Noname manuscript No.
(will be inserted by the editor)

Decomposition for adjustable robust linear optimization subject to uncertainty polytope

Josette Ayoub · Michael Poss

Received: date / Accepted: date

Abstract We present in this paper a general decomposition framework to solve exactly adjustable robust linear optimization problems subject to polytope uncertainty. Our approach is based on replacing the polytope by the set of its extreme points and generating the extreme points on the fly within row generation or column-and-row generation algorithms. The novelty of our approach lies in formulating the separation problem as a feasibility problem instead of a max-min problem as done in recent works. Applying the Farkas lemma, we can reformulate the separation problem as a bilinear program, which is then linearized to obtained a mixed-integer linear programming formulation. We compare the two algorithms on a robust telecommunications network design under demand uncertainty and budgeted uncertainty polytope. Our results show that the relative performance of the algorithms depend on whether the budget is integer or fractional.

Keywords Adjustable robust optimization · Uncertainty polytope · Benders decomposition · Mixed-integer linear programming · Network design

1 Introduction

Robust Optimization is now a well-developed paradigm to tackle optimization problems under uncertainty. The framework has experienced its revival in the late nineties independently by Ben-Tal and Nemirovski (1998); El Ghaoui et al (1998); Kouvelis and Yu (1997), and has witnessed an increasing attention in

J. Ayoub
CDS Consultant at Murex, Lebanon
E-mail: josetteayoub@hotmail.com

M. Poss
UMR CNRS 5506 LIRMM, Université de Montpellier, rue Ada 161, 34095 Montpellier cedex 5, France
E-mail: michael.poss@lirmm.fr

the past twenty years. Its essence lies in the use of convex sets to model uncertainty that can arise when solving optimization problems. In the robust counterpart of an optimization problem, the constraints involving uncertain parameters must be feasible for all values of the uncertain parameters in the convex sets. In particular, the optimization variables are fixed independently of the values taken by the uncertain parameters; it is not possible to adjust them after the uncertainty is known. When the problem constraints are linear, this approach leads to tractable optimization problems. For instance, the robust counterparts of linear constraints subject to polyhedral uncertainty are still linear constraints.

The framework can fail to model design problems that involve actions that are delayed in time, such as network design problems or facility location problems, among many others. In each of these optimization problem, we must take part of the decisions today, e.g. implantation of new links or building facilities. Then, when the new links or facilities are operational, we must choose how to use them optimally to provide a service to the customers. In these problems, the demand of the customers is usually not known with precision until the links or the facilities are constructed.

Adjustable robust optimization has been introduced by Ben-Tal et al (2004) to improve over static robust optimization by allowing a subset of variables to account for the uncertainty. Namely, the framework partitions the optimization variables into two sets: part of them must fix their values before the uncertainty is revealed while the rest of them can adjust themselves according to the values taken by the uncertain parameters. These variables become functions defined on the uncertainty set. Ben-Tal et al (2004) prove that adjustable robust optimization is untractable in general, so that they focus on approximations that restrict the adjustable variables to affine functions of the uncertainty, yielding the so-called affine decision rules. The main advantage of affine decision rules is their tractability, since they lead to robust optimization problems with the structure of classical robust counterparts. The properties of affine decision rules have been studied in subsequent papers: Bertsimas et al (2011b) and Iancu et al (2013) study conditions in which affine decision rules are optimal, while Bertsimas and Goyal (2012) study the suboptimality of affine decision rules from a worst-case perspective. Authors have also studied more complex decision rules that offer more flexibility than affine decision rules while providing tractable optimization problems. Among others, Chen et al (2008) introduce *deflected linear* and *segregated linear* decision rules, Chen and Zhang (2009) propose to define affine decision rules on extended descriptions of the uncertainty set, Goh and Sim (2010) introduce complex piece-wise linear decision rules defined through liftings and projections, and Bertsimas and Georghiou (2015) propose piece-wise affine decision rules having fixed number of affine pieces. Alternatively, Bertsimas and Caramanis (2010) dynamically partition the uncertainty set and use constant decision rules in each set of the partition. The performance guarantee of the latter scheme is studied theoretically by Bertsimas et al (2011a). This idea has been revived recently by Bertsimas and Dunning (2014) and Postek and Den Hertog (2014) who extend it to multi-stage linear mixed-integer linear programs and test it numerically on different problems.

To assess numerically the quality of the aforementioned approximations, one needs to compute the (exact) optimal solutions, at least on small instances. Hence, in contrast with approximations approaches, some authors have tried to solve exactly the adjustable problems. Indeed, when the robust constraints are linear and the uncertainty set is a polytope, we know that the latter can be replaced by the finite set of its extreme points. This reformulation as such is not very useful because the number of extreme points is usually prohibitive. However, recent works have proposed decomposition algorithms that generate the extreme points on the fly. The first work in that line of research was carried out by Bienstock and Özbay (2008) who propose cutting plane algorithms (denoted RG in the following) for computing optimal base-stock levels in a supply chain. Similar approaches have been used subsequently by Mattia (2013) who study a network design problem with integer link capacities under demand uncertainty, by Gabrel et al (2014) who study a facility location problem under demand uncertainty, and by Bertsimas et al (2013) who study a unit commitment problem under nodal injection uncertainty. In these four papers, the uncertainty is limited to the right-hand side of the constraints and problem specific algorithms are proposed. Zeng and Zhao (2013) improved over the previous papers by proposing a row-and-column generation algorithm (denoted RCG in the following) and comparing the latter numerically to RG. Their results show that RCG can be up to three order of magnitudes faster than RG. The idea behind RCG has also been combined with RG heuristically by Bertsimas et al (2013). More general problems (based on algorithm RG) have also been considered by Billionnet et al (2014).

This papers contributes to this line of research, proposing an alternative approach to solve exactly two-stage robust linear programs with continuous second stage variables. We implement cutting plane algorithms and row-and-column generation algorithms very close to those proposed by Zeng and Zhao (2013), with the difference that we consider the separation problem as a feasibility problem instead of a min-max problem as done by Zeng and Zhao (2013). Using the Farkas' lemma, we can reformulate the feasibility problem into a bilinear program. Whenever the uncertainty set can be obtained as an affine projection of a $0-1$ polytope, which is the case for the budgeted uncertainty polytope from Bertsimas and Sim (2004), the bilinear program can be linearized to obtain a mixed-integer linear program. We assess our algorithms on a difficult telecommunication network design problem that has previously been studied in the literature by Poss and Raack (2013), comparing our results with the affine decision rules.

2 Problem overview

We consider in this paper the following type of two-stage robust optimization problems:

$$\begin{aligned} & \min \quad c \cdot x && (1) \\ & \text{s.t.} \quad x \in \mathcal{S} \\ (P) \quad & \quad T(\xi)x + Wy(\xi) \geq h, \quad \xi \in \Xi && (2) \end{aligned}$$

where $u \cdot v$ denotes the scalar product between any pair of vectors u and v, $c \in \mathbb{R}^{|I|}$, $W \in \mathbb{R}^{|M|\times|J|}$, $h \in \mathbb{R}^{|M|}$, $\mathcal{S} \subset \mathbb{R}^{|I|}$ denotes the first-stage feasibility polyhedron, $\Xi \subset \mathbb{R}^{|K|}$ denotes the uncertainty polytope, $T(\xi) \in \mathbb{R}^{|M|\times|I|}$ denotes the realization of the uncertain first-stage coefficient matrix, and $y(\xi)$ denotes the second-stage decision vector. We follow a classical assumption from the robust optimization literature and suppose that T depends *affinely* on uncertain parameter ξ. Hence, there exists matrices T^0 and T^{1k} with the same dimensions as T such that

$$T(\xi) := T^0 + \sum_k T^{1k}\xi^k.$$

One readily sees that (P) encompasses more general problems where (i) h depends affinely on ξ and (ii) second stage variables y have fixed costs given by vector k. Similarly, the approaches presented in this paper can be applied (with minor modifications) to problems where set $\mathcal{S}$ is the intersection of $\mathbb{Z}^{|I_1|} \times \mathbb{R}^{|I_2|}$ and a polyhedron.

When Ξ is not a singleton, problem (P) is a linear program that contains an infinite number of variables, since y is defined for each $\xi \in \Xi$, as well as an infinite number of constraints (2). The ideas presented in this paper rely on considering only the extreme points of Ξ, which is formalized in the result below (whose proof can be found in Ben-Tal and Nemirovski (2002), among others).

Lemma 1 *Let* $\text{vert}(\Xi)$ *be the set of extreme points of* Ξ *and* $x \in \mathbb{R}^{|I|}$ *a given vector. Vector* x *can be extended to an optimal solution* (x, y) *to* (P) *if and only if it can be extended to an optimal solution* (x, y') *to*

$$\begin{aligned} & \min \quad c \cdot x \\ & s.t. \quad x \in \mathcal{S} \\ (P') \quad & \quad T(\xi)x + Wy'(\xi) \geq h \quad \xi \in \text{vert}(\Xi). && (3) \end{aligned}$$

We provide next an example showing that Lemma 1 may not hold if the recourse matrix W depends on the uncertainty parameters ξ, which implies that our approach cannot handle problems with random recourse. Consider a unique robust constraint defined by $W(\xi) = -1 + 2\xi$, $T(\xi) = -2 + 3\xi$, $h = 0$, and $\Xi = [0, 1]$, and consider $x^* = 1$. The constraint becomes:

$$\xi = 0 \quad \Rightarrow \quad 2 - y(0) \geq 0 \tag{4}$$
$$\xi = 0.5 \quad \Rightarrow \quad -0.5 \geq 0 \tag{5}$$
$$\xi = 1 \quad \Rightarrow \quad 1 + y(1) \geq 0. \tag{6}$$

Constraints (4) and (6) are feasible while constraint (5) is infeasible. Hence, x^* is feasible for the constraints induced by each $\xi \in \text{vert}([0, 1])$ but infeasible for $\xi = 0.5$, providing a counter-example to Lemma 1 when W is an affine function of ξ.

Let $\mathcal{K}$ denote the projection of the set defined by (3) on variables x. Set $\mathcal{K}$ is a polyhedron. Hence, if a vector x does not belong to $\mathcal{K}$, there exists a separating hyperplane between x and $\mathcal{K}$. We study in the following section how to find such a hyperplane and how to use it to solve (P').

3 Solution approach

We propose in this section an algorithmic framework to solve (P') by generating scenarios in vert(Ξ) on the fly. First, we provide in Section 3.1 a mathematical program for the following separation problem: does x belong to $\mathcal{K}$? Then, assuming that the separation problem can be solved adequately, we describe in Section 3.2 two algorithms for addressing (P').

3.1 Separation

A naive approach to the separation problem would enumerate all elements of vert(Ξ). This is not suitable for practical problems since Ξ is likely to have a very large number of extreme points. We propose instead to address the problem by solving a mathematical program. The next result is based on Farka's lemma.

Theorem 1 *Let $x^* \in \mathbb{R}^n$ be given. Vector x^* belongs to $\mathcal{K}$ if and only if the optimal solution of the following optimization problem is non-positive*

$$(SP) \quad \begin{array}{ll} \max & (h - T(\xi)x^*) \cdot \pi \\ s.t. & \xi \in \Xi \\ & W^T \pi = 0 \\ & \mathbf{1} \cdot \pi = 1 \\ & \pi \geq 0. \end{array} \tag{7}$$

Before proving the Theorem, we introduce without proof a well-known property of bilinear optimization.

Lemma 2 *Let $\mathcal{P}$ be a polytope, $\mathcal{Q}$ a closed and bounded set, and $f(p,q)$ a bilinear function. It holds that*

$$\{\max f(p,q) \ s.t. \ p \in \mathcal{P}, q \in \mathcal{Q}\} = \{\max f(p,q) \ s.t. \ p \in \text{vert}(\mathcal{P}), q \in \mathcal{Q}\}.$$

Proof (Proof of Theorem 1) Consider a given vector $x^* \in \mathbb{R}^n$. Because x^* is fixed, constraints (3) become separable for each $\xi \in \text{vert}(\Xi)$. Hence, for each $\xi \in \text{vert}(\Xi)$, vector $y(\xi)$ must satisfy to

$$Wy(\xi) \geq h - T(\xi)x^*. \tag{8}$$

Let $\pi(\xi)$ be the dual multipliers associated to constraints (8). Using Farkas' Lemma, we know that constraints (8) have a solution if and only if

$$(h - T(\xi)x^*) \cdot \pi(\xi) \leq 0$$

for all $\pi(\xi)$ that satisfy

$$W^T\pi(\xi) = 0 \tag{9}$$

$$\pi(\xi) \geq 0. \tag{10}$$

Notice that the coefficients of constraints (9) and (10) do not depend on ξ. Hence, considering Farkas' conditions for all $\xi \in \text{vert}(\Xi)$ simultaneously, we obtain that constraints (8) for each $\xi \in \text{vert}(\Xi)$ are consistent if and only if the optimal solution of

$$\begin{aligned} \max \quad & (h - T(\xi)x^*) \cdot \pi && (11) \\ \text{s.t.} \quad & \xi \in \text{vert}(\Xi) && (12) \\ & W^T\pi = 0 \\ & \pi \geq 0 \end{aligned}$$

is non-positive. Adding normalization constraint

$$\mathbf{1} \cdot \pi = 1$$

to the problem above does not impact the sign of the optimal solution of the above optimization problem. Finally, we can replace (12) by $\xi \in \Xi$ because objective (11) is bilinear in ξ and π and the constraints of ξ and π are independent from each other, so that Lemma 2 holds. □

The objective function of (SP) is bilinear so that (SP) belongs to a class of problems NP-hard to solve exactly (Matsui, 1996). Unreported results show that algorithms based on spatial branching (Ryoo and Sahinidis, 2003) are unable to cope with (SP) even for small instances. Hence, we propose to address (SP) through mixed-integer linear reformulations, see Section 4.

3.2 Algorithms

We propose in this section two solution algorithms for (P'), denoted by *RG* and *RCG*, assuming that we can solve (SP) through a black-box method. The algorithms are both based on generating dynamically a subset $\hat{\Xi} \subset \text{vert}(\Xi)$. Both algorithms start by solving the following master problem

$$(MP) \quad \begin{array}{ll} \min & c \cdot x \\ \text{s.t.} & x \in \mathcal{S}. \end{array}$$

Given an optimal solution to (MP), the algorithms solve the separation problem (SP), yielding an optimal solution (ξ^*, π^*). The two algorithms differ then by the way they include more information to (MP) when the solution cost of (SP) is positive. Algorithm *RG* is a classical Benders' decomposition approach that adds a violated Benders' cut to the master problem

$$(h - T(\xi^*)x) \cdot \pi^* \leq 0. \tag{13}$$

The drawback of *RG* is that a single cut is added to the master problem at each iteration, facing the risk that many iterations may be required before obtaining a feasible solution for (P'). To incorporate more information to the master problem at each iteration, Algorithm *RCG* adds to (MP) all constraints and variables associated to ξ^*

$$T(\xi^*)x + Wy(\xi^*) \geq h. \tag{14}$$

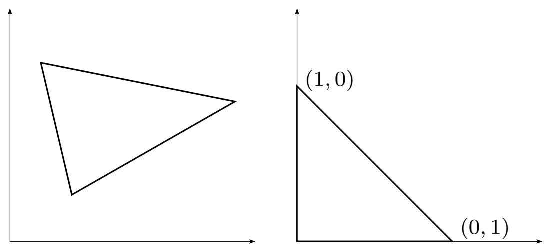
For completeness, both algorithms are described formally in Algorithm 1.

Algorithm 1: *RG* and *RCG*

```
repeat
    solve (MP);
    let x* be an optimal solution;
    solve (SP);
    let (ξ*, π*) be an optimal solution and z* be the optimal solution cost;
    if z* > 0 then
        RG: add constraint (13) to (MP);
        RCG: add constraint (14) to (MP);
until z* > 0;
```

4 Mixed-integer programming reformulations

In this section, we propose two mixed-integer linear reformulations for (SP). Recall that a polytope Ω is called a $0-1$ polytope if each of its extreme points is a binary vector. Our first mixed-integer linear formulation for (SP) is based on reformulating Ξ as the affine mapping of a $0 - 1$ polytope $\Omega \subset \mathbb{R}^{|K'|}$ of



(a) Ξ (b) $\Omega = a(\Xi)$

Fig. 1 Example of affine mapping such that $\Omega = a(\Xi)$ and Ω is a $0-1$ polytope.

reasonable dimension $|K'|$. Notice that such a mapping always exists since we can represent Ξ as the set of all convex combinations of the vectors in $\mathrm{vert}(\Xi)$. This example is useless, however, because the associated polytope Ω would be the unit simplex of dimension $|\mathrm{vert}(\Xi)|$. A more useful example of pair of polytopes Ξ and Ω is given in Figure 1.

Theorem 2 *Let $x^* \in \mathbb{R}^{|I|}$ be given and suppose that we know a $0-1$ polytope $\Omega \subset \mathbb{R}^{|K'|}$ and an affine mapping $a(\omega) = A\omega + b$ such that $\Xi = a(\Omega)$. Vector x^* belongs to $\mathcal{K}$ if and only if the optimal solution of the following optimization problem is non-positive*

$$
\begin{aligned}
& \max && (h - \tilde{T}^0 x^*) \cdot \pi - \sum_{k \in K} (\tilde{T}^{1k} x^*) \cdot v^k \\
(SPL) \quad & s.t. && \omega \in \Omega \\
& && W^T \pi = 0 \\
& && \mathbf{1} \cdot \pi = 1 \\
& && v_m^k \geq \pi_m - (1 - \omega^k) \qquad k \in K', m \in M \qquad (15) \\
& && v_m^k \leq \omega^k \qquad k \in K', m \in M \qquad (16) \\
& && \pi, v_m^k \geq 0, \\
& && \omega \in \{0,1\}^{|K'|}
\end{aligned}
$$

where $\tilde{T}^{1k} = \sum_{h \in K} T^{1h} A_{hk}$ *for each* $k \in K$ *and* $\tilde{T}^0 = T^0 + \sum_{k \in K} T^{1k} \sum_{h \in K} A_{kh} b^k$.

Proof The proof consists of two steps. First, we take (SP) and make the change of variable $\xi = a(\omega)$:

$$
\begin{aligned}
\max \quad & (h - \tilde{T}(\omega) x^*) \cdot \pi && (17) \\
\text{s.t.} \quad & \omega \in \Omega && (18) \\
& W^T \pi = 0 \\
& \mathbf{1} \cdot \pi = 1 \\
& \pi \geq 0.
\end{aligned}
$$

Then, using Lemma 2, we replace constraint (18) by $\omega \in \text{vert}(\Omega)$. Because Ω is a $0-1$ polytope, we can add binary restrictions $\omega \in \{0,1\}^{|K'|}$ to the problem. Hence, each product $\omega^k \pi_m$ involved in objective function (17) can be reformulated by introducing auxiliary variables v_m^k and using big-M coefficients. Finally, the big-M coefficients can be set to 1, because $0 \leq \pi_m \leq 1$, which yields constraints (15) and (16). □

An interesting example of non-bijective affine mapping arises with the budgeted uncertainty set introduced in Bertsimas and Sim (2004):

$$\Xi^\Gamma \equiv \left\{ \xi \in [0,1]^{|K|} \text{ s.t. } \sum_{k \in K} \xi^k \leq \Gamma \right\}.$$

Polytope Ξ^Γ is a $0-1$ polytope only when Γ is integer. Nevertheless, Ξ^Γ for fractional values of Γ can be obtained as the affine transformation of the following polytope

$$\Omega^\Gamma \equiv \left\{ (\omega^1, \omega^2) \in [0,1]^{2|K|} \text{ s.t. } \omega^{1k} + \omega^{2k} \leq 1, k \in K, \sum_{k \in K} \omega^{1k} \leq \lfloor \Gamma \rfloor, \sum_{k \in K} \omega^{2k} \leq 1 \right\},$$

using mapping $a(\omega) = \omega^1 + (\Gamma - \lfloor \Gamma \rfloor)\omega^2$. One readily sees that Ω^Γ is a $0-1$ polytope.

A different approach has been used by Mattia (2013) for a specific network design problem. The result below generalizes the one of Mattia (2013) in two aspects: we do not assume that $\mathcal{S}$ has a specific form, and we allow the first-stage constraint matrix to depend on the uncertainties (while Mattia (2013) only considers right-hand-side uncertainty).

Theorem 3 *Let $x^* \in \mathbb{R}^n$ be given and let $\Xi = \{\xi \in \mathbb{R}^{|K|}$ s.t. $B\xi \leq b, \xi \geq 0\}$. Vector x^* belongs to $\mathcal{K}$ if and only if the optimal solution of the following optimization problem is non-positive*

$$\begin{aligned}
\max \quad & (h^0 - T^0 x^*) \cdot \pi + b \cdot \gamma & \\
s.t. \quad & W^T \pi = 0 & \\
& \mathbf{1} \cdot \pi = 1 & \\
& \pi \geq 0 & \\
& B^T \gamma \geq \sum_{k \in K} u^k (T^{1k} x^*) & B\xi \leq b \\
& \gamma \geq 0 & \xi \geq 0 \\
& -B^T \gamma - M\mu \geq -M\mathbf{1} - \sum_{k \in K} u^k (T^{1k} x^*) & \xi - M\mu \leq 0 \\
& \gamma + M\nu \leq M\mathbf{1} & -B\xi - M\nu \leq -b \\
& \mu \in \{0,1\}^n, \nu \in \{0,1\}^m &
\end{aligned}$$

Proof Introducing notations $\tau^0(x^*) := h^0 - T^0 x^*$ and $\tau^{1k}(\pi, x^*) := (T^{1k} x^*) \cdot \pi$ for each $k \in K$, $(h - T(\xi)x^*)$ can be rewritten as

$$\tau^0(x^*) \cdot \pi + \tau^1(\pi, x^*) \cdot \xi, \tag{19}$$

where only the second term of (19) depends on ξ. Hence, for any polyhedron Π, we have that

$$\max_{\pi \in \Pi, \xi \in \Xi} \tau^0(x^*) \cdot \pi + \tau^1(\pi, x^*) \cdot \xi = \max_{\pi \in \Pi} \left(\tau^0(x^*) \cdot \pi + \max_{\xi \in \Xi} \tau^1(\pi, x^*) \cdot \xi \right)$$

so that we can rewrite (SP) as a bilevel program sending ξ to the follower's decisions:

$$\begin{aligned} \max \quad & \tau^0(x^*) \cdot \pi + \tau^1(\pi, x^*) \cdot \xi \\ \text{s.t.} \quad & W^T \pi = 0 \\ & \mathbf{1} \cdot \pi = 1 \\ & \pi \geq 0 \\ & \xi \in \operatorname{argmax} \ \tau^1(\pi, x^*) \cdot \xi \\ & \qquad \xi \in \Xi \end{aligned} \tag{20}$$

Recall that Ξ is described by $\{\xi : B\xi \leq b, \xi \geq 0\}$. Because Ξ is bounded and non-empty, we know by linear programming duality that

$$\{\max \tau^1(\pi, x^*) \cdot \xi : B\xi \leq b, \xi \geq 0\} = \{\min b \cdot \gamma : B^T \gamma \geq \tau^1(\pi, x^*), \gamma \geq 0\}.$$

Then, replacing the follower's problem by its dual and substituting the bilinear term $\tau^1(\pi, x^*) \cdot \xi$ in the objective function (20) with the objective function of the dual, $b \cdot \gamma$, we obtain:

$$\begin{aligned} \max \quad & \tau^0(x^*) \cdot \pi + b \cdot \gamma \\ \text{s.t.} \quad & W^T \pi = 0 \\ & \mathbf{1} \cdot \pi = 1 \\ & \pi \geq 0 \\ & \gamma \in \operatorname{argmax} \ -b \cdot \gamma \\ & \qquad B^T \gamma \geq \tau^1(\pi, x^*) \\ & \qquad \gamma \geq 0. \end{aligned} \tag{21}$$

The result finally follows from applying the mixed-integer reformulation from (Audet et al, 1997, Corollary 3.2) to the bilevel problem above. □

Notice that in spite of its generality, the reformulation from Theorem 3 can be very hard to solve to optimality (Mattia, 2014). Hence, it should be combined with heuristic separation of violated extreme points as done by Mattia (2013) for a network loading problem.

5 Application to telecommunications network design

We illustrate in this section the performance of Algorithm 1 on an adjustable robust optimization problem previously studied in the literature.

5.1 Problem description

Telecommunications networks have evolved quickly, especially in the last two decades. This is due to the expansion of the Internet, on-line gaming, instant messaging, file sharing and plenty of other applications requiring fast and reliable communication technologies. This has resulted in increasing demands for higher bit-rates. Network operators are then under constant pressure to design high speed networks with larger capacities such as to satisfy the growing need for fast connections with no interruptions and latency. However, network operators also tend to design networks while minimizing capital costs and taking into account that resources are limited. Given a directed graph (V, A) and a set of point-to-point commodities K, network design can be defined as a planning process that involves setting up link capacities x_a for each $a \in A$ and traffic routing y_a^k for each $a \in A$ and $k \in K$ in order to route data packets for each commodity $k \in K$ from its source s^k to its destination t^k. Its goal is to minimize the total capacity cost $\sum_{a \in A} c_a x_a$ while satisfying all traffic demands d^k for $k \in K$.

Traditional models for network design ignored demand uncertainty and overestimated demand values to avoid blockages. This led to a waste in network capacities and investments. In order to obtain a more robust and efficient network, demand fluctuation has to be taken into consideration throughout the design procedure. As a result, the idea of demand uncertainty must be included in the mathematical program that models the planning process of the network.

Robust network design considers that the vector of demands d is uncertain and depends affinely on the uncertain parameters ξ that can take any value in a predetermined uncertainty polytope. The problem becomes an adjustable robust optimization problem: capacities x are first-stage decision variables while routings y are second-stage decision variables. Let δ_v^+ and δ_v^- be the sets of outgoing arcs and incoming arcs, respectively, for each $v \in V$. The mathematical program for the problem is given below.

$$\min \quad \sum_{a\in A} c_a x_a \tag{22}$$

$$(RND) \quad \text{s.t.} \quad \sum_{a\in\delta_v^-} y_a^k(\xi) - \sum_{a\in\delta_v^+} y_a^k(\xi) = \begin{cases} -d^k(\xi) \text{ if } v = s^k \\ d^k(\xi) \text{ if } v = t^k \\ 0 \text{ otherwise} \end{cases} \quad v \in V, k \in K \tag{23}$$

$$\sum_{k\in K} y_a^k(\xi) \le x_a \quad a \in A \tag{24}$$

$$x \ge 0, y \ge 0.$$

Objective function (22) minimizes the total capacity installation cost subject to flow balance constraints (23) and capacity constraints (24).

The above optimization problem has been studied previously by Poss and Raack (2013). Poss and Raack (2013) solves (RND) by enumerating the extreme points of Ξ using Lemma 1. This approach is of course restricted to uncertainty polytopes having limited number of extreme points and they address larger problems by applying affine decision rules to routing variables y:

$$y^k(\xi) = f^{0k} + \sum_{h\in K} f^{hk}\xi^h, \quad k \in K. \tag{25}$$

Other approximations of (RND) have been considered in the literature, see Poss (2013) and the references therein. Among them, static routing which enforces y^k to depend linearly on ξ^k by adding constraints

$$y^k(\xi) = f^k \xi^k, \quad k \in K \tag{26}$$

to (RND), where f are non-negative optimization variables. It must be said that (RND) with static or affine routing is much easier to solve than (RND) because the problem can then be reformulated as a static robust optimization problem, which can be solved by dualizing the robust constraints.

In the next section, we apply the two versions of Algorithm 1 to the instances studied by Poss and Raack (2013).

5.2 Numerical results

We consider three realistic networks from SNDlib (Orlowski et al, 2010) : janos-us, sun, and giul-39. These networks have 26/27/39 nodes and 84/102/172 arcs, respectively. The networks are originally undirected and we direct them by replacing each edge by two arcs with opposite directions. To reduce the size of the formulations and to be able to do a series of runs, we considered the largest 20-50 commodities with respect to the mean value of $\overline{d}^k$. Our uncertainty set is based on the budgeted uncertainty polytope from Bertsimas and Sim

(2004). Namely, $d^k(\xi) = \overline{d}^k + \xi^k \hat{d}^k$ where the deviation $\hat{d}^k$ is set to $0.4\overline{d}^k$, and $\Xi \equiv \{\xi \in \mathbb{R}^{|K|} \text{ s.t. } \sum_{k\in K} \xi^k \leq \Gamma, 0 \leq \xi^k \leq 1 \text{ for each } k \in K\}$. Polytope Ξ is a $0-1$ polytope so that we can reformulate the subproblem as (SPL) from Theorem 2. Notice that our uncertainty set does not consider downward deviations. Hence, Ξ corresponds to $\mathcal{D}^{\sigma}_{+}$ used in Poss and Raack (2013). We consider two sets of instances. In the first set, Γ is integer and belongs to $\{1, 2, \ldots, 6\}$. In the second set, Γ is fractional and belongs to $\{1.5, 2.5, \ldots, 5.5\}$. Our algorithms have been coded using the Concert Technology library for JAVA from CPLEX 12.6 (IBM-ILOG, 2015) on a 64-bit 2.53Ghz Quad-Core CPU with 6GB of memory and 16 threads. CPLEX is called with its default parameters and we have set a time limit of $\mathbf{T}$ seconds of CPU time for every individual run and our computing times are presented in seconds.

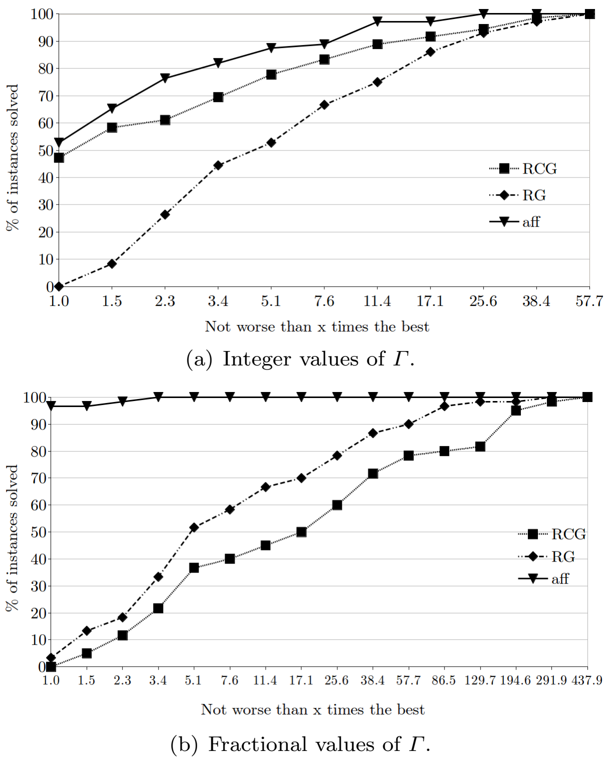
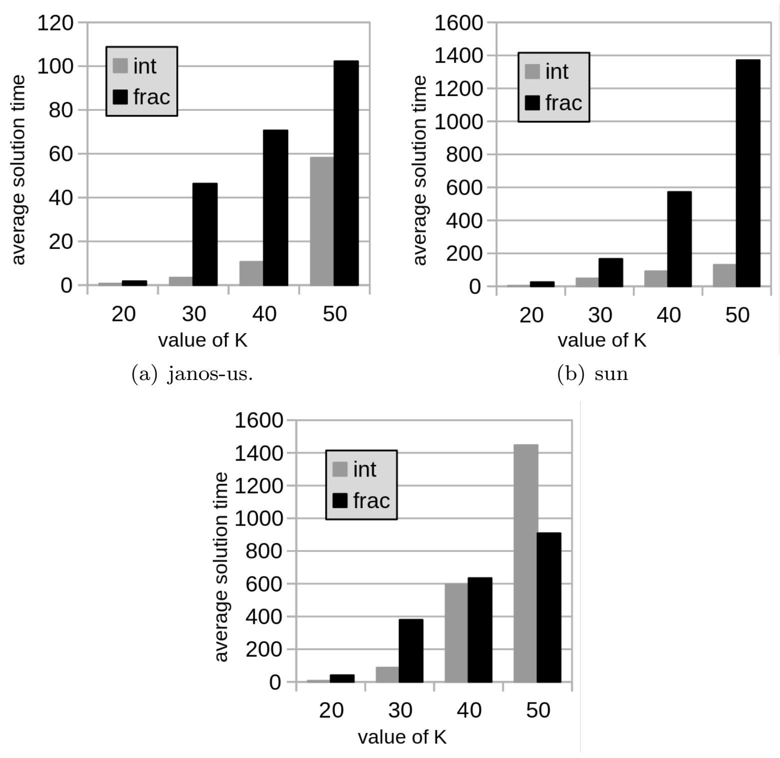


Fig. 2 Performance profiles comparing the solution times of RG, RCG, and aff.

Figure 2 shows performance profiles (Dolan and Moré, 2002) that compare the solution times of algorithms RG and RCG as well as affine routing (denoted aff). Figure 2(a) reports the profile for $\Gamma \in \{1, 2, \ldots, 6\}$ while Figure 2(b) reports the profile for $\Gamma \in \{1.5, 2.5, \ldots, 5.5\}$. Notice that the solution times of unsolved instances have been set to $\mathbf{T}$ seconds, so that the right part of the profiles should be considered as approximations of the *true* relative performance of the algorithms. Despite this, one can observe from the profiles that RG and RCG behave differently on the two sets of instances. Namely, when Γ is integer, RCG clearly outperforms RG, and the performance of RCG is close to the one of affine routing. In contrast, when Γ is fractional, RCG is outperformed by RG and both algorithms behave much worse than affine routing. This behavior can be explained by the following observations. First, the solution times of (SPL) is usually higher when Γ is fracional (see below). Second, algorithm RCG generates much more extreme points when Γ is fractional. Hence, the time spent solving (MP) increases, making it an important part of the total solution time. In contrast, the time spent solving (MP) is insignificant for RG, which concentrates all of its effort on solving (SPL).



(a) janos-us. (b) sun

(c) giul-39

Fig. 3 Average solution times in seconds for (SPL) for each value of $K \in \{20, \ldots, 50\}$. Integer and fractional values of Γ are reported in gray and black, respectively.

We present the extensive computational results on Tables 1 and 2. The columns of these tables report the number of commodities (K), the value of Γ, the optimal solution cost for static routing (opt_{stat}), the cost reduction (gap_{aff}) when using affine routing constraints (25) instead of static routing constraints (26), the cost reduction (gap_{dyn}) for the true optimal solution of (RND), the time required by RCG (t_{RCG}) and RG (t_{RG}), the number of iterations of both algorithms ($iter$), and the time that they spent solving (SPL) as a percentage of the total solution time (t_{SPL} (%)). We also report the solution times spent to solve the problem with affine routing (t_{aff}) and to solve the problem that contains all extreme points ($t_{P'}$), although this last model very often consumes all memory available. Time and memory hits are denoted by **T** and **M**, respectively. Table 1 confirms the results summarized in Figure 2(a); that is, RCG is more efficient than RG on instances with integer values of Γ, solving the instances faster than the latter. In contrast Table 2 shows that RCG can solve only two instances with fractional values of Γ, while RG can solve roughly half of them. The tables also show that both algorithms spend most of their time in the solution of (SPL) apart from RCG for the instances with high numbers of iterations. Figure 3 reports the arithmetic averages of the solution times for solving (SPL) for each instance, which are based on the separations problems that were solved to optimality. Figure 3(a) and Figure 3(b) show that for the instances janos-us and sun, the solution times of (SPL) is much higher for fractional values of Γ than for the integer ones. In contrast, Figure 3(b) shows the opposite behavior for sun, particularly when $K = 50$.

6 Conclusion

We propose in this paper decomposition algorithms to solve adjustable robust linear programs under polytope uncertainty by generating the extreme points of the uncertainty polytope on the fly. We discuss algorithmic frameworks for decomposing the robust problem and solution algorithms to address the subproblem. Our numerical results show that the relative efficiency of row generation and row-and-column generation algorithms depends on the type of polytope considered. In any case, the bottleneck of these algorithms lies in finding the extreme point of the polytope that is mostly violated by the current solution, which amounts to solve a mixed-integer linear program (MIP).

Aknowledgements

We would like to thank Sara Mattia for fruitful discussions on the topic of this paper. We also thank the editors and the referees for useful remarks and suggestions that helped in improving the presentation and the content of the paper.

References

Audet C, Hansen P, Jaumard B, Savard G (1997) Links between linear bilevel and mixed 0-1 programming problems. J Optim Theory Appl 93:273–300

instance	K	Γ	opt_{stat}	gap_{dyn}(%)	gap_{aff}(%)	t_{RCG}	t_{SPL} (%)	iter	t_{RG}	t_{SPL} (%)	iter	$t_{P'}$	t_{aff}
janos-us	20	1.0	4,65E5	7.2	7.2	9	81	10	29	100	73	2	84
	20	2.0	5,12E5	6.5	6.2	36	70	18	61	100	67	7	78
	20	3.0	5,12E5	2.5	0.9	33	67	18	76	100	84	55	68
	20	4.0	5,12E5	0.0	0.0	40	63	20	57	100	70	293	47
	20	5.0	5,12E5	0.0	0.0	21	80	13	34	100	55	**M**	44
	20	6.0	5,12E5	0.0	0.0	22	71	15	37	100	59	**M**	37
	30	1.0	6,12E5	7.5	7.5	30	89	12	77	100	85	7	345
	30	2.0	6,72E5	8.7	8.4	85	82	19	170	100	86	22	490
	30	3.0	6,99E5	7.0	6.6	137	85	21	235	100	82	307	401
	30	4.0	6,99E5	2.9	2.4	245	81	27	474	100	92	**M**	371
	30	5.0	6,99E5	0.7	0.0	276	83	27	587	100	87	**M**	290
	30	6.0	6,99E5	0.0	0.0	139	83	22	335	100	84	**M**	252
	40	1.0	6,72E5	8.2	8.2	52	80	14	130	100	90	34	567
	40	2.0	7,32E5	8.8	8.4	205	76	22	376	100	91	63	988
	40	3.0	7,63E5	7.6	6.7	432	82	25	747	100	88	**M**	1490
	40	4.0	7,66E5	4.1	2.9	977	80	32	2231	100	98	**M**	1865
	40	5.0	7,66E5	1.5	0.2	1300	80	35	2186	100	100	**M**	2427
	40	6.0	7,66E5	0.0	0.0	1036	77	34	1372	100	89	**M**	1213
	50	1.0	7,32E5	8.4	8.4	90	81	15	251	100	95	103	2511
	50	2.0	7,93E5	8.9	8.5	464	76	25	794	100	98	122	4494
	50	3.0	8,27E5	8.2	7.3	1836	91	28	2776	100	101	**M**	5454
	50	4.0	8,39E5	6.0	4.9	4188	93	33	10093	100	104	**M**	4419
	50	5.0	8,41E5	3.2	2.2	4079	88	39	**T**	100	96	**M**	3529
	50	6.0	8,41E5	0.8	0.1	6309	78	50	**T**	100	99	**M**	4866
sun	20	1.0	4,31E2	9.9	9.9	22	88	12	215	100	152	1	50
	20	2.0	4,67E2	9.2	8.9	90	81	23	429	100	160	8	58
	20	3.0	4,82E2	6.4	5.5	207	84	29	586	100	149	74	63
	20	4.0	4,87E2	3.5	2.0	230	85	29	730	100	161	439	82
	20	5.0	4,88E2	1.8	0.7	326	82	34	826	100	156	**M**	36
	20	6.0	4,88E2	1.0	0.2	234	85	29	595	100	145	**M**	34
	30	1.0	5,56E2	9.3	9.3	55	93	13	766	100	200	5	224
	30	2.0	6,03E2	10.5	10.1	347	83	31	1413	100	190	22	248
	30	3.0	6,3E2	9.6	8.8	1790	90	43	4009	100	174	319	293
	30	4.0	6,4E2	6.9	5.7	4921	98	36	**T**	100	161	**M**	282
	30	5.0	6,47E2	5.2	3.2	**T**	98	41	**T**	100	137	**M**	259
	30	6.0	6,49E2	4.2	1.9	**T**	100	24	**T**	100	104	**M**	311
	40	1.0	6,69E2	8.6	8.6	119	97	12	1908	100	227	12	867
	40	2.0	7,23E2	10.7	10.3	770	93	30	4313	100	209	54	840
	40	3.0	7,58E2	10.9	10.2	10038	98	42	**T**	100	138	1126	1028
	40	4.0	7,78E2	10.3	8.9	**T**	100	18	**T**	100	89	**M**	975
	40	5.0	7,91E2	11.6	7.2	**T**	100	9	**T**	100	78	**M**	1152
	40	6.0	7,96E2	11.4	5.5	**T**	100	8	**T**	100	62	**M**	1421
	50	1.0	7,34E2	8.0	8.0	139	98	11	2437	100	207	22	3059
	50	2.0	7,92E2	10.4	10.1	1498	97	27	9056	100	199	108	3539
	50	3.0	8,3E2	11.1	10.4	**T**	100	22	**T**	100	100	**M**	3347
	50	4.0	8,53E2	12.7	9.6	**T**	100	9	**T**	100	64	**M**	3828
	50	5.0	8,7E2	13.1	8.6	**T**	100	7	**T**	100	50	**M**	3487
	50	6.0	8,81E2	14.9	7.4	**T**	100	5	**T**	100	44	**M**	2927
giul-39	20	1.0	8,97E1	4.8	4.8	33	93	9	166	100	85	3	56
	20	2.0	9,93E1	6.2	6.2	62	95	10	383	100	93	13	64
	20	3.0	1,04E2	5.1	4.5	177	94	15	696	100	92	94	59
	20	4.0	1,07E2	3.1	2.3	338	95	18	973	100	102	735	83
	20	5.0	1,09E2	1.3	0.3	401	95	19	985	100	99	**M**	78
	20	6.0	1,09E2	0.8	0.0	571	95	21	731	100	96	**M**	46
	30	1.0	1,42E2	6.1	6.1	111	96	9	837	100	131	19	416
	30	2.0	1,52E2	7.4	7.4	366	95	14	2655	100	144	58	501
	30	3.0	1,61E2	7.9	7.7	2588	99	18	**T**	100	117	927	446
	30	4.0	1,68E2	7.3	7.2	10718	100	18	**T**	100	83	**M**	405
	30	5.0	1,72E2	6.0	5.1	**T**	100	14	**T**	100	92	**M**	407
	30	6.0	1,73E2	3.2	2.3	**T**	100	18	**T**	100	72	**M**	554
	40	1.0	1,9E2	6.9	6.9	469	94	16	3081	100	188	31	4372
	40	2.0	2,04E2	8.4	8.4	1480	97	18	**T**	100	181	119	3000
	40	3.0	2,15E2	10.4	8.6	**T**	100	11	**T**	100	67	**M**	3376
	40	4.0	2,22E2	12.6	8.1	**T**	100	6	**T**	100	31	**M**	2231
	40	5.0	2,27E2	21.7	7.2	**T**	100	2	**T**	100	13	**M**	2728
	40	6.0	2,3E2	12.6	5.3	**T**	100	5	**T**	100	5	**M**	3184
	50	1.0	2,21E2	8.0	8.0	900	96	16	5767	100	188	82	9680
	50	2.0	2,38E2	10.3	9.8	5982	99	20	**T**	100	139	221	7390
	50	3.0	2,49E2	13.6	9.8	**T**	100	8	**T**	100	63	**M**	7324
	50	4.0	2,57E2	18.6	9.8	**T**	100	3	**T**	100	13	**M**	6574
	50	5.0	2,63E2	23.8	9.1	**T**	100	2	**T**	100	7	**M**	5526
	50	6.0	2,68E2	–	7.9	**T**	100	1	**T**	100	1	**M**	5852

Table 1 Results for Γ integer.

instance	K	Γ	opt_{stat}	gap_{dyn}(%)	gap_{aff}(%)	t_{RCG}	t_{SPL} (%)	iter	t_{RG}	t_{SPL} (%)	iter	$t_{P'}$	t_{aff}
janos-us	20	1.5	4,91E5	7.2	6.8	**T**	7	399	95	100	78	16	81
	20	2.5	5,12E5	4.5	2.8	**T**	6	202	155	100	75	260	138
	20	3.5	5,12E5	1.2	0.0	**T**	16	387	178	100	82	**M**	53
	20	4.5	5,12E5	0.0	0.0	73	84	20	136	100	72	**M**	46
	20	5.5	5,12E5	0.0	0.0	81	81	20	96	100	64	**M**	39
	30	1.5	6,44E5	8.4	8.0	**T**	15	357	200	100	80	46	392
	30	2.5	6,89E5	11.1	7.8	**T**	64	242	894	100	84	**M**	308
	30	3.5	6,99E5	7.1	4.4	**T**	86	127	2230	100	85	**M**	383
	30	4.5	6,99E5	3.0	0.6	**T**	95	96	5004	100	90	**M**	566
	30	5.5	6,99E5	0.8	0.0	**T**	100	25	**T**	100	79	**M**	201
	40	1.5	7,04E5	12.4	8.3	**T**	21	251	538	100	97	**M**	511
	40	2.5	7,5E5	11.2	7.7	**T**	64	88	2859	100	91	**M**	1659
	40	3.5	7,64E5	7.9	4.6	**T**	82	55	9438	100	97	**M**	2248
	40	4.5	7,66E5	2.8	1.3	**T**	87	55	9079	100	89	**M**	1448
	40	5.5	7,66E5	1.5	0.0	**T**	94	49	**T**	100	94	**M**	577
	50	1.5	7,64E5	12.2	8.4	**T**	26	149	1148	100	105	**M**	3635
	50	2.5	8,13E5	11.4	8.1	**T**	91	46	**T**	100	102	**M**	4249
	50	3.5	8,33E5	9.1	5.9	**T**	99	27	**T**	100	88	**M**	5026
	50	4.5	8,4E5	6.3	3.4	**T**	99	25	**T**	100	79	**M**	4404
	50	5.5	8,41E5	3.3	1.1	**T**	99	27	**T**	100	80	**M**	4308
sun	20	1.5	4,51E2	10.0	9.6	**T**	16	444	610	100	180	21	65
	20	2.5	4,75E2	11.5	7.3	**T**	35	383	1798	100	172	277	57
	20	3.5	4,85E2	5.0	3.3	**T**	84	200	3504	100	159	**M**	56
	20	4.5	4,88E2	3.8	1.3	**T**	93	114	4807	100	151	**M**	68
	20	5.5	4,88E2	1.9	0.5	**T**	95	71	9673	100	165	**M**	33
	30	1.5	5,8E2	10.0	9.8	**T**	37	343	1446	100	174	61	254
	30	2.5	6,18E2	13.2	9.6	**T**	92	120	**T**	100	167	**M**	250
	30	3.5	6,35E2	10.7	7.2	**T**	100	28	**T**	100	89	**M**	419
	30	4.5	6,43E2	10.2	4.2	**T**	100	9	**T**	100	46	**M**	352
	40	1.5	6,96E2	9.8	9.5	**T**	73	227	5114	100	200	120	813
	40	2.5	7,41E2	13.1	10.3	**T**	100	30	**T**	100	96	**M**	988
	40	3.5	7,69E2	13.7	9.5	**T**	100	9	**T**	100	59	**M**	952
	40	4.5	7,85E2	13.8	8.0	**T**	100	6	**T**	100	34	**M**	854
	40	5.5	7,94E2	17.0	6.2	**T**	100	4	**T**	100	11	**M**	880
	50	1.5	7,64E2	11.5	9.2	**T**	88	141	**T**	100	195	**M**	3443
	50	2.5	8,12E2	12.8	10.3	**T**	100	15	**T**	100	75	**M**	3125
	50	3.5	8,42E2	15.2	10.1	**T**	100	5	**T**	100	35	**M**	3424
	50	4.5	8,62E2	17.2	9.0	**T**	100	4	**T**	100	5	**M**	3534
	50	5.5	8,76E2	20.0	8.0	**T**	100	3	**T**	100	3	**M**	3694
giul-39	20	1.5	9,45E1	5.5	5.5	**T**	17	213	329	100	94	29	62
	20	2.5	1,02E2	5.7	5.2	**T**	55	168	1465	100	94	479	71
	20	3.5	1,06E2	6.7	3.4	**T**	83	73	4706	100	96	**M**	71
	20	4.5	1,08E2	4.1	1.0	**T**	81	84	5819	100	102	**M**	84
	20	5.5	1,09E2	1.0	0.0	**T**	97	42	4940	100	107	**M**	51
	30	1.5	1,47E2	6.8	6.8	**T**	23	69	2116	100	141	159	393
	30	2.5	1,57E2	7.7	7.5	**T**	91	41	**T**	100	116	**M**	420
	30	3.5	1,64E2	10.0	7.5	**T**	100	11	**T**	100	45	**M**	425
	30	4.5	1,7E2	10.6	6.1	**T**	100	6	**T**	100	19	**M**	463
	30	5.5	1,72E2	11.4	3.7	**T**	100	4	**T**	100	15	**M**	448
	40	1.5	1,97E2	7.7	7.7	**T**	53	60	9010	100	200	**M**	2879
	40	2.5	2,09E2	10.9	8.4	**T**	100	17	**T**	100	49	**M**	3576
	40	3.5	2,18E2	12.3	8.4	**T**	100	8	**T**	100	15	**M**	2981
	40	4.5	2,25E2	12.2	7.7	**T**	100	8	**T**	100	14	**M**	3067
	40	5.5	2,29E2	14.0	6.2	**T**	100	5	**T**	100	11	**M**	3426
	50	1.5	2,3E2	11.2	9.0	**T**	88	39	**T**	100	166	**M**	8576
	50	2.5	2,43E2	12.9	9.8	**T**	100	11	**T**	100	21	**M**	10808
	50	3.5	2,53E2	17.3	9.8	**T**	100	4	**T**	100	10	**M**	6153
	50	4.5	2,6E2	17.8	9.4	**T**	100	4	**T**	100	11	**M**	6951
	50	5.5	2,66E2	15.0	8.6	**T**	100	6	**T**	100	9	**M**	7001

Table 2 Results for Γ fractional.

Ben-Tal A, Nemirovski A (1998) Robust convex optimization. Mathematics of Operations Research 23(4):769–805

Ben-Tal A, Nemirovski A (2002) Robust optimization methodology and applications. Mathematical Programming 92:453–480

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

Bertsimas D, Caramanis C (2010) Finite adaptability in multistage linear optimization. Automatic Control, IEEE Transactions on 55(12):2751–2766

Bertsimas D, Dunning I (2014) Multistage robust mixed integer optimization with adaptive partitions. Under review

Bertsimas D, Georghiou A (2015) Design of near optimal decision rules in multistage adaptive mixed-integer optimization. Operations Research Accepted

Bertsimas D, Goyal V (2012) On the power and limitations of affine policies in two-stage adaptive optimization. Mathematical Programming 134(2):491–531

Bertsimas D, Sim M (2004) The price of robustness. Operations Research 52:35–53

Bertsimas D, Goyal V, Sun XA (2011a) A geometric characterization of the power of finite adaptability in multistage stochastic and adaptive optimization. Math Oper Res 36(1):24–54

Bertsimas D, Iancu D, Parrilo P (2011b) A hierarchy of near-optimal policies for multistage adaptive optimization. Automatic Control, IEEE Transactions on 56(12):2809–2824

Bertsimas D, Litvinov E, Sun XA, Zhao J, Zheng T (2013) Adaptive robust optimization for the security constrained unit commitment problem. IEEE Transactions on Power Systems 28(1)

Bienstock D, Özbay N (2008) Computing robust basestock levels. Discrete Optimization 5(2):389–414

Chen X, Zhang Y (2009) Uncertain linear programs: Extended affinely adjustable robust counterparts. Operations Research 57(6):1469–1482

Chen X, Sim M, Sun P, Zhang J (2008) A linear decision-based approximation approach to stochastic programming. Operations Research 56(2):344–357

Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. Mathematical Programming 91(2):201–213

El Ghaoui L, Oustry F, Lebret H (1998) Robust solutions to uncertain semidefinite programs. SIAM Journal on Optimization 9(1):33–52

Gabrel V, Lacroix M, Murat C, Remli N (2014) Robust location transportation problems under uncertain demands. Discrete Applied Mathematics 164(1):100 − 111

Goh J, Sim M (2010) Distributionally robust optimization and its tractable approximations. Oper Res 58(4-Part-1):902–917

Iancu D, Sharma M, Sviridenko M (2013) Supermodularity and affine policies in dynamic robust optimization. Operations Research 61(4):941–956

IBM-ILOG (2015) IBM-ILOG Cplex

Kouvelis P, Yu G (1997) Robust Discrete Optimization and its Application. Kluwer Academic Publishers, London

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

Mattia S (2014) Private communication

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

Billionnet A, Costa M-C, Poirion P-L (2014) 2-stage robust MILP with continuous recourse variables. Discrete Applied Mathematics 170:21–32

Poss M (2013) A comparison of routing sets for robust network design. Optimization Letters In press

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

Postek K, Den Hertog D (2014) Multi-stage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. CentER Discussion Paper Series

Ryoo HS, Sahinidis NV (2003) Global optimization of multiplicative programs. J of Global Optimization 26(4):387–418

Zeng B, Zhao L (2013) Solving two-stage robust optimization problems by a constraint-and-column generation method. Operations Research Letters 41(5):457461