				1 LMI of size $\lvert K\rvert+2$
(9)	–	–	–	1 LMI of size 1
				1 variable

Table 2: Numbers of constraints and additional variables in the conic reformulations.

Our results are grouped into two sets of experiments. In Section 6.2.1, we consider only four networks (*di-yuan, nobel-us, abilene1, and germany 17*) and compare all algorithms for all parameters combinations in terms of computational times. This yields a set of 288 different instances, each of them solved for the three different uncertainty sets. Then, we study sensibility of the computational times and solutions costs to each individual parameter.

In Section 6.2.2, we fix all parameters to a single value and solve the fifteen networks for the different uncertainty sets with the most efficient algorithm (CP). We provide detailed results, including the numbers of cutting planes generated and iterations of the algorithm. In this second set of experiments, we have also



(a) Ξ^{bud}

(b) Ξ^{int}

(c) Ξ^{ell}

(d) CP for the three uncertainty sets.

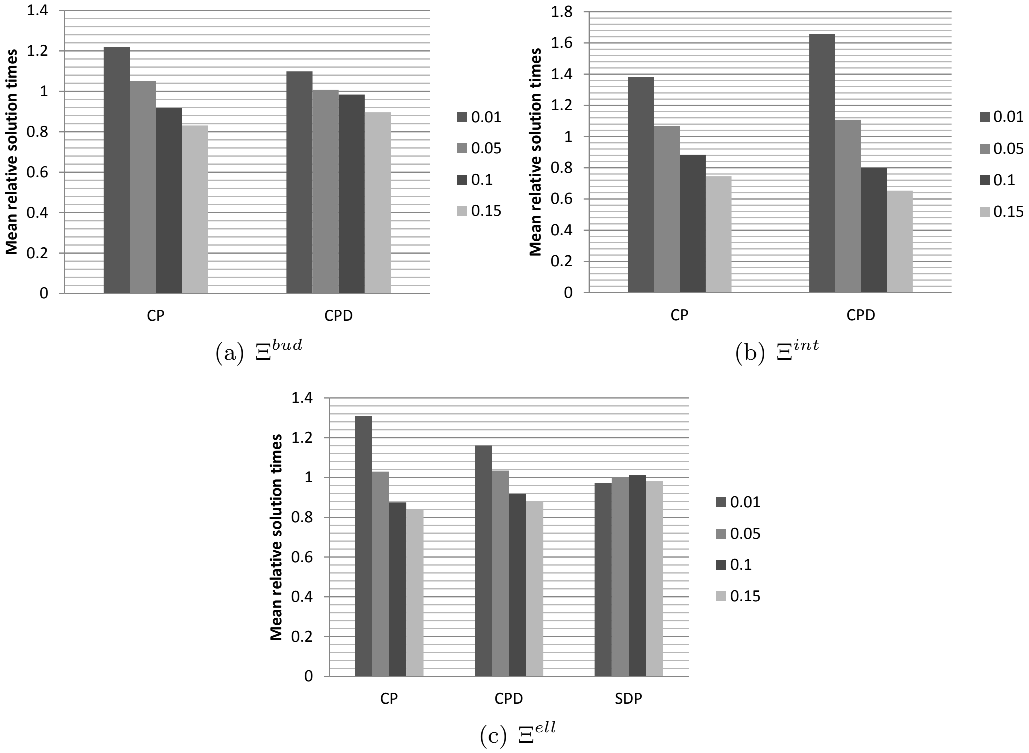
Figure 1: Performances profiles aggregating all solution times.

considered a simplification of (RND) where no outsourcing is allowed. We also compare these models to the deterministic models where the uncertainty sets are reduced to the nominal values.

6.2.1 Comparison of all algorithms on four networks

We present in this section aggregated results realized on four networks: *di-yuan, nobel-us, abilene1, and germany 17.* We chose networks of moderated dimensions to avoid memory overloads with SDP and very high solution times with $CP{+}D$ and SDP.

Figure 1 exhibits performance profiles (Dolan and Moré, 2002) that compare the different algorithms available for each uncertainty set. We see from Figure 1(b) and Figure 1(c) that for Ξ^{ell} and Ξ^{int}, CP outperforms by far $CP{+}D$. For instance, half of the instances are solved around 32 times faster by CP than $CP{+}D$. We also see from Figure 1(c) that $CP{+}D$ clearly outperforms SDP. The difference between CP and $CP{+}D$ is even more important, the former being at least 128 times faster than the latter for half of the instances, a tenth of them being solved at least 1000 times faster. Figure 1(a) shows that in the case of polyhedron Ξ^{bud}, the performance of CP and $CP{+}D$ are much closer, CP having a slight advantage over $CP{+}D$. Since CP is the fastest algorithm for the three uncertainty sets, the objective of Figure 1(d) is compare CP among different uncertainty sets. We see that problems under Ξ^{bud} are easier to solve than the others, while problems under Ξ^{ell} are slightly easier than those under Ξ^{int}.

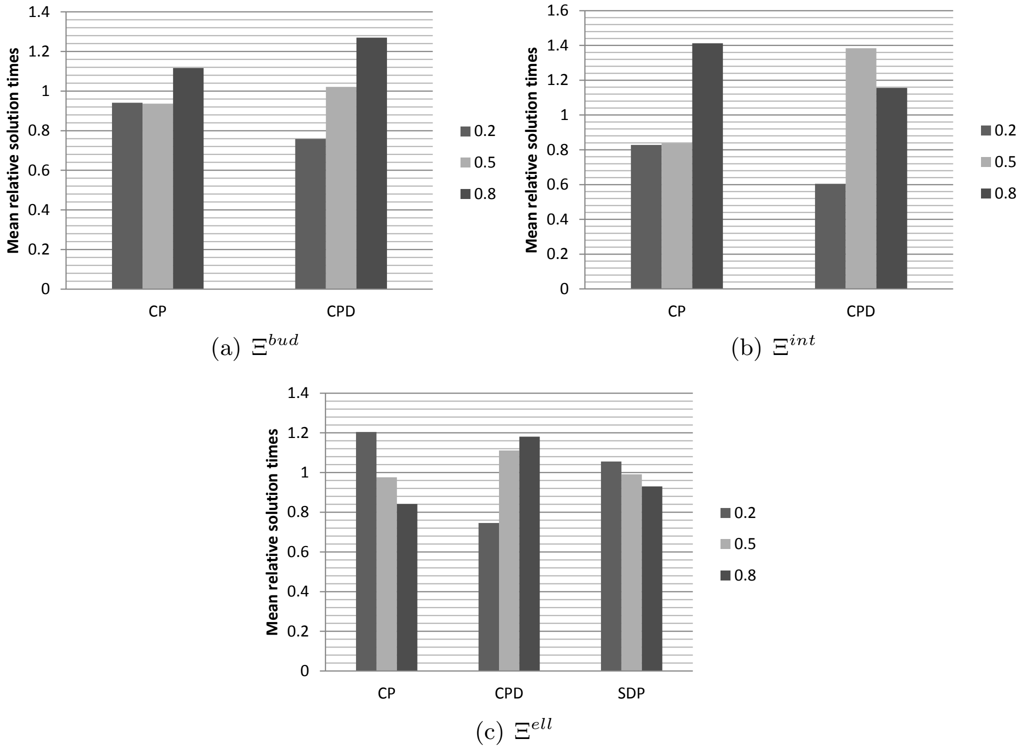


(a) Ξ^{bud}

(b) Ξ^{int}

(c) Ξ^{ell}

Figure 2: Geometric means of the relative solution times for the different values of ϵ.

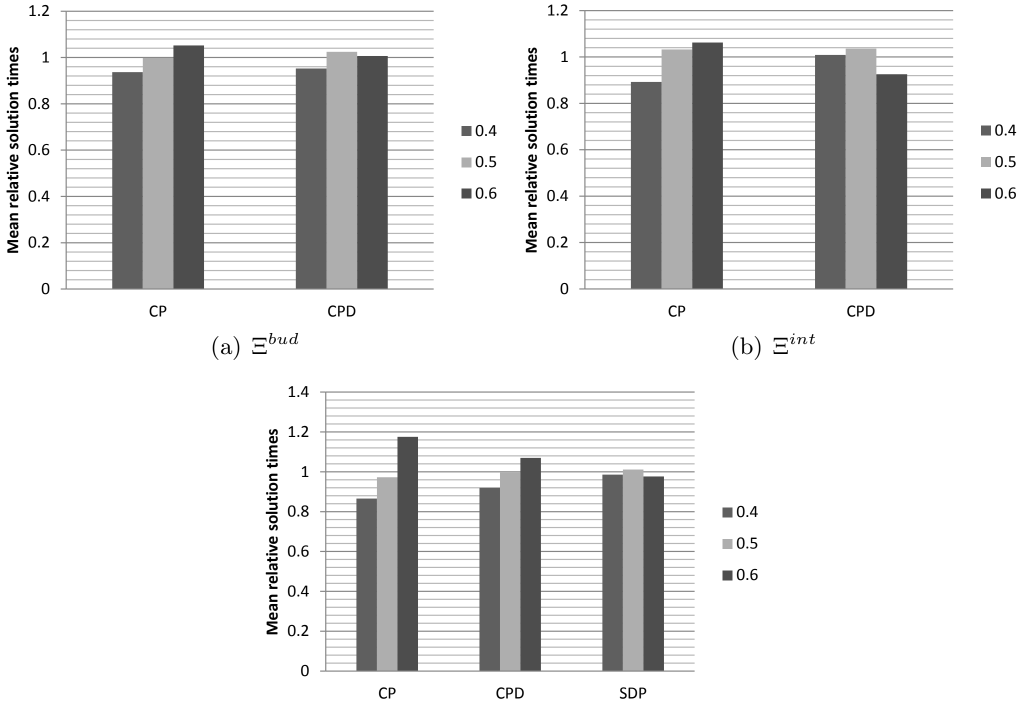


(a) Ξ^{bud}

(b) Ξ^{int}

(c) Ξ^{ell}

Figure 3: Geometric means of the relative solution times for the different values of α/β.

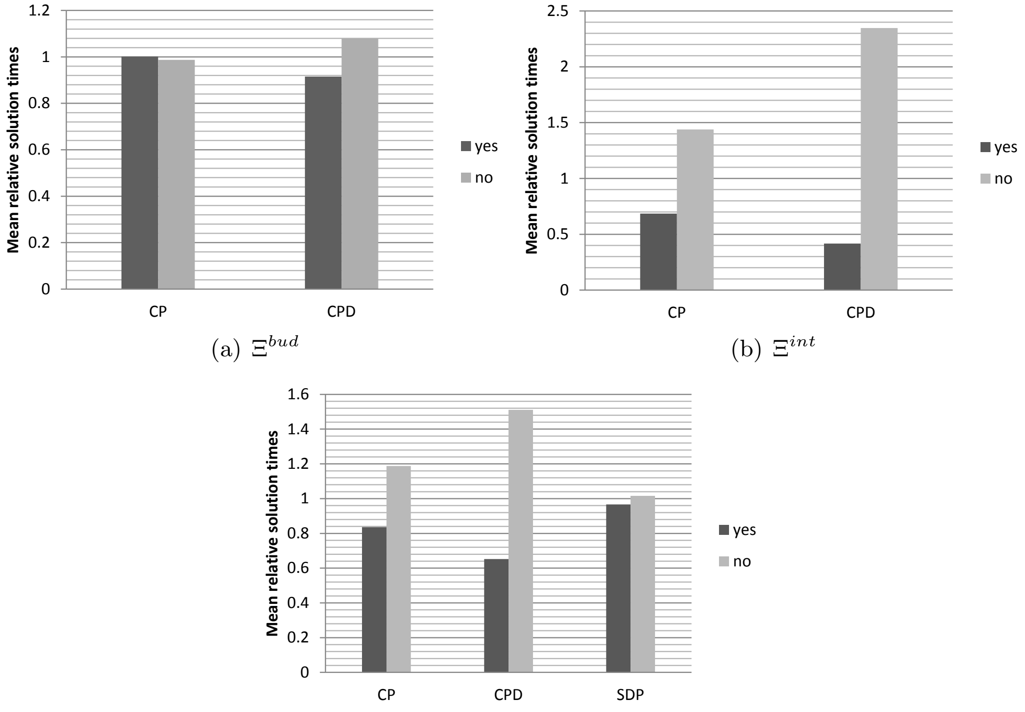


(a) Ξ^{bud}

(b) Ξ^{int}

(c) Ξ^{ell}

Figure 4: Geometric means of the relative solution times for the different values of Δ.



(a) Ξ^{bud}

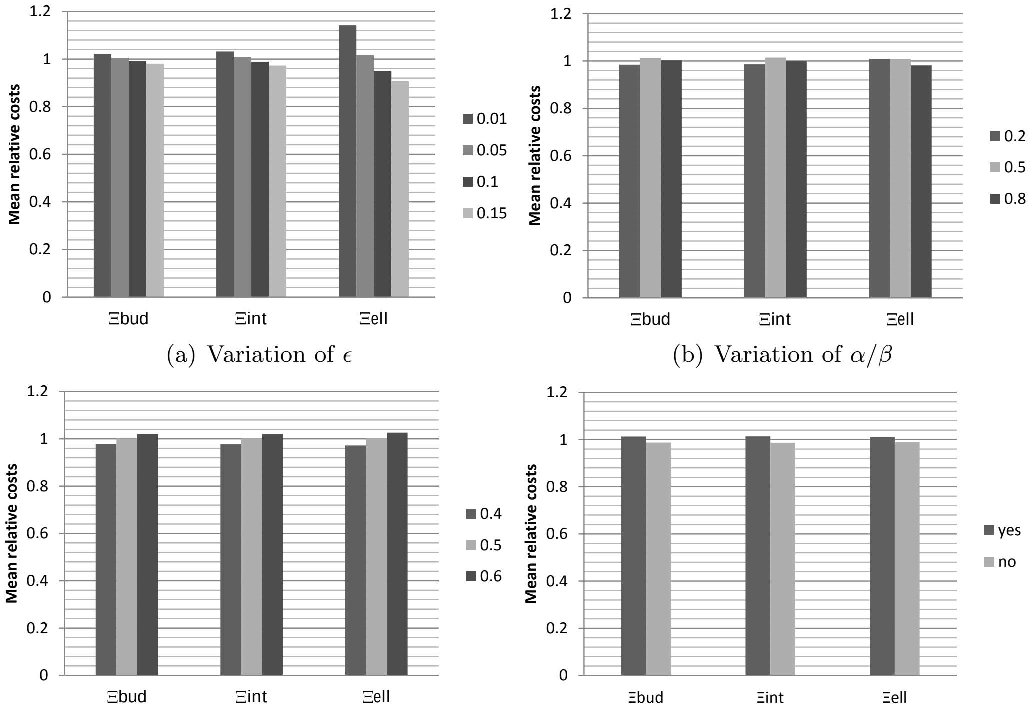
(b) Ξ^{int}

(c) Ξ^{ell}

Figure 5: Geometric means of the relative solution times depending on whether g^k is build using shortest paths.

We study next how varying individually each parameter affects the solution times of each algorithm for each uncertainty set. Figures 2–5 report the mean relative solution times, computed as follows. For each network, algorithm and uncertainty set, we compute the geometric mean of all solution times. Then, we divide each solution time by the computed mean, to obtain normalized solution times. Finally, Figures 2–5 report the geometric means of the normalized solution times, associating to different columns the instances that correspond to different values of the studied parameter. Notice that using these relative solution times instead of absolute solution times allows us aggregate solution times for instances that have different dimensions and complexity. Figure 2 analyzes the solution times when varying the probabilistic guarantee ϵ in $\{0.01, 0.05, 0.10, 0.15\}$. For CP and $CP{+}D$, smaller values of ϵ yields harder optimization problems, and the means of the normalized solution times vary between 0.8 and 1.65 times the mean value. The effect is particularly important for $CP{+}D$ applied to Ξ^{int} where instances corresponding to $\epsilon = 0.15$ are solved, on average, 2.5 times faster than instances corresponding to $\epsilon = 0.01$. Figure 2(c) shows that varying ϵ has only a marginal effect on SDP, and in fact, Figures 3(c), 4(c), and 5(c) yield the same conclusion when varying the other parameters. Figure 3 analyzes the solution times when varying the price elasticity of demand $-\alpha/\beta$ in $\{-0.2, -0.5, -0.8\}$, setting β to 0.5. The mean of the normalized solution times ranges between 0.6 and 1.4 times the mean value. No clear tendency can be drawn from the picture, since the different types of instances react differently for distinct algorithms and uncertainty sets. Figure 4 analyzes the solution times when varying the nominal outsourcing cost through parameter Δ taking value in $\{1.04, 1.05, 1.06\}$. This parameter seems to have only a marginal effect on the solution times. Nevertheless, for CP it seems that instances corresponding to smaller values of Δ are slightly easier to solve than those corresponding to higher values of the parameter. Finally, Figure 5 analyzes how the solution times are impacted when considering the shortest paths values in the outsourcing costs (denoted by *yes*). Apart from Ξ^{bud} and SDP applied to Ξ^{ell}, instances built according to shortest paths are clearly easier to solve than those which ignore the shortest paths. The effect is maximized for $CP{+}D$ applied to Ξ^{int}, where the former are solved almost 6 times faster than the latter, in average.

Figures 6 and 7 analyze the optimal solution costs when varying each parameter individually. Figure 6 reports the geometric means of relative solution costs, defined similarly to the relative computational times. Figure 6(a) shows that, as expected, the solution costs increase with the probabilistic level of protection ϵ, and that the effect is more marked for the ellipsoidal uncertainty set. Similarly, Figure 6(c) logically shows that increasing the nominal outsourcing cost leads to a cost increase. According to Figure 6(d), it seems that outsourcing costs depending on shortest paths lead to slightly more expensive solutions, and Figure 3(b) shows that the more expensive solutions are related to average elasticity. Overall, the solution costs seem to be marginally impacted by these changes in parameters values. This is contrast with Figure 7, which



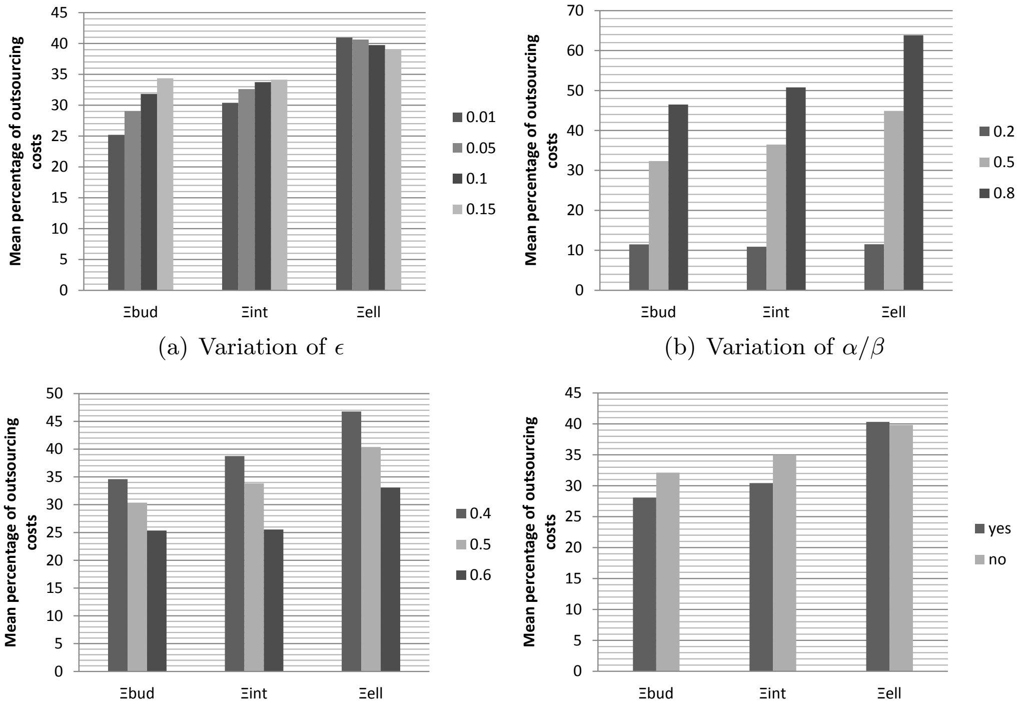
(a) Variation of ϵ

(b) Variation of α/β

(c) Variation of Δ

(d) Using Shortest Paths ?

Figure 6: Geometric means of the relative solution costs.



(a) Variation of ϵ

(b) Variation of α/β

(c) Variation of Δ

(d) Using Shortest Paths ?

Figure 7: Arithmetic means of proportional outsourcing costs (expressed in %).

presents the arithmetic means of the outsourcing costs expressed as percentages of the total costs. In the latter figure, the results can vary heavily with the values of the parameters. Figure 6(a) shows that the percentages may increase or decrease with ϵ, depending on the model of uncertainty. Figure 6(b) shows that the percentages increase significantly when the elasticity rises. More expectedly, Figure 6(c) shows that increasing nominal outsourcing costs leads to a lesser use of outsourcing. Finally, Figure 6(d) shows that the impact of using shortest paths in the outsourcing costs gives different results for the different uncertainty sets.

6.2.2 Cutting plane algorithms for all networks

We present next detailed results for the fifteen networks, setting the parameters as follows: $\epsilon = 0.10$, $-\alpha/\beta = -0.5$, $\Delta = 0.5$, and the shortest paths are not used in the outsourcing costs. We also focus on the cutting-plane algorithm, CP, which appears to be the most efficient algorithms. This reduced set of instances allows us to present detailed results, including the numbers of cutting planes and iterations of the algorithm. We also compare (RND) to the model without outsourcing, (RND^0), obtained from (RND) by adding constraint

$$g^k = 0, \qquad k \in K.$$

Additionally, we compare both models to their deterministic versions, (ND) and (ND^0), respectively. The deterministic models rely on *optimistic* or *pessimistic* data, where *optimistic* data consider that parameters take their nominal values, $d^k = \overline{d}^k$ and $r^k = \overline{r}^k$, for each $k \in K$, while *pessimistic* data consider that parameters take their worst values and ignore the price elasticity of demand, $d^k = \overline{d}^k(1+\alpha) + \hat{d}^k$ and $r^k = (1+\beta)\overline{r}^k$, for each $k \in K$. Although cheaper than model *pessimistic*, model *optimistic* yields infeasible solutions. Namely, if we were to use the fractional routings and capacities corresponding to its optimal solution, we would overload our capacities, which would result in very high delays and a poor quality of service (e.g. Ouorou et al. (2000)). Hence, in the absence of robust or stochastic models, decision makers are forced to use *pessimistic* data, which results in waste of capital. Therefore, the purpose of the robust models, such as those considered in this paper, is to reduce the cost of the pessimistic models, while ensuring a high protection against unexpected outcomes.

Table 3 presents the optimal solution costs for all models. For the deterministic models (ND) and (ND^0), we provide these costs explicitly. For the robust models, we provide the cost reduction of each model when compared to the cost of the pessimistic version of (ND), denoted $(ND)_{pessimistic}$ and $(ND^0)_{pessimistic}$, respectively. Namely, we report

$$100\frac{cost((ND)_{pessimistic}) - cost(M)}{cost((ND)_{pessimistic})}, \tag{35}$$

Network	(ND)		(ND^0)		(RND)			(RND^0)		
	optimistic	pessimistic	optimistic	pessimistic	Ξ^{bud}	Ξ^{int}	Ξ^{ell}	Ξ^{bud}	Ξ^{int}	Ξ^{ell}
di-yuan	3.29E+06	6.18E+06	3.53E+06	6.18E+06	4	6	0	0	0	-13
pdh	1.67E+07	2.93E+07	1.67E+07	2.93E+07	0	0	-15	0	0	-24
polska	2.94E+04	5.30E+04	3.03E+04	5.30E+04	8	10	7	0	6	1
nobel-us	3.93E+06	7.52E+06	4.30E+06	7.52E+06	16	19	16	0	6	2
atlanta	2.26E+07	1.08E+08	2.34E+07	1.09E+08	12	20	10	2	11	-1
newyork	1.15E+05	2.13E+05	1.25E+05	2.19E+05	12	15	10	-3	2	-5
france	1.81E+04	3.30E+04	1.89E+04	3.30E+04	12	18	15	3	14	7
india35	3.12E+03	5.59E+03	3.20E+03	5.59E+03	14	19	17	1	14	7
cost266	6.66E+06	1.24E+07	7.16E+06	1.25E+07	17	24	21	1	15	8
abilene1	1.04E+01	3.17E+01	1.09E+01	3.17E+01	3	5	-35	0	4	-38
abilene2	1.02E+01	2.13E+01	1.09E+01	2.13E+01	8	13	10	1	7	3
germany17	1.36E+01	3.32E+01	1.44E+01	3.33E+01	9	15	7	1	8	-3
geant1	1.30E+01	2.65E+01	1.34E+01	2.65E+01	11	15	8	1	8	2
geant2	1.21E+01	2.80E+01	1.27E+01	2.81E+01	10	14	4	0	6	-4
germany50	1.80E+01	4.29E+01	1.85E+01	4.29E+01	8	15	10	1	10	0
Arithmetic means					9	14	6	0	7	-4

Table 3: Optimal solution costs for deterministic models and percental cost reductions for robust models (expressed in %).

for each model M with outsourcing, and similarly for the models without outsourcing. We make next some comments about the table. First, we see that the optimistic models are much less expensive than the other models. Unfortunately, they cannot be used in practice. Fortunately, the cost reductions offered by the robust models can be significant, rising up to 24% and 15% for (RND) and (RND^0), respectively. The table also suggests that Ξ^{int} offers the highest cost reduction for both (RND) and (RND^0). In contrast with these good results obtained by Ξ^{int}, Ξ^{ell} can yield solutions even more expensive than the pessimistic model, represented by the negative numbers in the table. This situation happens because the diameter of the set prescribed by Theorem 5, κ_2, is usually greater than one, so that individual deviations can be greater than

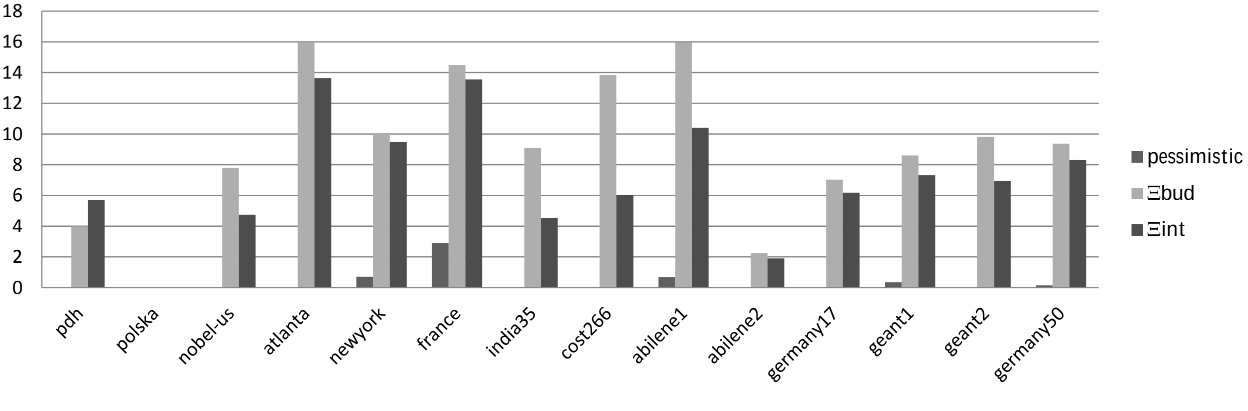


Figure 8: Cost reduction when allowing outsourcing (%).

the pessimistic values described above. While Ξ^{bud} never yields negative cost reductions, its performance is not much better than Ξ^{ell}, reminding the discussion of Poss (2013b) on the conservatism of the probabilistic bounds for Ξ^{bud}. Figure 8 provides a different perspective on the numbers from Table 3, by comparing the cost reductions between models allowing outsourcing or not, using formulae similar to (35). We see from the figure that there is almost no benefit to allow outsourcing for the pessimistic model, while outsourcing yields cost reductions of up to 16% and 14%, for Ξ^{bud} and Ξ^{int}, respectively.

Tables 4 and 5 present detailed computational results for models (RND) and (RND^0). Namely, we report solution times in seconds, numbers of iterations, and numbers of cuts generated throughout CP. We also report geometric and arithmetic means of these results. The arithmetic mean give more weight to hard instances, the geometric mean allows to aggregate results obtained for instances of different computational complexities in a balanced manner. Table 4 shows that Ξ^{bud} almost always leads to simpler optimization problems than Ξ^{ell}, which itself leads to easier optimization problems than Ξ^{int}. In Table 5, we see that Ξ^{int} is still the more complex uncertainty set, while Ξ^{bud} and Ξ^{ell} yields optimization problems of similar difficulties, Ξ^{ell} being faster than Ξ^{bud} for the largest instances. Analyzing the number of cutting planes, we see that Ξ^{int} requires, on average, almost twice as many cost cuts as the other models. We also realize that Ξ^{int} and Ξ^{ell} generate more capacity cuts than Ξ^{bud} in almost all instances. Let us compare finally the solutions of (RND) and (RND^0). For set Ξ^{bud}, (RND) is clearly easier than (RND^0), being one order of magnitude faster to solve, in average. This is a very good result since the introduction of outsourcing does not make the problem harder, even the contrary. For Ξ^{int} and Ξ^{ell} the results are more contrasted. Hard instances are easier to solve for (RND) than for (RND^0) (this is especially true for Ξ^{int}), while it is the opposite for easy instances.

7. Concluding remarks

We introduce in this paper a new model of robust telecommunications network design with outsourcing. Differently from previous literature on the topic that ignores the uncertainty in outsourcing costs, we argue that the price elasticity of demand leads to a negative correlation between uncertain demands and outsourcing costs. We reformulate the resulting optimization problem as a linear program that involves the separation of an additional set of inequalities. The separation problem is convex and can be solved in polynomial time, and therefore, the full model can be solved in polynomial time. The theoretical complexity is confirmed by our numerical experiments, which show that considering outsourcing does not make the problem harder to solve. For the budgeted uncertainty set, our experiments even show that outsourcing makes the optimization problem much easier to solve. We point out that the nice complexity properties hold only because the random

Network	Time			Iterations			Cost cuts			Capacity cuts		
	Ξ^{bud}	Ξ^{int}	Ξ^{ell}	Ξ^{bud}	Ξ^{int}	Ξ^{ell}	Ξ^{bud}	Ξ^{int}	Ξ^{ell}	Ξ^{bud}	Ξ^{int}	Ξ^{ell}
di-yuan	0.03	0.36	0.31	13	57	29	11	56	28	96	114	241
pdh	0.03	0.08	1.00	10	14	127	7	12	125	33	36	88
polska	0.23	1.53	0.84	49	125	49	48	124	48	195	772	1047
nobel-us	0.14	0.87	0.92	24	82	48	23	81	47	123	415	716
atlanta	0.75	3.88	2.32	49	133	87	48	129	81	446	1561	1187
newyork	0.41	2.12	2.22	36	113	48	35	112	47	388	613	1437
france	4.07	40.9	17.1	102	123	84	62	107	58	814	4171	2691
india35	127	691	155	79	132	51	78	105	49	2596	9693	4900
cost266	44.2	211	68.1	146	92	61	145	74	48	1537	5604	3497
abilene1	0.13	0.56	0.31	37	54	23	36	53	22	176	324	261
abilene2	0.13	0.69	0.59	33	65	41	30	64	40	174	277	420
germany17	0.34	1.81	0.56	55	115	28	46	114	21	260	294	326
geant1	1.30	3.40	5.71	73	107	85	72	106	84	465	408	752
geant2	0.81	1.95	2.12	66	84	42	65	83	41	379	262	476
germany50	53.1	98.1	60.5	125	141	29	124	140	23	2026	1702	1856
Geom. mean	0.83	3.93	3.04	47	85	49	43	80	45	346	644	791
Arithm. mean	15	70	21	60	96	55	55	91	51	647	1750	1326

Table 4: Detailed computational results for (RND).

Network	Time			Iterations			Capacity cuts		
	Ξ^{bud}	Ξ^{int}	Ξ^{ell}	Ξ^{bud}	Ξ^{int}	Ξ^{ell}	Ξ^{bud}	Ξ^{int}	Ξ^{ell}
di-yuan	0.06	0.03	0.13	12	57	16	123	165	336
pdh	0.03	0.03	0.06	4	14	4	33	37	220
polska	0.09	0.33	0.30	16	125	55	193	716	752
nobel-us	0.09	0.66	0.34	14	82	45	318	1127	832
atlanta	0.59	1.76	0.78	40	133	102	460	1278	812
newyork	0.80	2.71	2.78	25	113	69	768	1768	2002
france	10.7	100	84.6	95	123	288	1444	6597	6198
india35	692	2610	389	100	132	296	6000	15773	8360
cost266	402	397	96.9	109	92	127	5175	8687	3834
abilene1	0.08	0.06	0.06	23	54	16	219	212	175
abilene2	0.08	0.09	0.09	22	65	22	249	285	271
germany17	0.55	1.36	0.84	30	115	56	572	738	718
geant1	12.9	6.82	7.39	75	107	39	1796	1205	1177
geant2	13.4	5.85	3.76	76	84	36	1744	1118	997
germany50	1190	254	139	105	141	63	5203	3287	3019
Geom. mean	1.91	2.89	2.16	34	85	49	661	1047	1040
Arithm. mean	154	226	48	50	96	82	1620	2866	1980

Table 5: Detailed computational results for (RND^0).

part of outsourcing cost is identical for all commodities. If this were not the case, the resulting optimization problem would be $co\mathcal{NP}$-hard for arbitrary polyhedral uncertainty sets.

Another important contribution of this work points out the weakness of the classical semi-definite programming reformulation of robust inequalities with quadratic dependency on the uncertainty and ellipsoidal uncertainty set. While convex, this reformulation is up to three orders of magnitude slower than the simpler cutting plane algorithms implemented in this paper. Moreover, our approaches can handle uncertainty sets richer and more meaningful than the pure ellipsoidal model. Finally, cutting plane algorithms can, in theory at least, easily be embedded into branch-and-cut and branch-and-cut-and-price algorithms. Hence, an interesting topic of future research would see how to efficiently adapt our algorithms to robust network design problems with outsourcing and modular capacities and/or single path routing.

8. Aknowledgements

We are grateful to the referees and editors for suggestions that helped in improving the content of the paper. Thanks are also due to Manuel Kutschka for providing us with the instances from Koster et al. (2013). This research benefited from the support of the "FMJH Program Gaspard Monge in optimization and operation research", and from the support to this program from EDF. Moreover, a part of this paper was written during the stay of Artur Pessoa at UTC funded in the framework of Labex MS2T managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02). Artur Pessoa also received grants from FAPERJ and CNPq.

References

Agiakloglou, C., D. Yannelis. 2006. Estimation of price elasticities for international telecommunications demand. *International Advances in Economic Research* **12**(1) 131–137.

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

Andrade, R., A. Lisser, N. Maculan, G. Plateau. 2006. Enhancing a branch-and-bound algorithm for two-stage stochastic integer network design-based models. *Management Science* **52**(9) 1450–1455.

Babonneau, F., O. Klopfenstein, A. Ouorou, J.-P. Vial. 2013. Robust capacity expansion solutions for telecommunication networks with uncertain demands. *Networks* In press.

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

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

Ben-Tal, A., A. Nemirovski, C. Roos. 2002. Robust solutions of uncertain quadratic and conic-quadratic problems. *SIAM Journal on Optimization* **13**(2) 535–560.

Bertsimas, D., D. B. Brown, C. Caramanis. 2011. Theory and applications of robust optimization. *SIAM Review* **53**(3) 464–501.

Bertsimas, D., I. Dunning, M. Lubin. 2014. Reformulations versus cutting planes for robust optimization: A computational and machine learning perspective. Available at Optimization Online.

Bertsimas, D., M. Sim. 2004. The Price of Robustness. *Operations Research* **52**(1) 35–53.

Borchers, Brian. 1999. Csdp 2.3 user's guide. *Optimization Methods and Software* **11**(1-4) 597–611.

CPLEX. 2013. *IBM ILOG CPLEX 12.4 Reference Manual.*. ILOG CPLEX Division, Gentilly, France.

Dolan, E. D., J. J. Moré. 2002. Benchmarking optimization software with performance profiles. *Mathematical Programming* **91**(2) 201–213.

Fischetti, M., M. Monaci. 2012. Cutting plane versus compact formulations for uncertain (integer) linear programs. *Mathematical Programming Computation* **4**(3) 239–273.

Garbacz, C., H. G. Thompson Jr. 2007. Demand for telecommunication services in developing countries. *Telecommunications Policy* **31** 276–289.

Garstka, Stanley J. 1973. Stochastic programs with recourse: Random recourse costs only. *Management Science* **19**(7) 747–762.

Grötschel, M., L. Lovász, A. Schrijver. 1993. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag.

Kelley, Jr., J. E. 1960. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics* **8**(4) 703–712.

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

Kreinovich, V., A. Neumaier, G. Xiang. 2008. Towards a combination of interval and ellipsoid uncertainty. *Computational Technologies* **13**(6) 5–16.

Lee, C., K. Lee, K. Park, S. Park. 2012. Technical note—branch-and-price-and-cut approach to the robust network design problem without flow bifurcations. *Oper. Res.* **60**(3) 604–610.

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

Matsui, T. 1996. Np-hardness of linear multiplicative programming and related problems. *Journal of Global Optimization* **9**(2) 113–119.

Mattia, S. 2013. The robust network loading problem with dynamic routing. *Comp. Opt. and Appl.* **54**(3) 619–643.

Monaci, M., U. Pferschy, P. Serafini. 2013. Exact solution of the robust knapsack problem. *Computers & OR* **40**(11) 2625–2631.

Orlowski, S., M. Pióro, A. Tomaszewski, R. Wessäly. 2010. SNDlib 1.0–Survivable Network Design Library. *Networks* **55**(3) 276–286. doi:10.1002/net.20371. `http://sndlib.zib.de`.

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

Ouorou, A., P. Mahey, J.-P. Vial. 2000. A survey of algorithms for convex multicommodity flow problems. *Management Science* **46**(1) 126–147.

Ouorou, A., J.-P. Vial. 2007. A model for robust capacity planning for telecommunications networks under demand uncertainty. *6th International Workshop on Design and Reliable Communication Networks, 2007. DRCN 2007*. 1–4.

Pólik, I., T. Terlaky. 2007. A survey of the s-lemma. *SIAM Review* **49**(3) 371–418.

Poss, M. 2013a. A comparison of routing sets for robust network design. *Optimization Letters* In press.

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

Poss, M. 2014. Robust combinatorial optimization with variable cost uncertainty. *European Journal of Operational Research* **237**(3) 836 – 845.

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

Sen, S., R. D. Doverspike, S. Cosares. 1994. Network planning with random demand. *Telecommunication Systems* **3**(1) 11–30.