Robust optimization for models with uncertain SOC and SDP constraints

Jianzhe Zhen*
Department of Econometrics and Operations Research, Tilburg University, Tilburg, The Netherlands

Frans J.C.T. de Ruiter
CQM, Eindhoven, The Netherlands
Department of Econometrics and Operations Research, Tilburg University, Tilburg, The Netherlands

Dick den Hertog
Department of Econometrics and Operations Research, Tilburg University, Tilburg, The Netherlands

In this paper we consider uncertain second-order cone (SOC) and semidefinite programming (SDP) constraints with polyhedral uncertainty. We propose to reformulate an uncertain SOC or SDP constraint as a set of adjustable robust linear optimization constraints with an ellipsoidal or semidefinite representable uncertainty set, respectively. The resulting adjustable problem can then (approximately) be solved by using adjustable robust linear optimization techniques. For example, we show that if linear decision rules are used, then the final robust counterpart consists of SOC or SDP constraints, respectively, which have the same computational complexity as the nominal version with the original constraints. We also propose an efficient method to obtain good lower bounds, and extend our approach to other classes of robust optimization problems. Finally, we apply our approach to a robust regression problem and a robust sensor network problem. We use linear decision rules to solve the resulting adjustable robust linear optimization problems and the solutions found are optimal or near optimal.

Key words: Robust optimization, second-order cone, semidefinite programming, adjustable robust optimization, linear decision rules.

1. Introduction

Practical optimization problems often contain uncertain parameters. Uncertainty arises, because of, e.g., estimation or prediction errors. One way of dealing with such uncertainty is robust optimization. The papers El Ghaoui and Lebret (1997), El Ghaoui et al. (1998) and Ben-Tal and Nemirovski (1998) are considered as the birth of this field.

In robust optimization the uncertainty is not modeled by probability distributions as in stochastic programming, but as uncertainty sets. Such an uncertainty set contains all scenarios for the uncertain parameters for which the decision maker would like to safeguard herself. It is enforced that the constraints should hold for all scenarios in this uncertainty set.

* Corresponding author: j.zhen@tilburguniversity.edu
The paper Ben-Tal et al. (2004) extends the robust optimization methodology to problems that also contain wait-and-see or adjustable variables. Such variables often occur in multi-stage problems. For adjustable variables one can wait until (a part of) the uncertain parameters have revealed their values. Many efficient methods have been proposed in literature to (approximately) solve such *adjustable robust optimization* problems.

The advantages of robust optimization are among others the computational tractability and the fact that there is no need to specify a probability distribution. For many classes of robust optimization problems it has been shown that one can reformulate the problem into tractable ones. Many of these cases are treated in the book Ben-Tal et al. (2009). A detailed and unified approach to derive computationally tractable reformulations is given in Ben-Tal et al. (2015). In that paper it is shown that, loosely speaking, if the constraints are concave in the uncertain parameters, and of course convex in the optimization variables, then the robust counterpart can be formulated by convex constraints.

For several problems that contain constraints that are not concave in the uncertain parameters, computationally tractable approximations have been proposed. For robust second-order cone (SOC) and robust semidefinite programming (SDP) constraints that are convex in the uncertain parameters, exact and approximate tractable reformulations for specific (simple) ellipsoidal or norm-bounded uncertainty sets have been proposed by El Ghaoui and Lebret (1997), El Ghaoui et al. (1998), and Ben-Tal et al. (2002), which are summarized in the book Ben-Tal et al. (2009). In all these approaches, both for uncertain SOC and SDP constraints, the final *robust counterpart* contains an SDP constraint. We are not aware of papers that deal with uncertain SOC or SDP constraints with general polyhedral uncertainty.

In this paper we consider uncertain SOC and SDP constraints with polyhedral uncertainty. We propose to reformulate an uncertain SOC or SDP constraint as a set of *adjustable robust linear optimization* constraints with an ellipsoidal or semidefinite representable uncertainty set, respectively. The resulting adjustable problem can then (approximately) be solved by using adjustable robust linear optimization techniques described in the literature. For example, we show that if linear decision rules are used, then the final robust counterpart consists of SOC or SDP constraints, respectively, which have the same computational complexity as the nominal version of the original constraints. We also propose an efficient method to obtain good lower bounds, and extend our approach to other classes of robust optimization problems. Finally, we apply our approach to a robust regression problem and a robust sensor network problem. We use linear decision rules to solve the resulting adjustable robust linear optimization problems and the solutions found are optimal or near optimal.
This paper is organized as follows. In §2 we treat uncertain SOC constraints, and in §3 uncertain SDP constraints with polyhedral uncertainty. §4 describes how to obtain sharp lower bounds in an efficient way. In §5 extensions of our approach to other classes of robust optimization problems are given. §6 and §7 contain the numerical results for the robust regression and robust sensor network problem, respectively. §8 contains recommendations for future research.

2. Uncertain second-order cone constraints

Consider the following uncertain second-order cone constraint:

$$\forall \zeta \in U : \quad a(x)^\top \zeta + \|A(x)\zeta + b(x)\|_2 \leq c(x).$$ (1)

Here \(\| \cdot \|_2\) denotes the 2-norm, \(x \in \mathbb{R}^{nx}\) is the decision (or optimization) variable in a given domain \(X \subseteq \mathbb{R}^n\), e.g., \(X = \mathbb{R}^{nx}\) or \(X = \mathbb{Z}_+^{nx}\), and \(\zeta \in \mathbb{R}^n\) is the uncertain parameter that resides in the uncertainty set \(U \subseteq \mathbb{R}^n\). The vectors \(a(x) = (a_1(x), \ldots, a_n(x))^\top\) and \(b(x) = (b_1(x), \ldots, b_m(x))^\top\), \(A(x) \in \mathbb{R}^{m \times n}\), and \(c(x) \in \mathbb{R}\) have entries that are affine functions of \(x\). That is, \(a_j(\cdot) : \mathbb{R}^{nx} \to \mathbb{R}\), \(b_i(\cdot) : \mathbb{R}^{nx} \to \mathbb{R}\), \(A_{ij}(\cdot) : \mathbb{R}^{nx} \to \mathbb{R}\), and \(c(\cdot) : \mathbb{R}^{nx} \to \mathbb{R}\) are affine functions for all \(i = 1, \ldots, m\), and \(j = 1, \ldots, n\). In the following examples, we show that a robust regression model and an uncertain quadratic constraint can be expressed by (1). For more details on the modeling power of (1) and a numerical experiment on robust regression, we postpone the discussion to §5 and §6, respectively.

**Example 1. Robust Regression.** In regression models we try to find a vector of coefficients \(x \in \mathbb{R}^{nx}\) such that the norm (or squared norm) of \(Ax - b\) is minimized. The standard least-squares solution is the optimal solution to the model:

$$\min_{x \in \mathbb{R}^{nx}} \|Ax - b\|_2.$$

Here \(A \in \mathbb{R}^{m \times nx}\) and \(b \in \mathbb{R}^m\) are data. Each row of the matrix \(A\) is a different observation and the columns refer to the features. The \(i\)-th entry of \(b\) corresponds to the response, or target value, of the \(i\)-th observation. Often, some of the data entries in \(A\) and/or \(b\) are obtained via measurements, and therefore subject to uncertainty. Suppose there are uncertainties in the entries of the matrix \(A\). For robust regression, we can replace the matrix \(A\) in the least-squares model by the term \(A + \zeta\), where \(\zeta \in \mathbb{R}^{m \times nx}\) is a matrix with uncertain parameters, and minimize \(\tau\) subject to:

$$\forall \zeta \in U : \|(A + \zeta)x - b\|_2 \leq \tau,$$ (2)

where \(U\) is a polyhedral set and \(\tau \in \mathbb{R}\) is an optimization variable. □
Example 2. **Uncertain quadratic constraints.** Consider the following constraint:

$$\forall \zeta \in U : \quad \zeta^\top H(x)^\top H(x)\zeta + f(x)^\top \zeta \leq g(x),$$

where the entries of $H(\cdot) : \mathbb{R}^{nx} \to \mathbb{R}^{nx \times nx}$, $f(\cdot) : \mathbb{R}^{nx} \to \mathbb{R}^n$ and $g(\cdot) : \mathbb{R}^{nx} \to \mathbb{R}$ are affine functions. This is equivalent to an uncertain second-order cone constraint in the form of (1):

$$\forall \zeta \in U : \quad \left\| \frac{(1 + f(x)^\top \zeta - g(x))}{H(x)\zeta} \right\|_2 \leq \frac{(1 - f(x)^\top \zeta + g(x))}{2}.\quad \Box$$

Robust regression models and uncertain quadratic constraints with specific norm bounded uncertainty sets were studied by El Ghaoui and Lebret (1997). Throughout this paper, we focus on different uncertainty sets, namely, polyhedral uncertainty sets of the form:

$$U = \{ \zeta \geq 0 : D\zeta \leq d \} \neq \emptyset,$$

with $D \in \mathbb{R}^{r \times nx}$ and $d \in \mathbb{R}^r$. The condition $\zeta \geq 0$ can also be omitted, but is included here for ease of exposition. Constraint (1) with polyhedral uncertainty set $U$ is equivalent to:

$$\max_{\zeta \in U} \{ a(x)^\top \zeta + \| A(x)\zeta + b(x) \|_2 \} \leq c(x).$$

The left-hand side of inequality (4) is a convex maximization problem, which is generally intractable. One way of solving this type of problems is by considering all the extreme points of the polyhedral set, which is impractical as there are often way too many extreme points. We further discuss this method in §4.2. Alternatively, in the following theorem we propose a novel technique to reformulate the second-order cone constraint (1) into a set of two-stage robust linear constraints.

**Theorem 1.** Let $U$ be the polyhedral uncertainty set given in (3). Then $x \in \mathbb{R}^{nx}$ satisfies constraint (1) if and only if it satisfies the following set of two-stage robust linear constraints:

$$\forall w \in W \exists \lambda \geq 0 : \begin{cases} d^\top \lambda + b(x)^\top w \leq c(x) \\ D^\top \lambda \geq a(x) + A(x)^\top w, \end{cases}$$

where $W = \{ w \in \mathbb{R}^m : \| w \|_2 \leq 1 \}$ and $\lambda \in \mathbb{R}^r$.

**Proof.** For constraint (1) we can derive the following equivalences:

$$\forall \zeta \in U : \quad a(x)^\top \zeta + \max_{w: \| w \|_2 \leq 1} w^\top (A(x)\zeta + b(x)) \leq c(x)$$

$$\iff \forall w \in W \forall \zeta \in U : \quad a(x)^\top \zeta + w^\top (A(x)\zeta + b(x)) \leq c(x),$$

(6)
with \( W = \{ w \in \mathbb{R}^m : \| w \|_2 \leq 1 \} \). By dualizing over \( \zeta \), using strong duality for linear programming, we can further deduce that (6) is equivalent to:

\[
\forall w \in W: \max_{\zeta \in U} \left\{ a(x) ^\top \zeta + w ^\top (A(x) \zeta + b(x)) \right\} \leq c(x)
\]

\[
\Leftrightarrow \forall w \in W: w ^\top b(x) + \min_{\lambda \geq 0} \left\{ d ^\top \lambda \mid D ^\top \lambda \geq a(x) + A(x) ^\top w \right\} \leq c(x)
\]

\[
\Leftrightarrow \forall w \in W \exists \lambda \geq 0:\begin{cases}
d ^\top \lambda + b(x) ^\top w \leq c(x) \\
D ^\top \lambda \geq a(x) + A(x) ^\top w.
\end{cases}
\]

As the result of the reformulation, the newly introduced variables \( w \) and \( \lambda \) appear linearly in constraints (5). The set of constraints (5) can be seen as the constraints of a two-stage robust linear optimization model where \( w \), that resides in the second order cone \( W \), can be considered as the uncertain parameter. The first-stage or here-and-now decision \( x \) is decided before the realization of the uncertainty parameter \( w \), and the second-stage or wait-and-see decision \( \lambda \) is determined after the value of \( w \) is revealed, where \( w \) resides in an ellipsoidal uncertainty set \( W \). The coefficients of \( \lambda \) (i.e., \( d \) and \( D \)) are constant, which corresponds to the stochastic programming format known as fixed recourse. Two-stage robust linear optimization models are generally intractable because the wait-and-see decision is in fact a decision rule, or infinite dimensional variable, instead of a finite vector of decision variables (see Ben-Tal et al. (2004)).

Theoretically, for several two-stage robust linear models, the structure of the optimal decision rules has been characterized. For instance, by using Zhen and den Hertog (2017a), it can be shown that there exists a polynomial of (at most) degree \( m \) and linear in each of the \( w_i \), \( i = 1, ..., m \), that is optimal for \( \lambda \) in (5). One can use Zhen et al. (2017) to establish the optimality of the following decision rules for \( \lambda \) in (5). There exists a piecewise affine function that is optimal for \( \lambda \) in (5). More specifically, if \( U \) is simplicial, there exists a linear decision rule that is optimal for \( \lambda \) in (5); if \( U \) is a box, there exists a two-piecewise affine function that is optimal for \( \lambda \) in (5), and the techniques proposed in Gorissen and den Hertog (2013) and Ardestani-Jaafari and Delage (2016) can then be applied to solve Problem (5) approximately. Unfortunately, even when the structure of optimal decision rules is known, it is often hard to find optimal solutions due to the computational intractability of such rules.

Numerically, the main advantage of (5) is that it can be (approximately) solved by any method applicable to two-stage robust linear models such as linear decision rules (see Ben-Tal et al. (2004)), Fourier-Motzkin elimination (see Zhen et al. (2017)), finite adaptability approaches (see Postek and den Hertog (2016), Bertsimas and Dunning (2016), Georghiou et al. (2017)) etc. These solution methods will be discussed in §4.1. Two numerical experiments with uncertain second-order cone
constraints are conducted, i.e., a robust regression model (§6) and a robust sensor network model (§7), to evaluate the performance of the proposed methods.

We again note that the condition $\zeta \geq 0$ in the uncertainty set $\mathcal{U}$ can also be omitted. In that case we will have equality constraints $D^\top \lambda = a(x) + A(x)^\top w$. The equalities can be used to eliminate some of $\lambda$ via Gaussian elimination. It is well-known that eliminating the wait-and-see variables in the equalities of a two-stage fixed-recourse robust model is equivalent to imposing linear decision rules (Zhen and den Hertog 2017a, Lemma 2).

3. Uncertain semidefinite programming constraints

Consider the following uncertain semidefinite programming constraint:

$$\forall \zeta \in \mathcal{U}: \ A(x, \zeta) \succeq 0,$$

(7)

where

$$A(x, \zeta) = A^{(0)}(x) + \sum_{i=1}^{n} A^{(i)}(x) \zeta_i,$$

$x \in \mathcal{X}$ and $A^{(i)}(x) \in \mathbb{R}^{m \times m}$ for all $i = 0, \ldots, m$. More precisely, $A^{(i)}(\cdot): \mathbb{R}^n \to \mathbb{R}^{m \times m}$ is a (componentwise) affine function for each $i = 0, \ldots, n$. The following theorem shows that an uncertain semidefinite programming constraint with polyhedral uncertainty can also be reformulated into a set of two-stage robust linear constraints with a semidefinite representable uncertainty set.

**Theorem 2.** Let $\mathcal{U}$ be a polyhedral uncertainty set as in (3). Then $x \in \mathbb{R}^n$ satisfies constraint (7) if and only if it satisfies

$$\forall W \succeq 0 \ \exists \lambda \succeq 0:\ \begin{cases} 
\text{Tr}(A^{(0)}(x)W) - d^\top \lambda \geq 0 \\
D^\top_i \lambda \geq -\text{Tr}(A^{(i)}(x)W) 
\end{cases} \quad i = 1, \ldots, n,$$

(8)

where $\text{Tr}(\cdot)$ denotes the trace function, $\lambda \in \mathbb{R}^r$ and $D_i$ is the $i$-th column of $D$ for $i = 1, \ldots, n$.

**Proof.** From Lemma 3 (see Appendix A) we know that a matrix $A(x, \zeta)$ is positive semidefinite if and only if the trace of the product with any positive semidefinite matrix is positive. For constraint (7) we then can derive the following equivalences:

$$\forall \zeta \in \mathcal{U}: \ A(x, \zeta) \succeq 0$$

$\iff$

$$\forall W \succeq 0 \forall \zeta \in \mathcal{U}: \ \text{Tr}(A(x, \zeta)W) \geq 0$$

$\iff$

$$\forall W \succeq 0 \forall \zeta \in \mathcal{U}: \ \text{Tr}(A^{(0)}(x)W) + \sum_{i=1}^{n} \text{Tr}(A^{(i)}(x)W) \zeta_i \geq 0$$

$\iff$

$$\forall W \succeq 0: \ \text{Tr}(A^{(0)}(x)W) + \min_{\zeta \in \mathcal{U}} \left\{ \sum_{i=1}^{n} \text{Tr}(A^{(i)}(x)W) \zeta_i \right\} \geq 0.$$
By dualizing over $\zeta$, using strong duality for linear programming, we obtain:

$$\forall W \succeq 0 : \quad \text{Tr} \left( A^{(0)}(x)W \right) + \max_{\lambda \geq 0} \left\{ -d^T \lambda \mid D_i^T \lambda \geq -\text{Tr} \left( A^{(i)}(x)W \right), \; i = 1, \ldots, n \right\} \geq 0$$

$$\Leftrightarrow \forall W \succeq 0 \exists \lambda \geq 0 : \quad \begin{cases} 
\text{Tr} \left( A^{(0)}(x)W \right) - d^T \lambda \geq 0 \\
D_i^T \lambda \geq -\text{Tr} \left( A^{(i)}(x)W \right) 
\end{cases} \quad i = 1, \ldots, n.$$

Notice that since the system (8) is homogeneous in $\lambda$ and $W$, one can in fact replace the unbounded uncertainty set ‘$\forall W \succeq 0$’ by the bounded set ‘$\forall W : I \succeq W \succeq 0$’ without affecting the feasible region of $x$, where $I \in \mathbb{R}^{m \times m}$ denotes the identity matrix. Any solution method applicable for two-stage robust optimization models can be used to solve problems with constraints (8). These solution methods will be discussed in §4.

4. Tractable conservative and progressive approximations

4.1. Conservative approximation

One popular remedy for the intractability of two-stage robust linear optimization models is to restrict the wait-and-see decisions in (5) and (8) to be simple functions of the uncertain parameters, e.g., linear decision rules (also known as, affine policies, see Ben-Tal et al. (2004)). In the following lemma we present the tractable conservative approximation of constraints in (5) via linear decision rules, which is also a conservative approximation of (1).

**Lemma 1.** The vector $x \in X$ satisfies constraint (5) if there exist $v \in \mathbb{R}^r$ and $V \in \mathbb{R}^{r \times m}$ such that $x$ also satisfies:

$$\begin{cases} 
d^T v + \left\| V^T d + b(x) \right\|_2 \leq c(x) \\
a_i(x) + \left\| A_i(x) - V^T D_i \right\|_2 \leq D_i^T v \\
\left\| (V^T)_j \right\|_2 \leq v_j, 
\end{cases} \quad i = 1, \ldots, n$$

where $a_i$ and $v_i$ denote the $i$-th elements of $a$ and $v$, respectively, $A_i, D_i$ and $(V^T)_j$ denote the $i$-th column of $A, D$ and $V^T$, respectively.

**Proof.** By restricting $\lambda$ to the linear decision rule in (5):

$$\lambda = v + Vw,$$

we obtain the following conservative approximation of (5):

$$\forall w \in \mathcal{W} : \quad \begin{cases} 
d^T (v + Vw) + b(x)^T w \leq c(x) \\
D_i^T (v + Vw) \geq a_i(x) + A_i^T w \\
v + Vw \geq 0, 
\end{cases} \quad i = 1, \ldots, n$$

where the entries of vector $v \in \mathbb{R}^r$ and coefficient matrix $V \in \mathbb{R}^{r \times m}$ are optimization variables. By using well-established reformulation techniques (Ben-Tal et al. (2015)) to get rid of the ‘$\forall w \in \mathcal{W}$’,...
it can be verified that the tractable conservative approximation of constraints (10) are exactly the constraints in (9).

Since we restrict the decision rule $\lambda$ to be affine, the set of constraints (9) is indeed a conservative approximation (1). Note that the set of second-order cone constraints (9) has the same computational complexity as the nominal version (that is, with no uncertainty) of (1).

A simple but powerful enhancement of linear decision rules has been proposed recently by de Ruiter and Ben-Tal (2017), where the authors use a lifted variant of $W$:

$$\hat{W} = \left\{ (w, z) \in \mathbb{R}^m \times \mathbb{R}^m : w_i^2 \leq z_i, i = 1, \ldots, m, \sum_{i=1}^m z_i \leq 1 \right\},$$

and show that the resulting linear decision rule is equivalent to the following nonlinear decision rule:

$$\lambda^t = v + Vw + Uz,$$

where $z_i = w_i^2$ for $i = 1, \ldots, m$ and $U \in \mathbb{R}^{r \times m}$. Notice that the projection of $\hat{W}$ onto its $w$-space is $W$.

The tractable robust counterpart of (5) with $\hat{W}$ can be derived by first imposing decision rule (11) on $\lambda$, and then apply the standard robust optimization techniques. The resulting robust counterpart is a set of second-order cone constraints (see Appendix B), that is, in the same complexity class as (9), and it is a possibly tighter conservative approximation of (1) than (9).

Similarly, the following lemma gives the tractable conservative approximation of the robust semidefinite programming constraints in (8) via linear decision rules.

**Lemma 2.** The vector $x \in X$ satisfies constraint (8) if there exist $v \in \mathbb{R}^r$ and $V^j \in \mathbb{R}^{m \times m}$, $j = 1, \ldots, r$, such that $x$ also satisfies:

$$\begin{align*}
\begin{cases}
   d^Tv \leq 0 \\
   A^{(0)}(x) - \sum_{j=1}^r d_j V^{(j)} \succeq 0 \\
   D_i^Tv \geq 0 & i = 1, \ldots, n, \\
   \sum_{j=1}^r D_{ij}V^{(j)} + A^{(i)}(x) \succeq 0 & i = 1, \ldots, n \\
   v \geq 0 \\
   V^{(j)} \succeq 0 & j = 1, \ldots, r,
\end{cases}
\end{align*}$$

(12)

**Proof.** Similarly as for the proof of (9), by restricting $\lambda$ to a linear decision rule, we obtain the following conservative approximation of (8):

$$\forall W \succeq 0 : \begin{cases}
   \text{Tr} \left( A^{(0)}(x)W \right) - d^Tv - \text{Tr} \left( \sum_{j=1}^r d_j V^{(j)}B \right) \geq 0 \\
   D_i^Tv + \text{Tr} \left( \sum_{j=1}^r D_{ij}V^{(j)}B \right) \geq -\text{Tr} \left( A^{(i)}(x)W \right) & i = 1, \ldots, n \\
   v_j + \text{Tr} \left( V^{(j)}W \right) \geq 0 & j = 1, \ldots, r,
\end{cases}$$

(13)

where the vector $v \in \mathbb{R}^r$ and coefficient matrix $V^{(j)} \in \mathbb{R}^{m \times m}$, $j = 1, \ldots, r$, are optimization variables.

The tractable robust counterpart (9) can be easily obtained via well-established reformulation techniques (Ben-Tal et al. (2015)) to get rid of the ‘$\forall w \in W$’.

□
The set of semidefinite constraints (12) has the same computational complexity as the nominal version of (7).

As discussed in §2, the inner approximations via linear decision rules are tight if the uncertainty set $U$ in (1) and (7) is simplicial. In §6 and §7, we show that linear decision rules give close-to-optimal solutions for problems with uncertain second-order cone constraints in the numerical experiments.

Another popular approach for solving two-stage robust optimization problems is finite adaptability in which the uncertainty set $W$ is split into a number of smaller subsets, each with its own set of recourse decisions. The number of these subsets can be either fixed a priori or decided by the optimization model (Vayanos et al. (2011), Bertsimas and Caramanis (2010), Hansusanto et al. (2014), Postek and den Hertog (2016), Bertsimas and Dunning (2016), Georghiou et al. (2017)). Recently it has been shown by Zhen et al. (2017) that one can use Fourier-Motzkin elimination to eliminate the wait-and-see decisions in (5), and establish guaranteed optimality in finite number of steps. In this way, problems of small size can be solved to optimality. For larger problems one could eliminate a subset of the wait-and-see decisions and then apply the existing methods, e.g., linear decision rules, finite adaptability approaches, to solve the resulting problems.

4.2. Progressive approximation

Since the methods discussed in §4.1 are conservative, the solutions are in general suboptimal. It is therefore important to find effective outer approximations to assess the quality of conservative approximations of §4.1. In this subsection, we focus on progressive outer approximation methods for uncertain second-order cone constraint (1). We would like to point out that the discussed methods can be applied in an analogous way to uncertain semidefinite constraint (7).

One simple way of obtaining an outer approximation of (1) is to only consider a finite subset of scenarios $\{\zeta^{(1)}, \ldots, \zeta^{(K)}\}$ from the uncertainty set $U$. The outer approximation is therefore the “sampled version” of (1):

$$a(x)^\top \zeta^{(k)} + \|A(x)\zeta^{(k)} + b(x)\|_2 \leq c(x) \quad k = 1, \ldots, K. \quad \text{(14)}$$

These are standard second-order cone constraints. Clearly the set of constraints (14) is an outer approximation of (1), since a feasible $\hat{x}$ of (14) is only feasible for a finite subset of the uncertainty set. There could be realizations in $U$ for which $\hat{x}$ is infeasible. For a polyhedral $U = \{\zeta \geq 0 : D\zeta \leq d\}$, if the set contains all the extreme points $\zeta^{(1)}, \ldots, \zeta^{(K)}$ of $U$, any feasible solution $\hat{x}$ of (14) is also feasible for (1). Of course, the set of extreme points of a polyhedral uncertainty set $U$ is in practice way too large. As we see in our numerical examples, this is only doable when the uncertainty set has
only a few extreme points. We apply the same reasoning to (5) to obtain an outer approximation for the reformulation of the second-order cone constraint:

\[
\begin{align*}
&d^\top \lambda^{(k)} + b(x)^\top w^{(k)} \leq c(x) \quad k = 1, \ldots, K \\
&D^\top \lambda^{(k)} \geq a(x)^\top A^\top w^{(k)} \quad k = 1, \ldots, K \\
&\lambda^{(k)} \geq 0 \quad k = 1, \ldots, K,
\end{align*}
\]

(15)
is also a valid outer approximation of (1). Here \( \{w^{(1)}, \ldots, w^{(K)}\} \subseteq W = \{w \in \mathbb{R}^m : \|w\|_2 \leq 1\} \) and \( \lambda^{(k)} \in \mathbb{R}^r \) is a here-and-now decision for \( k = 1, \ldots, K \). In this case there are infinitely many extreme points of the second-order cone \( W \). A complete enumeration of all the extreme point would be impossible. Given two finite scenario sets \( \{\zeta^{(1)}, \ldots, \zeta^{(K)}\} \) and \( \{w^{(1)}, \ldots, w^{(K)}\} \), one can of course combine the constraints in (14) and (15) to obtain a possibly tighter outer approximation of (1).

Hadjijyiannis et al. (2011) propose a way to obtain a small and effective finite set of scenarios for two-stage fixed-recourse robust linear constraints. For any feasible \( (\hat{x}, \hat{\nu}, \hat{V}) \) of (10), their method takes scenarios that are worst case for the constraints in (10), hoping that the same set of scenarios is also worst case for the optimal (nonlinear) decision rule. For instance, such a scenario of (10) can be:

\[
\hat{w} = \arg \max_{w \in W} \left\{ d^\top \left( \hat{\nu} + \hat{V} w \right) + b(\hat{x})^\top w \right\} = \frac{\hat{V}^\top d + b(\hat{x})}{\| \hat{V}^\top d + b(\hat{x}) \|_2},
\]

(16)

where \( (\hat{x}, \hat{\nu}, \hat{V}) \) satisfies (10). For each constraint one can obtain one such scenario. The obtained scenarios \( \{\hat{w}^{(1)}, \ldots, \hat{w}^{(r)}\} \) can then be used in (15) to obtain an outer approximation of (1). For more details on the method we refer to the original paper by Hadjijyiannis et al. (2011). One direct extension of the method of Hadjijyiannis et al. (2011) is to use the obtained scenarios \( \{\hat{w}^{(1)}, \ldots, \hat{w}^{(r)}\} \) to recover scenarios \( \{\tilde{\zeta}^{(1)}, \ldots, \tilde{\zeta}^{(r)}\} \subseteq \mathcal{U} \), where

\[
\tilde{\zeta}^{(k)} = \arg \max_{\zeta \in \mathcal{U}} \left\{ a(\hat{x})^\top \zeta + (\hat{w}^{(k)})^\top (A(\hat{x})\zeta + b(\hat{x})) \right\} \quad k = 1, \ldots, r,
\]

(17)

which can then be used in (14) to obtain an outer approximation of (1). One can again combine constraints (14) with \( \{\tilde{\zeta}^{(1)}, \ldots, \tilde{\zeta}^{(r)}\} \) and constraints (15) with \( \{\hat{w}^{(1)}, \ldots, \hat{w}^{(r)}\} \) to obtain a possibly tighter outer approximation of (1). However, for a special case of (1) where \( a(x) = a \) and \( A(x) = A \), the constraints (15) with \( \{\hat{w}^{(1)}, \ldots, \hat{w}^{(r)}\} \) are redundant with respect to constraints (14) with \( \{\tilde{\zeta}^{(1)}, \ldots, \tilde{\zeta}^{(r)}\} \).

**Theorem 3.** Let \( a(x) = a \), \( A(x) = A \), \( \{\hat{w}^{(1)}, \ldots, \hat{w}^{(r)}\} \subseteq W \) be a finite set of scenarios and \( \{\tilde{\zeta}^{(1)}, \ldots, \tilde{\zeta}^{(r)}\} \subseteq \mathcal{U} \) be the corresponding set of scenarios from (17). Then \( x \in \mathbb{R}^n \) satisfies the constraints (14) with \( \{\tilde{\zeta}^{(1)}, \ldots, \tilde{\zeta}^{(r)}\} \) also satisfies the constraints (15) with \( \{\hat{w}^{(1)}, \ldots, \hat{w}^{(r)}\} \).
Proof. Let $\bar{x}$ be a vector that satisfies:

$$a^\top \bar{\zeta}^{(k)} + \|A\bar{\zeta}^{(k)} + b(x)\|_2 \leq c(x) \quad k = 1, \ldots, r$$

$$\iff a^\top \bar{\zeta}^{(k)} + \max_{w: \|w\|_2 \leq 1} w^\top (A\bar{\zeta}^{(k)} + b(x)) \leq c(x) \quad k = 1, \ldots, r.$$

Since $\{\bar{w}^{(1)}, \ldots, \bar{w}^{(r)}\} \subseteq \mathcal{W}$, then by definition $\bar{x}$ also satisfies:

$$a^\top \bar{\zeta}^{(k)} + (\bar{w}^{(k)})^\top (A\bar{\zeta}^{(k)} + b(x)) \leq c(x) \quad k = 1, \ldots, r$$

$$\iff \max_{\zeta \in U} \left\{a^\top \zeta + (\bar{w}^{(k)})^\top (A\zeta + b(x))\right\} \leq c(x) \quad k = 1, \ldots, r,$$

(18)

where we have used the definition of $\bar{\zeta}^{(k)}$ from (17). Note that the equivalence here is due to $a(x) = a$ and $A(x) = A$. By dualizing over $\zeta$ in (18), using strong duality for linear programming, then $\bar{x}$ also satisfies:

$$\begin{cases} a^\top \lambda^{(k)} + b(x)^\top \bar{w}^{(k)} \leq c(x) & k = 1, \ldots, r \\ D^\top \lambda^{(k)} \geq a + A^\top \bar{w}^{(k)} & k = 1, \ldots, r \\ \lambda^{(k)} \geq 0 & k = 1, \ldots, r. \end{cases}$$

Theorem 3 shows that the scenarios from (17) are very efficient for the special case of (1), and therefore used in the numerical experiments.

5. Extensions

Uncertain SOC and SDP constraints with wait-and-see decisions

Now suppose a set of uncertain second-order cone constraints in the form (1) contains a wait-and-see decision $y$:

$$\forall \zeta \in U: \ a(x)^\top \zeta + h(y) + \|A(x)\zeta + By + b(x)\|_2 \leq c(x), \quad (19)$$

where $h(\cdot): \mathbb{R}^{n_y} \to \mathbb{R}$ is an affine function. One simple yet crucial observation is that, by imposing linear decision rule $y = u + Y\zeta$, where the vector $u \in \mathbb{R}^{n_y}$ and coefficient matrix $Y \in \mathbb{R}^{n_y \times n}$ are here-and-now decision variables, constraint (19) becomes an instance of (1):

$$\forall \zeta \in U: \ a(x)^\top \zeta + h(u + Y\zeta) + \|A(x)\zeta + Bu + BY\zeta + b(x)\|_2 \leq c(x).$$

The techniques discussed in the previous sections can be readily applied. Analogously, one can also use this simple technique to reformulate a set of uncertain SDP constraints with wait-and-see decisions into an instance of (8).
Bilinear uncertainty

Consider the following robust constraint with bilinear uncertainty:

\[ \forall w \in W \quad \forall \xi \in U : \quad a(x)^\top \xi + w^\top (A(x)\xi + b(x)) \leq c(x), \quad (20) \]

where \( W \) is a general convex set. In the following proposition, we reformulate constraint (20) into a set of two-stage robust linear constraints.

**Proposition 1** Let \( U \) be the polyhedral uncertainty set given in (3). Then \( x \in \mathbb{R}^n \) satisfies constraint (20) if and only if it satisfies

\[
\forall w \in W \exists \lambda \geq 0 : \begin{cases}
d^\top \lambda + b(x)^\top w \leq c(x) \\
D^\top \lambda \geq a(x) + A(x)^\top w.
\end{cases} \quad (21)
\]

**Proof.** The proof is similar to the proof of Theorem 1, hence, omitted. \( \square \)

Problems with constraints in the form of (21) can then be solved by using the methods discussed in §4.

**Uncertain nonlinear constraints with SOC representation**

In §2 and §3 we show how to reformulate uncertain second-order cone constraints and uncertain semi-definite constraints into a set of two-stage robust linear constraints. One can also consider constraint (1) with \( a_j(\cdot) : \mathbb{R}^n \to \mathbb{R}, \quad i = 1, \ldots, m, \) and \( -c(\cdot) \) are convex functions, and constraint (7) with the entries of \( A^{(i)}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{m \times m} \) are concave functions for each \( i = 0, \ldots, n \). The solution methods discussed in §4 can also be directly applied.

In the following example we extend the result in §2 to uncertain constraints with \( l_p \)-norms.

**Example 3. Uncertain constraint with \( l_p \)-norms.** Let us consider an uncertain constraint in the form:

\[ \forall \xi \in U : \quad a(x)^\top \xi + \|A(x)\xi + b(x)\|_p \leq c(x), \quad (22) \]

where \( p \in [1, \infty] \). By definition of the dual norm (see, for example, Lax (2007)), we know:

\[
\forall \xi \in U : \quad a(x)^\top \xi + \|A(x)\xi + b(x)\|_p \leq c(x) 
\]

\[ \iff \forall \xi \in U : \quad a(x)^\top \xi + \max_{w : \|w\|_q \leq 1} w^\top (A(x)\xi + b(x)) \leq c(x) \]

\[ \iff \forall w \in \mathcal{W} \forall \xi \in U : \quad a(x)^\top \xi + w^\top (A(x)\xi + b(x)) \leq c(x), \]

where the set \( \mathcal{W} = \{ w : \|w\|_q \leq 1 \} \) is convex, and \( q = \frac{p}{p-1} \) for \( p \in [1, \infty] \). Applying Proposition 1 yields a set of two-stage robust linear constraints in the form (21) with a \( l_q \)-norm uncertainty set.
In the following examples, we use the simple algebraic principle:

$$u^\top u \leq st, \quad s \geq 0, \quad t \geq 0 \quad \Leftrightarrow \quad \left\| \frac{2u}{s-t} \right\|_2 \leq s + t, \quad (23)$$

and show that a large class of uncertain convex constraints can be cast as uncertain second-order cone constraints with wait-and-see decisions. The reformulation techniques we use in the following examples are from Lobo et al. (1998).

**Example 4. Uncertain quadratic/linear fraction constraints.** Let us consider uncertain constraints in the form:

$$\forall \zeta \in \mathcal{U} : \begin{cases} \sum_{i=1}^K \left\| A^{(i)}(x) \zeta + b^{(i)}(x) \right\|_2^2 \leq \tau, \\ (a^{(i)}(x))^\top \zeta + c_i(x) > 0 \quad i = 1, \ldots, K, \end{cases} \quad (24)$$

where $a^{(i)}(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^n$, $c_i(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$, $A^{(i)}(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{m \times n_x}$, $b^{(i)}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n_y}$ are (componentwise) affine functions for $i = 1, \ldots, K$, and $\tau \in \mathbb{R}$ is a constant or an optimization variable. By introducing a wait-and-see variable $y$, we have an equivalent reformulation of (24) (one simple way to verify the equivalence is by eliminating the auxiliary variable $y$ via Fourier-Motzkin elimination, see Zhen et al. (2017)):

$$\forall \zeta \in \mathcal{U} \exists y \geq 0 : \begin{cases} \sum_{i=1}^K y_i \leq \tau, \\ (A^{(i)}(x) \zeta + b^{(i)}(x))^\top (A^{(i)}(x) \zeta + b^{(i)}(x)) \leq y_i ((a^{(i)}(x))^\top \zeta + c_i(x)) \quad i = 1, \ldots, K \\ (a^{(i)}(x))^\top \zeta + c_i(x) > 0 \quad i = 1, \ldots, K, \end{cases} \quad (25)$$

which is equivalent to (due to (23)):

$$\forall \zeta \in \mathcal{U} \exists y \geq 0 : \begin{cases} \sum_{i=1}^K y_i \leq \tau, \\ \left\| 2A^{(i)}(x) \zeta + 2b^{(i)}(x) \right\|_2 \leq (a^{(i)}(x))^\top \zeta + c_i(x) + y_i \quad i = 1, \ldots, K \\ (a^{(i)}(x))^\top \zeta + c_i(x) > 0 \quad i = 1, \ldots, K. \end{cases} \quad (26)$$

Since (26) has constraints of the format in (1) we can use our conservative (and progressive) techniques to find solutions. □

**Example 5. Product of uncertain nonnegative affine functions.** Let us consider uncertain constraints in the form:

$$\forall \zeta \in \mathcal{U} : \begin{cases} \prod_{i=1}^K ((a^{(i)}(x))^\top \zeta + c_i(x)) \frac{1}{K} \geq \tau, \\ (a^{(i)}(x))^\top \zeta + c_i(x) \geq 0 \quad i = 1, \ldots, K, \end{cases} \quad (27)$$

where $a^{(i)}(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^n$ and $c_i(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ are affine functions for $i = 1, \ldots, K$. If $\tau$ is maximized subject to the constraints (27), then it is equivalent to maximize the geometric mean of uncertain
nonnegative affine functions \((a^{(i)}(x))^\top \zeta + c_i(x) \geq 0, i = 1, \ldots, K\). For simplicity, we consider the special case \(K = 4\), and first reformulate the problem by introducing new variables \(y_1\) and \(y_2\):

\[
\forall \zeta \in U \ \exists y \geq 0 : \begin{cases}
  y_1 y_2 \geq \tau^2 \ (a^{(1)}(x))^\top \zeta + c_1(x) + (a^{(2)}(x))^\top \zeta + c_2(x) \geq y_1^2 \\
  (a^{(3)}(x))^\top \zeta + c_3(x) + (a^{(4)}(x))^\top \zeta + c_4(x) \geq y_2^2 \\
  (a^{(i)}(x))^\top \zeta + c_i(x) \geq 0 & \text{for } i = 1, \ldots, K.
\end{cases}
\]  

(28)

The extension to other values of \(K\) is straightforward. Applying (23) yields the following set of uncertain SOC constraints:

\[
\forall \zeta \in U \ \exists y \geq 0 : \begin{cases}
  \left\| \frac{2\tau}{y_1 - y_2} \right\|_2 \leq y_1 + y_2 \\
  \left( (a^{(1)}(x))^\top \zeta + c_1(x) - (a^{(2)}(x))^\top \zeta - c_2(x) \right)_2 \leq (a^{(1)}(x))^\top \zeta + c_1(x) + (a^{(2)}(x))^\top \zeta + c_2(x) \\
  \left( (a^{(3)}(x))^\top \zeta + c_3(x) - (a^{(4)}(x))^\top \zeta - c_4(x) \right)_2 \leq (a^{(3)}(x))^\top \zeta + c_3(x) + (a^{(4)}(x))^\top \zeta + c_4(x) \\
  (a^{(i)}(x))^\top \zeta + c_i(x) \geq 0 & \text{for } i = 1, \ldots, K.
\end{cases}
\]

\[
\forall \zeta \in U \ \exists y \geq 0 : \begin{cases}
  \left( (a^{(i)}(x))^\top \zeta + c_i(x) \right)_i \geq 0 & \text{for } i = 1, \ldots, K.
\end{cases}
\]

\[\square\]

**Example 6. Uncertain logarithmic constraints.** Let us consider uncertain constraints in the form:

\[
\forall \zeta \in U : |\log((a^{(i)}(x))^\top \zeta) - \log(c_i)| \leq \tau \quad i = 1, \ldots, K,
\]

(29)

where the entries of \(a^{(i)}(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}^n\) are (componentwise) affine functions for \(i = 1, \ldots, K\). We assume \(c_i \in \mathbb{R}_+\), and interpret \(\log((a^{(i)}(x))^\top \zeta)\) as \(-\infty\) when \((a^{(i)}(x))^\top \zeta \leq 0\). Suppose \(\tau\) is minimized subject to the constraints (29). Then it can be understood as approximately solving an overdetermined set of uncertain equations \((a^{(i)}(x))^\top \zeta \approx c_i, i = 1, \ldots, K\), measuring the worst case error by the maximum logarithmic deviation between the numbers \((a^{(i)}(x))^\top \zeta\) and \(c_i\). To cast these constraints as a set of uncertain SOC constraints, first note that:

\[
|\log((a^{(i)}(x))^\top \zeta) - \log(c_i)| = \log\max\left(\frac{a^{(i)}(x)^\top \zeta}{c_i}, \frac{c_i}{a^{(i)}(x)^\top \zeta}\right)
\]

(assuming \((a^{(i)}(x))^\top \zeta > 0\)). Then the constraints (29) is therefore equivalent to:

\[
\forall \zeta \in U \ \exists y \geq 0 : \begin{cases}
  y \leq \Omega \\
  \frac{c_i}{\tau} \leq a^{(i)}(x)^\top \zeta \leq y c_i & \text{for } i = 1, \ldots, K,
\end{cases}
\]

where \(y \in \mathbb{R}\) and \(\Omega = \log \tau\). Applying (23) yields the following set of uncertain SOC constraints:

\[
\forall \zeta \in U \ \exists y \geq 0 : \begin{cases}
  y \leq \Omega \\
  a^{(i)}(x)^\top \zeta \leq y c_i & \text{for } i = 1, \ldots, K \\
  \left\| \frac{2c_i}{y c_i - a^{(i)}(x)^\top \zeta} \right\|_2 \leq y c_i + a^{(i)}(x)^\top \zeta & \text{for } i = 1, \ldots, K.
\end{cases}
\]

\[\square\]
6. Robust regression

In this section, we first consider the robust regression model in Example 1, and then compare the performance of the robust regression model to the well-studied regression models, LASSO and Elastic Net. We use the Diabetes dataset from Efron et al. (2004), with \( m = 442 \) observations and each with 10 features, i.e., \( n_x = 11 \). The ten features are age, BMI, average blood pressure, six blood serum measurements, and gender. The response of interest \( b \in \mathbb{Z}^m_+ \) is a quantitative measure of disease progression one year after baseline. We divide the dataset into a testing set and a training set with the first 132 observations (\( \approx 30\%m \)) and remaining 310 observations (\( \approx 70\%m \)), respectively, and solve the considered models only using the training set. The testing set is used to evaluate the obtained solutions.

6.1. The robust model

Let us consider the robust regression model (2) in Example 1. This model admits a two-stage robust linear reformulation (see §2), and can be solved by the methods in §4. For the 10 features in the Diabetes dataset, we take that the gender of the patients is certain, i.e., \( \zeta_{i,10} = 0, i = 1, \ldots, 310 \), as well as the last column of \( A \), which is defined by us, the modeler, as the all one vector to represent an intercept, and at most \( \Gamma, \Gamma = 1, \ldots, 9 \), out of 9 remaining features can deviate from the original data \( A \), each feature can deviate up to 1%. We assume the response vector \( b \) can be measured exactly. To this end, the following budget uncertainty set is considered:

\[
U = \left\{ \zeta \in \mathbb{R}^{310 \times 11} : \begin{array}{c}
\zeta_{i,10} = 0, \quad \zeta_{i,11} = 1, \\
\frac{\delta}{\xi} \leq \zeta_{i,j}, \quad -\delta \leq \zeta_{i,j}, \quad \xi \leq 1, \\
\sum_{i=1}^{310} \xi_i \leq \Gamma \quad i = 1, \ldots, 310, \quad j = 1, \ldots, 9
\end{array} \right\},
\]

where \( \delta \in \mathbb{R}^9 \), \( \xi \in \mathbb{R}^9 \), and \( A_{ij} \) is the element of \( A \) at the \( i \)-th row and \( j \)-th column. The one-dimensional uncertain parameter \( \delta_j \) for the \( j \)-th feature is the same for all observations. In this way, we can model bias in the measurements of the \( j \)-th feature. All computations are carried out with MOSEK 8.0 (MOSEK ApS (2017)) on an Intel Core(TM) i5-4590 Windows computer running at 3.30GHz with 8GB of RAM. All modeling is done using the modeling package XProg (http://xprog.weebly.com).

<table>
<thead>
<tr>
<th>( \Gamma )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROC</td>
<td>938.9</td>
<td>942.6</td>
<td>946.2</td>
<td>948.5</td>
<td>949.4</td>
<td>950.4</td>
<td>951.2</td>
<td>951.6</td>
<td>951.7</td>
</tr>
<tr>
<td>Time(s)</td>
<td>0.3</td>
<td>0.5</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.4</td>
<td>0.4</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>%Gap</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Table 1 The solution quality and computation time for different values of \( \Gamma \) in (2). ROC denotes the (conservatively approximated) objective values obtained from solving (10), and Time(s) is the corresponding computational time in seconds. %Gap denotes the optimality gap, i.e., \( \%\text{Gap} = \frac{\text{LDR-OPT}}{\text{OPT}} \), where OPT denotes the optimal objective values.
From Table 1 one may observe that the approximated solutions of (2) from solving (10) can be computed efficiently, and they are very close to optimality. Since the extreme points are not too many in $\mathcal{U}$, we enumerate all the extreme points to obtain the optimal solutions.

6.2. Models for comparison

We compare the performance of the robust regression model (2) to some well-studied regression models. The first model for comparison is the LASSO model of Tibshirani (1996), which is a least squares regression model with a $l_1$-regularizer:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2m} \|Ax - b\|_2^2 + \Delta \|x\|_1,$$

where $\Delta$ is a regularization parameter. For the given data, we use the built-in MATLAB function

$$x = \text{lasso}(A,b,CV,5)$$

to compute $x$ for 100 different values of $\Delta$ (default), and select the one with the lowest prediction error. As suggested in Tibshirani (1996), the prediction error is estimated by fivefold cross-validation, i.e., $CV=5$.

The other popular variant of least-squares regression is Elastic Net regression (Zou and Hastie (2005)):

$$\min_{x \in \mathbb{R}^n} \frac{1}{2m} \|Ax - b\|_2^2 + \frac{1-\alpha}{2} \Delta \|x\|_2^2 + \alpha \Delta \|x\|_1,$$

We again use the built-in MATLAB function

$$x = \text{lasso}(A,b,CV,5,\alpha,0.1)$$

to compute $x$. Here we use the same 100 values of $\Delta$ as for LASSO. The prediction error is again estimated by fivefold cross-validation. We apply a binary search to find the $\alpha$ that gives the lowest estimated prediction error for the training data, i.e., $\alpha = 0.1$. Xu et al. (2009) shows that LASSO can be seen as the robust counterpart of a robust regression model with a specific uncertainty set. In Bertsimas and Copenhaver (2017) it is shown that a similar result holds for Elastic Net. Furthermore, they show that a closed form expression of the robust counterpart can only be derived for uncertainty sets that consist solely of seminorms. Therefore, robust regression models with general polyhedral uncertainty set, as considered in this paper, can not be cast as an equivalent closed form formulation.

For both regression via LASSO and Elastic Net regularization it must be noted that they can be solved very efficiently. The Elastic Net regression can be reduced to the Support Vector Machine, and efficiently solved by the Support Vector Machine solvers (see Zhou et al. (2015)). The pathwise
Table 2 The solutions $x$ for different values of $\Gamma$ in the robust regression model, LASSO and Elastic Net.

<table>
<thead>
<tr>
<th>$\Gamma$</th>
<th>$\Gamma = 0$</th>
<th>$\Gamma = 1$</th>
<th>$\Gamma = 2$</th>
<th>$\Gamma = 3$</th>
<th>$\Gamma = 4$</th>
<th>$\Gamma = 5$</th>
<th>$\Gamma = 6$</th>
<th>$\Gamma = 7$</th>
<th>$\Gamma = 8$</th>
<th>$\Gamma = 9$</th>
<th>LASSO</th>
<th>EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>$x_2$</td>
<td>6.0</td>
<td>6.1</td>
<td>6.0</td>
<td>6.1</td>
<td>6.1</td>
<td>6.2</td>
<td>6.2</td>
<td>6.2</td>
<td>6.2</td>
<td>6.0</td>
<td>6.0</td>
<td>5.6</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1.1</td>
<td>1.1</td>
<td>1.2</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>$x_4$</td>
<td>-1.4</td>
<td>-1.1</td>
<td>-0.8</td>
<td>-0.6</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.1</td>
<td>-0.1</td>
<td>-0.1</td>
<td>-0.2</td>
<td>-1.4</td>
<td>-1.1</td>
</tr>
<tr>
<td>$x_5$</td>
<td>1.1</td>
<td>0.8</td>
<td>0.5</td>
<td>0.3</td>
<td>-0.1</td>
<td>-0.1</td>
<td>-0.1</td>
<td>-0.1</td>
<td>-0.1</td>
<td>0.0</td>
<td>1.0</td>
<td>-0.1</td>
</tr>
<tr>
<td>$x_6$</td>
<td>0.9</td>
<td>0.5</td>
<td>0.2</td>
<td>-0.2</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.7</td>
<td>-0.7</td>
<td>-0.6</td>
<td>0.8</td>
<td>-0.7</td>
</tr>
<tr>
<td>$x_7$</td>
<td>10.0</td>
<td>9.6</td>
<td>9.0</td>
<td>8.1</td>
<td>6.6</td>
<td>6.8</td>
<td>6.5</td>
<td>5.8</td>
<td>5.7</td>
<td>5.7</td>
<td>9.9</td>
<td>5.5</td>
</tr>
<tr>
<td>$x_8$</td>
<td>70.0</td>
<td>59.8</td>
<td>52.2</td>
<td>45.0</td>
<td>35.2</td>
<td>35.1</td>
<td>35.3</td>
<td>34.4</td>
<td>34.7</td>
<td>37.0</td>
<td>69.3</td>
<td>33.5</td>
</tr>
<tr>
<td>$x_9$</td>
<td>0.2</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>-17.0</td>
<td>-17.0</td>
<td>-17.2</td>
<td>-16.9</td>
<td>-16.5</td>
<td>-16.3</td>
<td>-16.1</td>
<td>-15.9</td>
<td>-15.9</td>
<td>-15.9</td>
<td>-17.0</td>
<td>-14.1</td>
</tr>
<tr>
<td>$x_{11}$</td>
<td>-388.1</td>
<td>-352.5</td>
<td>-323.1</td>
<td>-293.1</td>
<td>-253.7</td>
<td>-255.0</td>
<td>-254.7</td>
<td>-244.9</td>
<td>-245.4</td>
<td>-253.3</td>
<td>-385.3</td>
<td>-238.2</td>
</tr>
</tbody>
</table>

coordinate descent algorithm for LASSO is among the most efficient algorithms among all convex optimization algorithms that exist and can solve instances with many millions of variables easily (Friedman et al. (2007)). The LASSO is in particular popular because it encourages sparsity in the subset selection. We do not envisage the robust optimization model with polyhedral uncertainty sets for those specific purposes in which those methods excel. However, we do believe there is a very nice interpretation to the budget uncertainty sets for practitioners.

6.3. Results

We use various performance metrics to compare the performance of the robust regression model (2), LASSO regression and Elastic Net regression. In Table 2, the solutions of the robust regression for different values of $\Gamma$, as well as the solutions of LASSO regression and Elastic Net regression are presented. For $\Gamma = 0$ in (2), we refer the corresponding solution to the nominal solution. The difference between the LASSO solution and the nominal solution is very small. Notice that the obtained robust solutions resides “in between” of the solutions from LASSO and Elastic Net.

In Figure 1, the solutions for different $\Gamma$ are evaluated. It shows that the proposed solution is the most robust against the worst case scenarios, whereas the LASSO solution and the nominal solution are very sensitive to those scenarios. However, the LASSO solution and the nominal solution have the lowest expected mean squared error (MSE) if the scenarios are uniformly distributed in $\mathcal{U}$. Our robust solutions are comparable with the LASSO solution and the nominal solution for the out-of-sample test. For this testing set, the robust solution for $\Gamma = 9$ has the lowest MSE.
Figure 1 The worst case, expected and out-of-sample mean squared errors of different models. Robust, LASSO and EN denote the solutions for the robust model, LASSO, Elastic Net. The nominal solution is denoted as Nom. The worst case MSE is obtained via a complete enumeration of all the vertices of $\mathcal{U}$. The expected MSE is estimated by averaging the MSEs for $10^5$ uniformly generated scenarios in $\mathcal{U}$. The out-of-sample MSE is obtained from evaluating the solutions using the testing set. Note that the scales of the MSE-axis are different for different tests. The ranges are 300, 60 and 25, respectively.
7. Robust sensor network

Here we consider a problem where there are $N$ points in $\mathbb{R}^2$ that must be connected by links. Some of the $N$ points are already placed and the decision maker has to decide where to place the remaining points. The goal is to minimize the total distance of all the links together. An application would be where the points represent wireless sensors and modules on a network that are interconnected and one wants to minimize the total energy needed for the wireless transmission over the links. This example is based on the placement and location problem, for which the nominal case is described in (Boyd and Vandenberghe 2004, Section 8.7). In the nominal case the model can be written as

$$\min \sum_{(i,j) \in A} \|y_i - y_j\|_2$$

s.t. $y_i = \bar{a}_i \quad \forall i \in L$,

where $y_i \in \mathbb{R}^2$ are the locations of the points for all $i = 1, \ldots, N$. The points that are fixed (already placed) are given by the set $L$ and their locations by $\bar{a}_i \in \mathbb{R}^2$. The remaining points not in $L$ are free to set by the optimizer. The set of prescribed (undirected) links is given by $A$.

7.1. The robust model

Here we consider the case where the locations of the fixed sensors $\bar{a}_i$ are not precisely known. This uncertainty in the locations could be due to sea currents for sensors placed at sea, wind drift for sensors are dropped from planes or other errors due to placement from catapults or missiles (see e.g. Akyildiz et al. (2002)). Here we model the uncertainty in the locations as:

$$a_i(\zeta) = \bar{a}_i + \hat{a}_i \zeta_i,$$

where $\hat{a}_i \in \mathbb{R}_+$ is the maximal (absolute) deviation from the nominal value $\bar{a}_i$ for all $i \in L$. The uncertain parameter $\zeta = (\zeta_1, \ldots, \zeta_{|L|})^T \in \mathbb{R}^{2|L|}$, where $\zeta_i^T \in \mathbb{R}^2$ for all $i \in L$, resides in a lifted budget uncertainty set $\mathcal{U}$ defined by

$$\mathcal{U} = \left\{ (\zeta, \xi) : \zeta \leq \xi, -\xi \leq \zeta \leq 1, \sum_{i=1}^{2|L|} \xi_i \leq \Gamma \right\},$$

where $\xi \in \mathbb{R}^{2|L|}$ and $\Gamma \geq 0$ is called the budget of uncertainty. Projecting $\mathcal{U}$ on the space of $\zeta$, one can recover the classical budget uncertainty set $\{ \zeta \in \mathbb{R}^{2|L|} : -1 \leq \zeta \leq 1, \|\zeta\|_1 \leq \Gamma \}$ of Bertsimas and Sim (2004). We use the lifted budget uncertainty set here because the improvement in solution quality often worth of the small extra computation effort due to the extra variable (de Ruiter and Ben-Tal (2017)). Some of the modules $x_i$ need to be placed before the exact locations of $a_i(\zeta)$ are known, whereas others can be placed after. We define the set of indices $H$ for those modules that have to be placed before the sensor locations are known. We associate a here-and-now variable $x_i \in \mathbb{R}^2$ for every $i \in H$. The robust model can now be written as:

$$\min \tau$$

s.t. $\forall (\zeta, \xi) \in \mathcal{U} :$

$$\begin{cases} \sum_{(i,j) \in A} \|y_i(\zeta, \xi) - y_j(\zeta, \xi)\|_2 \leq \tau \\ y_i(\zeta, \xi) = \bar{a}_i + \hat{a}_i \zeta_i \quad \forall i \in L \\ y_i(\zeta, \xi) = x_i \quad \forall i \in H. \end{cases}$$
For ease of exposition, we use the wait-and-see decision $y$ to represent the location of the sensors and modules. One can eliminate $y$ using the equalities, and model (30) then has constraints of the form (1). We first use Theorem 1 to reformulate the constraints (30) into a set of two-stage robust linear constraints, then we solve the resulting model via the solution methods in §4. The objective value of (30) gives the total energy required for the wireless transmissions in the network.

7.2. Numerical setting

For the experiments we use two sets of data. For illustrative purposes we first consider a small instance with $N = 14$ points, of which 8 nominal sensor locations are uncertain and 6 modules need to be placed. For this, data from (Boyd and Vandenberghe 2004, Section 8.7.3) is used, see Figure 2. Data is obtained from the CVX website http://web.cvxr.com/cvx/examples. The maximal deviation from the nominal locations is taken to be $\hat{a}_i \in \{0, 0.2, 0.5, 1\}$ for all $i \in L$. For instance for $\hat{a}_i = 0.2$, $i \in L$, Figure 3 illustrates the robust solution obtained from solving (30) via linear decision rules.

The second set of data shows results for larger instances. We choose $|L| \in \{10, ..., 70\}$ nominal sensor locations $\bar{a}_i$, $i \in L$ uniformly at random from $[-1, 1]^2$. The maximal deviation from the

Figure 2  Nominal solution from (Boyd and Vandenberghe 2004, Figure 8.16). The squares represent the sensors. The dots denote the modules. The prescribed (undirected) links between the sensors and modules are plotted using dashed lines.
nominal locations is $\hat{a}_i = 0.3$ for all $i \in L$. We have $0.4|L|$ modules that have to collect data from the sensors. Each module is randomly linked with $|L|$ sensors. The modules are randomly linked into a cycle. We link the sensors that are not connected with each of the $0.4|L|$ modules.

Table 3  Sensor network model with $N = 14$. ROC is the conservative approximation obtained from solving (9) in §4.1. LB(1) denotes the lower bounds obtained from one scenario. Time(s) reports the computation time (in seconds) for solving ROC.

<table>
<thead>
<tr>
<th>$\hat{a}_i$</th>
<th>0</th>
<th>0.2</th>
<th>0.5</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROC</td>
<td>21.91</td>
<td>23.86</td>
<td>26.88</td>
<td>32.07</td>
</tr>
<tr>
<td>LB(1)</td>
<td>21.91</td>
<td>23.79</td>
<td>26.64</td>
<td>31.46</td>
</tr>
<tr>
<td>Time(s)</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Figure 3  Robust solution for $\hat{a}_i = 0.2$, $i \in L$, with the nominal sensor locations. The squares represent the sensors. The dots denote the modules. The prescribed (undirected) links between the sensors and modules are plotted using dashed lines.

7.3. Results

We use the same computer and optimization software as mentioned in §6. Figure 4 depicts the robust solutions for $|L| = 30$. For the small instance with $N = 14$ points, as $\hat{a}_i$, $i \in L = \{1, \ldots, 8\}$, increases, the obtained objective values ROC from solving (30) via linear decision rules become larger (see Table 3), and the optimal module locations are slightly more spread out (see Figure...
The lower bounds \( LB(1) \) are obtained from solving (14) with only one scenario. This scenario is obtained from (17) using the scenario obtained from (16). The differences between the lower bounds \( LB(1) \) and upper bounds \( ROC \) are within 1%. This indicates that the optimality gap of the obtained objective values is at most 1% of the optimal value. For medium and large instances considered in Table 4, the approximated solutions of (30) via linear decision rules are near optimal, and can be computed efficiently.

**Figure 4** Robust solution for \( \hat{a}_i = 0.3, i \in L, \) and \( |L| = 30 \), with the nominal sensor locations. The squares represent the sensors. The dots denote the modules. The prescribed (undirected) links between the sensors and modules are plotted using dashed lines.

**Table 4** Sensor network model with \( \hat{a}_i = 0.3, i \in L, \) where \( |L| = \{10, ..., 70\} \). \( ROC \) is the conservative approximation obtained from solving (9) in §4.1. \( LB(1) \) denotes the lower bounds obtained from one scenario. \( Time(s) \) reports the computation time (in seconds) for obtaining \( ROC \). All the numbers are the average of 10 randomly generated instances.

| \( |L| \) | 10 | 20 | 30 | 40 | 50 | 60 | 70 |
|---|---|---|---|---|---|---|---|
| \( ROC \) | 24.4 | 70.8 | 149.5 | 257.0 | 392.0 | 564.1 | 762.9 |
| \( LB(1) \) | 24.3 | 70.8 | 149.4 | 256.9 | 391.9 | 564.1 | 762.9 |
| \( Time(s) \) | 0.35 | 2.4 | 10.4 | 30.6 | 59.4 | 102.5 | 193.3 |
8. Future research
On a theoretical level, one immediate future research direction would be to establish the optimality of the conservative approximations via linear decision rules for some uncertain SOC and SDP constraints with polyhedral uncertainty.

On a numerical level, we would like to conduct experiments on problems with constraints in the forms that are discussed in §5 to further evaluate the efficiency of the proposed methods.

Acknowledgments
The research of the first author is supported by the Netherlands Organisation for Scientific Research (NWO) Grant 613.001.208. The second author is partially supported by the NWO Research Talent Grant 406-14-067.
References


Proof of Lemma 3

Lemma 3. A matrix $Q \in \mathbb{R}^{m \times m}$ is positive semidefinite if and only if the trace of the product with any positive semidefinite matrix is positive.

Proof. “$\Leftarrow$”: Suppose for any $c \in \mathbb{R}^m$, $\text{Tr}(QW) \geq 0$, where $W = cc^\top \succeq 0$. Since $\text{Tr}(QW) = \text{Tr}(Qcc^\top) = c^\top Qc \geq 0$ holds for any $c \in \mathbb{R}^m$, by definition, $Q \succeq 0$. “$\Rightarrow$”: Suppose $Q \succeq 0$. For any $W \succeq 0$, there exists a $C$ such that $W = C^\top C$. We then have $\text{Tr}(QW) = \text{Tr}(QC^\top C) = \text{Tr}(CQC^\top) \geq 0$ because $Q \succeq 0$ implies $CQC^\top \succeq 0$. \hfill \Box

B. Robust counterparts of (5) with lifted uncertainty set $\hat{W}$

By imposing the linear decision rule (11), we derive the tractable robust counterparts of (5) with the lifted uncertainty set $\hat{W}$:

$$\forall (w, z) \in \hat{W} : \begin{cases} 
    d^\top (v + Vw + Uz) + b(x)^\top w \leq c(x) \\
    D^\top (v + Vw + Uz) \geq a(x) + A(x)^\top w \\
    v + Vw + Uz \geq 0.
\end{cases}$$

By using reformulation techniques (Ben-Tal et al. (2015)), the tractable robust counterparts can be derived:

$$\begin{align*}
    &\left\{ \begin{array}{l}
        d^\top v + \sum_{i=1}^{m} \kappa_i^{(1)} + \tau_i^{(1)} \leq c(x) \\
        \left\| 2(d^\top V_i + b_i(x)) \right\|_2 \leq 4\sigma_i^{(1)} + \kappa_i^{(1)} & i = 1, \ldots, m \\
        d^\top U_i + \sigma_i^{(1)} - \tau_i^{(1)} \leq 0 & i = 1, \ldots, m \\
        a_j(x) + \sum_{i=1}^{m} \kappa_{ij}^{(2)} + \tau_{ij}^{(2)} \leq D_j^\top v & j = 1, \ldots, n \\
        \left\| 2(A_{ij}(x) - V_i^\top D_j) \right\|_2 \leq 4\sigma_{ij}^{(2)} + \kappa_{ij}^{(2)} & i = 1, \ldots, m \ j = 1, \ldots, n \\
        -D_j^\top U_i + \sigma_{ij}^{(2)} - \tau_{ij}^{(2)} \leq 0 & i = 1, \ldots, m \ j = 1, \ldots, n \\
        \sum_{i=1}^{m} \kappa_{il}^{(3)} + \tau_{il}^{(3)} \leq v_l & l = 1, \ldots, r \\
        \left\| -2V_{il} \right\| \leq 4\sigma_{il}^{(3)} + \kappa_{il}^{(3)} & i = 1, \ldots, m \ l = 1, \ldots, r \\
        -U_{li} + \sigma_{il}^{(3)} - \tau_{il}^{(3)} \leq 0 & i = 1, \ldots, m \ l = 1, \ldots, r \\
    \end{array} \right. \\
    \sigma_i^{(1)}, \sigma_{ij}^{(2)}, \sigma_{il}^{(3)}, \tau_i^{(1)}, \tau_{ij}^{(2)}, \tau_{il}^{(3)} \geq 0 & i = 1, \ldots, m \ j = 1, \ldots, n \ l = 1, \ldots, r.
\end{align*}$$

\hfill \Box