Approximation of hard uncertain convex inequalities

Ernst Roos*, Dick den Hertog
CentER and Department of Econometrics and Operations Research, Tilburg University

Aharon Ben-Tal
Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology
Shenkar College, Israel

Frans J.C.T. de Ruiter
CQM, Eindhoven, The Netherlands

Jianzhe Zhen
Risk Analytics and Optimization Chair,
École Polytechnique Fédérale de Lausanne, Switzerland

June 27, 2018

Abstract
Robust Optimization is a widespread approach to treat uncertainty in optimization problems. Finding a computationally tractable formulation of the robust counterpart of an uncertain optimization problem is a key step in applying this approach. Although techniques for finding a robust counterpart are available for many types of constraints, no general techniques exist for problems requiring maximizing functions that are convex in the uncertain parameters. Such constraints are, however, quite common. In this paper, we treat these hard problems and provide a systematic way to construct a safe approximation to their robust counterpart given a polyhedral uncertainty set. We use convex analysis as well as adjustable robust optimization techniques to obtain these approximations. We demonstrate the quality of the approximations by performing numerical experiments.

*Corresponding author: e.j.roos@tilburguniversity.edu
1 Introduction

Optimization problems frequently contain uncertain parameters in practice. This uncertainty arises as a result of prediction, estimation or implementation errors, for example. Robust Optimization (RO) is one of the main approaches to address such uncertainty and was initiated by Ben-Tal and Nemirovski (1998); El Ghaoui and Lebret (1997).

Whereas other approaches, such as Stochastic Programming or Distributionally Robust Optimization, assume that the uncertain parameters are random variables with fully or partially known distribution functions, RO models uncertainty by means of an uncertainty set. This set contains all possible scenarios one wants to safeguard against.

One of the main advantages of robust optimization is the computational tractability of the resulting optimization problems. Indeed, for many types of robust problems it has been shown that its robust counterpart is tractable. Ben-Tal et al. (2009) gives a comprehensive account of robust optimization techniques used to obtain such tractable formulations. A general approach to finding computationally tractable reformulations is presented in Ben-Tal et al. (2015), where a technique is presented that is applicable to any constraint that is convex in the optimization variables and concave in the uncertain parameters.

In contrast, obtaining a tractable robust counterpart for constraints that are convex in the uncertain parameters is, in general, hard. Such constraints are, however, common; they appear, for example, in inventory management problems, geometric programming and conic optimization. For specific types of constraints such as second-order cone (SOC) and semidefinite programming (SDP) constraints combined with specific uncertainty sets, e.g., norm-bounded or polyhedral uncertainty sets, computationally tractable approximations exist. Some of the first techniques addressing these problems are discussed in El Ghaoui and Lebret (1997); El Ghaoui et al. (1998); Ben-Tal et al. (2002) and a more recent technique is discussed in Zhen et al. (2017).

Distributionally Robust Optimization (DRO) is an alternative approach to RO that assumes the uncertain parameter to follow some unknown distribution residing in an ambiguity set. While DRO is capable of dealing with functions that are convex in the uncertain parameters, it generally does not yield tractable robust counterparts.

In this paper, we consider constraints that are convex in both the optimization variables and the uncertain parameters where the latter reside in polyhedral uncertainty sets. More specifically, we convert the robust counterpart to an equivalent problem containing adjustable variables; this enables us to treat the converted problem with existing techniques for Adjustable Robust Optimization (ARO). In particular, using linear decision rules, the resulting problem is a safe approximation to the true robust counterpart. This technique is used to treat two examples of problems for which no computationally
tractable exact reformulation of the robust counterpart is known: sum-of-max constraints and geometric programming. We also provide numerical results for geometric programming problems to illustrate the quality of the safe approximation provided in this paper.

The paper is organized as follows. Section 2 treats the general reformulation we propose. In Section 3 and 4 we treat sum-of-max constraints and geometric programming, respectively. Section 5 contains numerical results for geometric programming problems.

2 The Robust Counterpart

2.1 Reformulation to ARO

In this paper, we consider a general convex constraint given by

$$f(A(x)\zeta + b(x)) \leq 0 \quad \forall \zeta \in U,$$

where $f : \mathbb{R}^p \mapsto \mathbb{R}$ is convex and closed, $A : \mathbb{R}^n \mapsto \mathbb{R}^{p \times L}$, $b : \mathbb{R}^n \mapsto \mathbb{R}^p$ are affine, and $U$ is a nonempty polyhedron given by

$$U = \{ \zeta \in \mathbb{R}^L \mid D\zeta = d \},$$

for some $D \in \mathbb{R}^{q \times L}$ and $d \in \mathbb{R}^q$.

This formulation allows for many important classes of constraints, such as sum-of-max and log-sum-exp functions, which we will both discuss later in the paper. Other examples are (sums of) norms and negative entropy. Furthermore, all functions $g(\zeta, x)$ that are jointly convex in $\zeta, x$ can be written as $f(A(x)\zeta + b(x))$ for some affine functions $A(x), b(x)$. A direct implication of this remark is that we can also treat constraints $g(x, y)$ that are jointly convex in $x$ and $y$, where $y$ is an adjustable variable, by substituting a linear decision rule for $y$. This does, however, yield an Adjustable Robust Optimization problem that is not equivalent but a safe approximation to the original constraint.

It is crucial to remark that by choosing to consider the function $f(A(x)\zeta + b(x))$ we cannot handle all constraints convex in both $x$ and $\zeta$. This means that we cannot, for example, treat geometric programming constraints with uncertainty in both the coefficients and exponents of the posynomials.

It is also important to discuss the added value of our approach for a more general convex function over the previously developed approaches for second-order cone programming (SOCP) and semidefinite programming (SDP) constraints by Zhen et al. (2017). Here, it is important to remark that while many (nominal) problems can be reformulated as an SOCP, the robust counterparts of these equivalent problems is not necessarily the
same. As an example, we consider a constraint on the sum of 2-norms:

\[
\sum_{i=1}^{m} \|A_i(x)\zeta + b_i(x)\|_2 \leq \tau \quad \forall \zeta \in U,
\]

and its SOCP reformulation

\[
\begin{cases}
\sum_{i=1}^{m} y_i \leq \tau \\
\|A_i(x)\zeta + b_i(x)\|_2 \leq y_i 
\end{cases} \quad \forall \zeta \in U, \quad i = 1, \ldots, m.
\]

While the non-robust versions of (2) and (3) are clearly equivalent, (2) and (3) are in fact only equivalent if the newly introduced variables \(y_i\) are considered to be adjustable (Gorissen et al., 2015). This means that in order to apply our proposed technique or techniques for SOCP problems to (3), some form of decision rule should be chosen for \(y_i\) and thus the above-mentioned techniques would be applied to a problem that is ‘only’ a safe approximation of the problem truly of interest. This implies that in general, directly applying our techniques to a convex optimization problem will yield a tighter approximation to its robust counterpart than applying this paper’s technique to its second-order cone reformulation.

In order to find a safe approximation to (1), we first use some convex analysis to transform the problem into an equivalent adjustable robust optimization problem.

**Theorem 1.** Let \(f : \mathbb{R}^p \rightarrow \mathbb{R}\) be a closed convex function and let \(A : \mathbb{R}^n \rightarrow \mathbb{R}^{p \times L}\), \(b : \mathbb{R}^n \rightarrow \mathbb{R}^p\) be affine functions. Let \(U \subseteq \mathbb{R}^L\) be a polyhedron, that is,

\[
U = \{\zeta \in \mathbb{R}_+^L : D\zeta = d\},
\]

for \(d \in \mathbb{R}^q\), \(D \in \mathbb{R}^{q \times L}\). Then, \(x \in \mathbb{R}^n\) satisfies

\[
f(A(x)\zeta + b(x)) \leq 0 \quad \forall \zeta \in U,
\]

if and only if it satisfies the set of adjustable robust optimization constraints:

\[
\forall w \in \text{dom } f^*, \exists \lambda \in \mathbb{R}^q : \begin{cases}
d^\top \lambda + w^\top b(x) - f^*(w) \leq 0 \\
D^\top \lambda \preceq A(x)^\top w,
\end{cases}
\]

where \(f^*(w) : \mathbb{R}^p \rightarrow \mathbb{R}\) is the convex conjugate of \(f\).

**Proof.** Because \(f\) is a closed convex function we have that

\[
f(z) = f^{**}(z) = \sup_{w \in \text{dom } f^*} \{z^\top w - f^*(w)\}.
\]
Substituting this into (1) yields
\[ \forall \zeta \in U : f(A(x)\zeta + b(x)) \leq 0 \]
\[ \iff \forall \zeta \in U : \sup_{w \in \text{dom } f^*} \left\{ \left( A(x)\zeta + b(x) \right)^\top w - f^*(w) \right\} \leq 0 \]
\[ \iff \sup_{w \in \text{dom } f^*} \left\{ \sup_{\zeta \in U} \left\{ \left( A(x)^\top \zeta \right)^\top w + b(x)^\top w - f^*(w) \right\} \right\} \leq 0 \]
\[ \iff \sup_{w \in \text{dom } f^*} \left\{ \inf_{\lambda \in \mathbb{R}^q} \left\{ d^\top \lambda | D^\top \lambda \geq A(x)^\top w + b(x)^\top w - f^*(w) \right\} \right\} \leq 0 \]
\[ \iff \forall w \in \text{dom } f^*, \exists \lambda \in \mathbb{R}^q : \left\{ d^\top \lambda + b(x)^\top w - f^*(w) \leq 0 \right\} \]
\[ D^\top \lambda \geq A(x)^\top w \]
where (5) and (6) are equivalent because of strong LP duality.

We note that a similar result holds if the nonnegativity constraint in $U$ is omitted. Then the inequality in (4b) becomes an equality constraint, which can be used to eliminate some adjustable variables $\lambda$. Eliminating variables in this way is equivalent to imposing linear decision rules (Zhen and Den Hertog, 2017, Lemma 2). We also remark that for ellipsoidal uncertainty sets, polyhedral approximations from (Ben-Tal and Nemirovski, 2001) can be used to enable the use of the theorems derived in this paper. Once again, the ARO problem obtained in Theorem 1 then no longer yields an equivalent problem, but a safe approximation instead.

### 2.2 Safe Approximation

Since the equivalent formulation derived in Theorem 1 is a set of adjustable robust linear constraints, conventional techniques for such problems can be applied. Possible techniques to be applied include, but are not limited to, finite adaptability, see e.g. (Bertsimas and Caramanis, 2007; Postek and Den Hertog, 2016), Fourier-Motzkin elimination (Zhen et al., 2016) and specifying a class of decision rules, done by (Ben-Tal et al., 2004; Bertsimas et al., 2011). A common technique to obtain a computationally tractable safe approximation to adjustable robust constraints is limiting the adjustable variables to be linear decision rules in the uncertain parameters as first introduced by Ben-Tal et al. (2004). The result of this technique applied to the problem in this paper is stated in Theorem 2.

**Theorem 2.** If there exist $u \in \mathbb{R}^q$ and $V \in \mathbb{R}^{q \times p}$ for a given $x \in \mathbb{R}^n$ such that
\[
\begin{align*}
\left\{ \begin{array}{l}
d^\top u + f \left( b(x) + V^\top d \right) \leq 0 \\
\delta^* \left( A_i(x) - V^\top D_i \big| \text{dom } f^* \right) \leq D_i^\top u & i = 1, \ldots, L,
\end{array} \right. \tag{8a}
\end{align*}
\]

\[
\begin{align*}
\delta^* \left( A_i(x) - V^\top D_i \big| \text{dom } f^* \right) \leq D_i^\top u & i = 1, \ldots, L,
\end{align*}
\]

\[
\begin{align*}
\left\{ \begin{array}{l}
d^\top u + f \left( b(x) + V^\top d \right) \leq 0 \\
\delta^* \left( A_i(x) - V^\top D_i \big| \text{dom } f^* \right) \leq D_i^\top u & i = 1, \ldots, L,
\end{array} \right. \tag{8b}
\end{align*}
\]
holds, then $x$ also satisfies (1).

**Proof.** We know from Theorem 1 that (1) is equivalent to (4). In this adjustable robust optimization problem, $\lambda$, the adjustable variable, can be thought of as a general decision rule. Here, we restrict $\lambda$ to be a linear decision rule, that is,

$$\lambda = u + Vw.$$  

Substituting this decision rule yields

$$
\begin{align*}
  d^\top \lambda + b(x)^\top w - f^*(w) &\leq 0 & \forall w \in \text{dom } f^* \\
  \iff d^\top u + \sup_{w \in \text{dom } f^*} \left\{ (b(x) + V^\top d)^\top w - f^*(w) \right\} &\leq 0 \\
  \iff d^\top u + f(b(x) + V^\top d) &\leq 0,
\end{align*}
$$

in the first constraint. For the second constraint we find

$$
\begin{align*}
  D_i^\top \lambda &\geq A_i(x)^\top w & \forall w \in \text{dom } f^*, \ i = 1, \ldots, L \\
  \iff \delta^* \left( A_i(x) - V^\top D_i \mid \text{dom } f^* \right) &\leq D_i^\top u & \ i = 1, \ldots, L,
\end{align*}
$$

of which the tractability is dependent on $\text{dom } f^*$.

In general, the tractability of the resulting problem (8) is dependent on the original function $f$ and $\text{dom } f^*$. Section 4 on geometric programming describes an example for which problem (8) can be solved about as efficiently as the original nominal problem. Moreover, if $\text{dom } f^*$ is hard to determine, it is easy to find a weaker safe approximation of (1) by replacing $\text{dom } f^*$ by some outer approximation.

A tighter safe approximation than the one described in Theorem 2 can be found by lifting the nonlinearity of $f^*(w)$ to the uncertainty set and using a slightly more involved decision rule.

**Theorem 3.** If there exist $u \in \mathbb{R}^q$, $V \in \mathbb{R}^{q \times p}$ and $r \in \mathbb{R}^q$ for a given $x \in \mathbb{R}^n$ such that

$$
\begin{align*}
  d^\top u + (1 + d^\top r) f \left( \frac{V^\top d + b(x)}{1 + d^\top r} \right) &\leq 0 & 1 + d^\top r > 0 \\
  -D_i^\top u + (-D_i^\top r) f \left( \frac{A_i(x) - V^\top D_i}{-D_i^\top r} \right) &\leq 0 & -D_i^\top r > 0 & \ i = 1, \ldots, L,
\end{align*}
$$

holds, then $x$ also satisfies (1). Here, $D_i$ and $A_i(x)$ denote the $i$-th column of $D$ and $A(x)$, respectively. Moreover, (10) is a tighter safe approximation than (8).

---

6
Proof. From Theorem 1 we know (1) is equivalent to (4). We lift the nonlinear term \( f^*(w) \) to the uncertainty set, that is, we introduce an auxiliary uncertain parameter \( w_0 \) such that we find that (4) is equivalent to

\[
\forall \left( \begin{array}{c} w_0 \\ w \end{array} \right) \in W, \exists \lambda \in \mathbb{R}^q : \begin{cases} 
 d^\top \lambda + w^\top b(x) + w_0 \leq 0 \\
 D^\top \lambda \geq A(x)^\top w,
\end{cases}
\]

(11a)

(11b)

where the new uncertainty set \( W \) is defined by

\[
W = \left\{ \left( \begin{array}{c} w_0 \\ w \end{array} \right) \in \mathbb{R}^{p+1} \g | w_0 + f^*(w) \leq 0 \right\}.
\]

(12)

The support function of this new uncertainty set is essential for deriving a tractable robust counterpart and is equal to:

\[
\delta^* \left( \left( \begin{array}{c} z_0 \\ z \end{array} \right) \g | W \right) = \sup_{(w_0 \ w)^\top \in W} \{ z_0w_0 + z^\top w \}
\]

\[
= \begin{cases} 
 \sup_{w \in \mathbb{R}^p} \{ z^\top w - z_0f^*(w) \} & \text{if } z_0 \geq 0 \\
 +\infty & \text{otherwise}
\end{cases}
\]

\[
= \begin{cases} 
 z_0f \left( \frac{z}{z_0} \right) & \text{if } z_0 > 0 \\
 +\infty & \text{otherwise}.
\end{cases}
\]

(13)

We note that \( z_0 \geq 0 \) becomes \( z_0 > 0 \) in this derivation, as for \( z_0 = 0 \) we find that \( \sup_{w \in \mathbb{R}^p} \{ z^\top w - z_0f^*(w) \} = +\infty \). Now, we once again use a linear decision rule for \( \lambda \) of the form

\[
\lambda = u + Vw + rw_0,
\]

(14)

where \( u \in \mathbb{R}^q, V \in \mathbb{R}^{q \times p} \) and \( r \in \mathbb{R}^q \), and thus we obtain a safe approximation for (1). Substituting this decision rule in (4a) yields

\[
\begin{align*}
\forall \left( \begin{array}{c} w_0 \\ w \end{array} \right) & \in W
\Longleftrightarrow \begin{cases} 
 d^\top \lambda + b(x)^\top w + w_0 \leq 0 \\
 d^\top (u + Vw + rw_0) + b(x)^\top w + w_0 \leq 0
\end{cases} \\
\Longleftrightarrow \begin{cases} 
 d^\top u + \left( \begin{array}{c} w_0 \\ w \end{array} \right)^\top \left( \begin{array}{c} 1 + d^\top r \\ V^\top d + b(x) \end{array} \right) \leq 0 \\
 d^\top u + \delta^* \left( \begin{array}{c} 1 + d^\top r \\ V^\top d + b(x) \end{array} \g | W \right) \leq 0
\end{cases} \\
\Longleftrightarrow \begin{cases} 
 d^\top u + \delta^* \left( \begin{array}{c} 1 + d^\top r \\ V^\top d + b(x) \end{array} \g | W \right) \leq 0 \\
 \left\{ d^\top u + (1 + d^\top r)^\top f \left( \frac{V^\top d + b(x)}{1 + d^\top r} \right) \leq 0 \right\} \Longleftrightarrow 1 + d^\top r > 0,
\end{cases}
\end{align*}
\]

(15)
where the last equivalence holds because of the definition of the support of $W$ in (13), and the constraint $1 + d^T r > 0$ is introduced because it is required by the domain of $\delta^*$. Note that (15) is exactly (10a). Similarly, substituting the linear decision rule for $\lambda$ in (4b) we find

\[
D^T \lambda \geq A(x)^T w \quad \forall \left(\begin{array}{c} w_0 \\ w \end{array}\right) \in W
\]

\[
\iff \quad D^T_i \lambda \geq A_i(x)^T w \quad \forall \left(\begin{array}{c} w_0 \\ w \end{array}\right) \in W, \quad i = 1, \ldots, L
\]

\[
\iff \quad -D^T_i u + \left(\begin{array}{c} w_0 \\ w \end{array}\right)^T \begin{pmatrix} -D^T_i r \\ A_i(x) - V^T D_i \end{pmatrix} \leq 0 \quad \forall \left(\begin{array}{c} w_0 \\ w \end{array}\right) \in W, \quad i = 1, \ldots, L
\]

\[
\iff \quad -D^T_i u + \delta^* \left(\begin{pmatrix} -D^T_i r \\ A_i(x) - V^T D_i \end{pmatrix} \left| W \right\right) \leq 0 \quad \forall \left(\begin{array}{c} w_0 \\ w \end{array}\right) \in W, \quad i = 1, \ldots, L
\]

\[
\iff \quad \begin{cases} 
-D^T_i u + (-D^T_i r) f \left(\frac{A_i(x) - V^T D_i}{D^T_i r} \right) \leq 0 \\
-D^T_i r > 0 
\end{cases} \quad \forall \left(\begin{array}{c} w_0 \\ w \end{array}\right) \in W, \quad i = 1, \ldots, L, \quad (16)
\]

which is exactly (10b). Because the decision rule (14) is a more general decision rule as the decision rule used in Theorem 2, it follows that (10) is a tighter safe approximation than (8).

The resulting system of inequalities (10) is convex and includes $2q + qp + n$ variables compared to the original $n$. We note that the nonnegativity constraints introduced due to the domain of the perspective function relate to the initial uncertainty set $U$. In fact, by appropriately applying Farkas’ Lemma, we can show that if the initial uncertainty set $U$ is non-empty, we know that for any $r$ such that $-1 < d^T r < 0$, it holds that $-D^T_i r > 0$.

## 3 Sum-of-Max Constraints

In this section we consider $f : \mathbb{R}^p \mapsto \mathbb{R}$ to be given by

\[
f(v) = \sum_{i=1}^{m} \max_{k \in K_i} \{v_k\}, \quad (17)
\]

such that (1) is given by

\[
\sum_{i=1}^{m} \max_{k \in K_i} \{A_k(x)\zeta + b_k(x)\} \leq 0 \quad \forall \zeta \in U, \quad (18)
\]

where $K_i \subseteq \{1, \ldots, p\}$, for $i = 1, \ldots, m$ and $A_k(x)$ is the $k$-th row of $A(x)$. In order to comment further on the results from Section 2 for sum-of-max functions, finding the
conjugate and its domain is essential. For this, we find

\[ f^*(w) = \sup_{v \in \mathbb{R}^p} \left\{ w^T v - \sum_{i=1}^{m} \max_{k \in K_i} \{v_k\} \right\} = \begin{cases} 0 & \text{if } w \geq 0, \sum_{k \in K_i} w_k = 1 \ i = 1, \ldots, m \ \ +\infty & \text{otherwise,} \end{cases} \]

and thus

\[ \text{dom } f^* = \left\{ w \in \mathbb{R}^p \mid \sum_{k \in K_i} w_k = 1, \ i = 1, \ldots, m \right\}. \]

Note that this is a simplex when \( m = 1 \) and the cartesian product of simplices when \( K_i \cap K_j = \emptyset \) for all \( i \neq j \). We note that this means the approximation is exact when \( m = 1 \). We moreover can assume \( K_i \cap K_j = \emptyset \) for all \( i \neq j \) without loss of generality and will do so in the remainder of this section.

Traditionally, the robust counterpart of a sum of max constraint can be reformulated by introducing auxiliary adjustable variables \( y_i \) to reformulate (18) (Gorissen and Den Hertog, 2013; Ardestani-Jaafari and Delage, 2016):

\[
\begin{align*}
\forall \zeta \in U, \ \exists y \in \mathbb{R}^m : \ \\
&\sum_{i=1}^{m} y_i \leq 0 \\
&A_k(x)\zeta + b_k(x) \leq y_i \ \forall k \in K_i, \ i = 1, \ldots, m. 
\end{align*}
\]

This approach is in fact equivalent to our approach when employing linear decision rules for both (8) and (19). For details on proving this we refer to Appendix A. Ardestani-Jaafari and Delage (2016) show that for box and budget uncertainty sets, linear decision rules are optimal in solving (19) under some additional assumptions regarding the structure of \( A(x) \). By the equivalence to our approach for linear decision rules, this means that linear decision rules are also optimal for box and budget uncertainty sets under these additional assumptions in our approach.

For polyhedral uncertainty sets that are not a box or a budget uncertainty set or when the additional assumptions made by Ardestani-Jaafari and Delage (2016) are not satisfied, linear decision rules are not necessarily optimal. In this case, using the approach we suggest can be beneficial as it allows for other techniques from adjustable robust optimization to be used.

When considering sum-of-max constraints, we can in fact also apply our approach for ellipsoidal uncertainty. To accomplish this, we use the fact that \( f^*(w) = 0 \) on its domain and that this domain is a simplex, and thus a polyhedron. This means we can apply Theorem 1 twice to find an equivalent linear ARO problem. It turns out that, in fact, this linear ARO problem is (19) with the original ellipsoidal uncertainty set. The full derivation can be found in Appendix B.
4 Geometric Programming

In this section we consider geometric programming problems and approximations to their robust counterparts. In general, a geometric programming constraint is given by (Boyd et al., 2007):

\[ h(Cx + c) \leq 0, \tag{20} \]

for some \( C \in \mathbb{R}^{p \times n}, \, c \in \mathbb{R}^p \), where \( h \) is the log-sum-exp function given by

\[ h(z) = \log (e^{z_1} + \cdots + e^{z_p}). \]

In this paper, we consider uncertainty in \( C \). We note that \( h \) is a convex function in both the uncertain parameters \( C \) and the decision variables \( x \). As the argument in (20) is affine in both \( C \) and \( x \) we note that there exist affine mappings \( A : \mathbb{R}^n \mapsto \mathbb{R}^{p \times L} \) and \( b : \mathbb{R}^n \mapsto \mathbb{R}^p \) such that the robust counterpart of (20) is given by

\[ h(A(x)\zeta + b(x)) \leq 0 \quad \forall \zeta \in U, \tag{21} \]

where \( \zeta \) contains all elements of \( C \) and thus \( L = p \cdot n \) and

\[ U = \{ \zeta \in \mathbb{R}_+^L : D\zeta = d \}. \]

Applying the results from Theorem 3 then yields that

\[
\begin{cases}
    d^T u + (1 + d^T r) h \left( \frac{V^T d + b(x)}{1 + d^T r} \right) \leq 0 \\
    1 + d^T r > 0 \\
    -D_i^T u + (-D_i^T r) h \left( \frac{A(x) - V^T D_i}{-D_i^T r} \right) \leq 0 \\
    -D_i^T r > 0 \quad i = 1, \ldots, L, \tag{22a}
\end{cases}
\]

is a safe approximation of (21).

We note that the resulting safe approximation involves the perspective of the log-sum-exp function. We know from (Serrano, 2015) that a constraint modeled by \( h \) can be represented as an exponential cone, and problems including such constraints can thus be solved efficiently. Appendix C shows that any perspective of a conically representable function is conically representable with the same cone, and thus (22) can be solved efficiently in theory as it can be represented using exponential cones.

In practice, however, interior point methods can have trouble solving problems such as (22) because they involve perspective functions for which gradient computations can be notoriously difficult when the denominator is close to zero. Alternatively, we can use the safe approximation as formulated in Theorem 2. To this end we note that the conjugate
of $h$ is given by (Ben-Tal et al., 2015):

$$
\begin{align*}
    h^*(w) &= \sup_{z \in \mathbb{R}^p} \left\{ w^\top z - \log \left( \sum_{i=1}^{p} e^{z_i} \right) \right\} \\
    &= \begin{cases} 
        \sum_{i=1}^{p} w_i \log w_i & \text{if } w_i \geq 0 \ \forall i \ \text{and} \ \sum_{i=1}^{p} w_i = 1 \\
        +\infty & \text{otherwise,}
    \end{cases}
\end{align*}
$$

and thus

$$
    \text{dom } h^* = \left\{ w \in \mathbb{R}_+^p \middle| \sum_{i=1}^{p} w_i = 1 \right\},
$$

which is a simplex. From Section 3 we know that the support function of a simplex is the maximum function and thus we find that

$$
\begin{align*}
    \begin{cases} 
        d^\top u + h \left( V^\top d + b(x) \right) \leq 0 \\
        \max_k \left\{ A_{ki}(x) - V_k^\top D_i \right\} \leq D_i^\top u & i = 1, \ldots, L,
    \end{cases}
\end{align*}
$$

(23a) (23b)

is a safe approximation of (21), where $A_{ki}(x)$ is the element on the $k$-th row and $i$-th column of $A(x)$ and $V_k$ is the $k$-th column of $V$. We note that (23b) can easily be reformulated as linear constraints.

## 5 Numerical Results

We test our approach on several randomly generated geometric programming instances. In particular, we use identically structured instances to Hsiung et al. (2008). In particular, this means we treat geometric programming problems with a linear objective, and a number of two-term log-sum-exp inequality constraints with uncertainty. In particular we aim to solve the problem

$$
\begin{align*}
    \min_x & \quad c^\top x \\
    \text{s.t.} & \quad h \left( \begin{pmatrix} -1 + B_i^{(1)} \zeta & x \\ -1 + B_i^{(2)} \zeta & x \end{pmatrix} \right) \leq 0 & \forall \zeta \in U, \ i = 1 \ldots, m,
\end{align*}
$$

(24)

where $c = 1 \in \mathbb{R}^n$ is the all ones vector, and $B_i^{(1)}, B_i^{(2)} \in \mathbb{R}^{n \times L}$ are randomly generated sparse matrices with sparsity density 0.1 whose nonzero elements are uniformly distributed on the interval $[0, 1]$. The uncertainty set $U$ is assumed to be a box, that is,

$$
U = \left\{ \zeta \in \mathbb{R}^L \middle| \|\zeta\|_\infty \leq 1 \right\}.
$$

Note that since $U$ is symmetric around 0, this is equivalent to having the elements of $B_i^{(1)}, B_i^{(2)}$ being uniformly distributed on $[-1, 1]$. 


Approximation Error (%) | Computation Time (s)
--- | ---
Lower bound from (23) | -0.01 | 4.1
Lower bound from (22) | -0.00 | 17.4
Exact Robust (22) | 0 | 17.9
(23) | 0.08 | 128.9
(23) | 1.44 | 7.7

Table 1: Average solution quality and computation time for 20 randomly generated instances of type (24) with $n = m = 100$ and $L = 5$.

We first consider a set of 20 small examples with $n = m = 100$ and $L = 5$. Since $L$ is small, the exact robust solution can be found by enumerating the $2^L$ vertices of $U$. For larger $L$, however, we will need to resort to comparing our solutions’ objective value to a lower bound. To this end, we use a lower bound based on the work of Hadjiyiannis et al. (2011) and Zhen et al. (2017) that uses the optimal solution to a safe approximation to find potentially critical scenarios in the uncertainty set. For more details we refer the reader to Appendix D.

To evaluate the quality of this lower bound, we have included this lower bound on the exact robust objective value based on the solution of both (22) and (23) in Table 1 as well as the approximation error compared to the exact robust solution and computation time of both (22) and (23). We define this approximation error (in percentage) equally to (Hsiung et al., 2008):

$$100 \left( \frac{e^{c^T \hat{x}}}{e^{c^T x^*}} - 1 \right),$$

where $\hat{x}$ is the solution to our safe approximation and $x^*$ is the exact robust solution. In other words, we compare the objective value of the robust geometric programming problem in posynomial form.

Clearly, for instances of this size the lower bound is particularly good. Moreover, it is an order of magnitude closer to the exact robust objective value compared to the solutions we find using our safe approximation. Therefore, we expect that using the lower bound instead of the exact robust solution for larger instances has hardly any effect on the values we report. We note that Hsiung et al. (2008) report approximation errors between 30% and 0.1% dependent on the quality of approximation used, for $L = 5$ and $n = m = 500$.

To analyze how our approach scales with more uncertain parameters, Figure 1 shows the average difference to the lower bound and computation time of both (22) and (23) for several values of $L$ over 20 random instances. We note that already for $L = 10$, (22) is computationally cumbersome to solve as can be seen by the starkly increasing amount
Figure 1: Average results of solving (22) in blue squares, (23) in green triangles and the exact robust solution in red circles over twenty randomly generated instances.

of time it takes. Moreover, we were not able to solve (22) to optimality for three of the twenty random instances generated. The results for those instances have therefore been excluded from the results in Figure 1.

For any number of uncertain parameters higher than 7, we run into memory constraints when trying to find the exact robust solution and these solutions have therefore not been included in Figure 1. Based on the computation time for $L = 5, 6, 7$ we predict solving (22) is faster than finding the exact robust solution for any $L \geq 10$.

The solutions to the geometric programming problems have been obtained using Julia and the IPOPT solver. The experiments were conducted on a desktop with 8 GB RAM and a 3.4 GHz Intel Core i7 processor.

**Acknowledgments**

We thank Erick Delage from HEC Montréal for sharing the proof to Theorem 1, which is more concise than our original proof. The research of the first author was funded by the Netherlands Organisation for Scientific Research (NWO) Research Talent [Grant 406.17.511].
A Equivalence of Sum-of-Max Reformulations

Substituting linear decision rules of the form \( y_i(\zeta) = u + V\zeta \) in (19), yields the following robust counterpart:

\[
\begin{align*}
&d^T z_0 + \sum_{i=1}^{m} u_i \leq 0 & \text{(25a)} \\
&D^T z_0 \geq \sum_{i=1}^{m} V_i^T & \text{(25b)} \\
&D^T z_{ik} - u_i + b_k(x) \leq 0 \quad \forall k \in K_i, \ i = 1, \ldots, m & \text{(25c)} \\
&D^T z_{ik} \geq A_k(x)^T - V_i^T \quad \forall k \in K_i, \ i = 1, \ldots, m & \text{(25d)} \\
&z_{ik} \geq 0 \quad \forall k \in K_i, \ i = 1, \ldots, m & \text{(25e)} \\
&z_0 \geq 0. & \text{(25f)}
\end{align*}
\]

Here, \( z_0 \in \mathbb{R}_+^q \) is the dual variable corresponding to the robust counterpart of (19a) and \( z_{ik} \in \mathbb{R}_+^q \) are the dual variables corresponding to the robust counterpart of (19b). Moreover, \( A_k(x) \) is the \( k \)-th row of \( A(x) \) and \( V_i \) is the \( i \)-th row of \( V \) in this notation.

If we continue, on the other hand, from (8), with \( f \) defined by (17), and introducing auxiliary variables \( z_{ij} \) for \( i = 1, \ldots, m, \ j = 0, \ldots, L \), to model the sum of max, we obtain the following formulation:

\[
\begin{align*}
&d^T u + \sum_{i=1}^{m} z_{i0} \leq 0 & \text{(26a)} \\
&D^T u \geq \sum_{i=1}^{m} z_i & \text{(26b)} \\
&D^T V_k - z_{i0} + b_k(x) \leq 0 \quad \forall k \in K_i, \ i = 1, \ldots, m & \text{(26c)} \\
&D^T V_k \geq A_k(x)^T - z_i \quad \forall k \in K_i, \ i = 1, \ldots, m. & \text{(26d)}
\end{align*}
\]

Here, \( z_i = \begin{bmatrix} z_{i1} & \ldots & z_{iL} \end{bmatrix}^T \) and \( A_k(x) \) is the \( k \)-th row of \( A(x) \), while \( V_k \) is the \( k \)-th column of \( V \).

Although the interpretation and naming of variables differs in the safe approximations (25) and (26), there is only one true difference. In (25) \( z_{ik} \in \mathbb{R}^q \) for all \( k \in K_i, \ i = 1, \ldots, m \), while the corresponding variable \( V \in \mathbb{R}^{q \times p} \) does not necessarily have the same dimension. The problems are thus only equivalent when all \( K_i \) are pair-wise disjoint, such that \( \sum_{i=1}^{m} |K_i| = p \).

We can, without loss of generality, formulate any sum-of-max constraint such that this holds, by appropriately defining \( A(x) \) and \( b(x) \). This implies that the two approaches are in fact equivalent.
B Ellipsoidal Uncertainty

Theorem 4. Suppose $U$ is a ball uncertainty set, i.e.,

$$U = \{ \zeta \in \mathbb{R}^L \mid \| \zeta \|_2 \leq \rho \}.$$

Then $x \in \mathbb{R}^n$ satisfies (18) if and only if it satisfies

$$\forall \tilde{w} \in U, \exists \lambda \in \mathbb{R}^m : \begin{cases} \mathbf{1}^\top \lambda \leq 0 \quad (27a) \\ \tilde{D}^\top \lambda \geq b(x) + A(x)\tilde{w}, \quad (27b) \end{cases}$$

with $\tilde{D} \in \mathbb{R}^{m \times p}$ given by

$$D_{ik} = \begin{cases} 1 & \text{if } k \in K_i \\ 0 & \text{otherwise.} \end{cases}$$

Proof. We first follow the reasoning of the proof of Theorem 1 to find that (18) is equivalent to

$$\sup_{\zeta \in U} \{ f(A(x)\zeta + b(x)) \} \leq 0$$

$$\iff \sup_{w \in \text{dom } f^*} \left\{ \sup_{\zeta \in U} \left\{ \left( A(x)^\top w \right)^\top \zeta + b(x)^\top w - f^*(w) \right\} \right\} \leq 0$$

$$\iff \sup_{w \in \text{dom } f^*} \left\{ \rho \| A(x)^\top w \|_2 + b(x)^\top w - f^*(w) \right\} \leq 0. \quad (28)$$

Subsequently we define

$$W = \text{dom } f^* = \left\{ w \in \mathbb{R}^p_+ \mid \sum_{k \in K_i} w_k = 1 \quad i = 1, \ldots, m \right\} = \left\{ w \in \mathbb{R}^p_+ \mid \tilde{D}w = \tilde{d} \right\},$$

where $\tilde{d} = 1 \in \mathbb{R}^m$ and $\tilde{D} \in \mathbb{R}^{m \times p}$ is defined by

$$D_{ik} = \begin{cases} 1 & \text{if } k \in K_i \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, we define $\tilde{f} : \mathbb{R}^{L+1} \mapsto \mathbb{R}$ and $\tilde{A} : \mathbb{R}^n \mapsto \mathbb{R}^{(L+1) \times p}$ by

$$\tilde{f} \left( \begin{pmatrix} z_0 \\ z \end{pmatrix} \right) = z_0 + \rho \| z \|_2, \quad \tilde{A}(x) = \begin{bmatrix} b(x)^\top \\ A(x)^\top \end{bmatrix},$$

such that (28) can be written as

$$\sup_{w \in W} \left\{ \tilde{f} \left( \tilde{A}(x)w \right) \right\} \leq 0, \quad (29)$$

as $f^*(w) = 0$ for all $w \in \text{dom } f^*$. Since $\tilde{f}$ is convex, $\tilde{A}$ is affine and $W$ is polyhedral, we can apply Theorem 1 to (29) and find (18) is equivalent to

$$\forall \begin{pmatrix} \tilde{w}_0 \\ \tilde{w} \end{pmatrix} \in \text{dom } \tilde{f}^*, \exists \lambda \in \mathbb{R}^m : \begin{cases} \mathbf{1}^\top \lambda - \tilde{f}^* \left( \begin{pmatrix} \tilde{w}_0 \\ \tilde{w} \end{pmatrix} \right) \leq 0 \quad (30a) \\ \tilde{D}^\top \lambda \geq \tilde{w}_0 b(x) + A(x)\tilde{w}. \quad (30b) \end{cases}$$
In analyzing $\tilde{f}^*$ we find:

$$\tilde{f}^* \left( \left( \tilde{w}_0 \right) \right) = \sup_{z_0, z} \left\{ z_0 \tilde{w}_0 + z^\top w - \tilde{f} \left( \begin{pmatrix} z_0 \\ z \end{pmatrix} \right) \right\}$$

$$= \sup_{z_0, z} \left\{ z_0 (\tilde{w}_0 - 1) + z^\top w - \rho \|z\|_2 \right\}$$

$$= \sup_{z_0} \left\{ z_0 (\tilde{w}_0 - 1) \right\} + \sup_z \left\{ z^\top w - \rho \|z\|_2 \right\}$$

$$= \begin{cases} 0 & \text{if } \tilde{w}_0 = 1, \|w\|_2 \leq \rho \\ +\infty & \text{otherwise,} \end{cases}$$

and thus

$$\text{dom } \tilde{f}^* = \left\{ \left( \tilde{w}_0 \right) \in \mathbb{R}^{L+1} \mid \tilde{w}_0 = 1, \|w\|_2 \leq \rho \right\}.$$
Clearly, \( S_{\text{per}} \) and \( T_{\text{per}} \) are affine mappings. Moreover we find

\[
\text{Epi } f^{\text{per}} = \left\{ (x, v, t) \mid vf\left(\frac{x}{v}\right) \leq t \right\} \\
= \left\{ (x, v, t) \mid \left(\frac{x}{v}, \frac{t}{v}\right) \in \text{Epi } f \right\} \\
= \left\{ (x, v, t) \mid \exists u \in \mathbb{R}^m, S\left(\frac{x}{v}, u, \frac{t}{v}\right) = 0, \ T\left(\frac{x}{v}, u, \frac{t}{v}\right) \in K \right\} \\
= \left\{ (x, v, t) \mid \exists u \in \mathbb{R}^m, \ S_{\text{per}}(x, u, t, v) = 0, \ T_{\text{per}}(x, u, t, v) \in K \right\} ,
\]

which concludes the proof.

\[\square\]

## D Progressive Approximation

As all sets of constraints described in Section 2.2 are safe approximations to our original constraint (1), they yield suboptimal solutions. In particular, we propose linear decision rules to solve (4), which is equivalent to (1), of which we know they generally do not solve adjustable robust optimization problems to optimality (Ben-Tal et al., 2004). Moreover, as our adjustable formulation (4) exhibits left-hand side uncertainty, that is, the uncertain parameter \( w \) directly interacts with decision variables \( x \), little is known with regard to the approximative power of linear decision rules.

In this section, therefore, we focus on finding a good progressive approximation to (1) such that we can gauge the quality of the conservative approximations we propose. A simple method detailed by Zhen et al. (2017) to obtain such approximation is to only require (1) to hold for a finite subset of scenarios from the uncertainty set \( U \). The approximation is then given by

\[
f\left(A(x)\zeta(1) + b(x)\right) \leq 0 \quad k = 1, \ldots, K,
\]

where \( \{\zeta(1), \ldots, \zeta(K)\} \subseteq U \). We note that these constraints are exactly as computationally tractable as the original constraint without uncertainty. In fact, because we assume a polyhedral set \( U \), (32) is equivalent to (1) if \( \{\zeta(1), \ldots, \zeta(K)\} \) contains all extreme points of \( U \). Generally, \( U \) has prohibitively many extreme points though, and we must resort to some other way of finding scenarios \( \zeta(1), \ldots, \zeta(K) \).

We can apply the same reasoning as above to (4) to find an approximation:

For \( k = 1, \ldots, K \), \( \exists \lambda^{(k)} \in \mathbb{R}^q : \)

\[
\begin{align*}
\begin{cases}
\left(w_0^{(k)} + b(x)^\top w^{(k)} + d^\top \lambda^{(k)}\right) \leq 0 \\
D^\top \lambda^{(k)} \geq A(x)^\top w^{(k)},
\end{cases}
\end{align*}
\]

where \( \left\{\left(w_0^{(1)}, \ldots, w_0^{(K)}\right), \ldots, \left(w_0^{(1)}, \ldots, w_0^{(K)}\right)\right\} \subseteq W \) and \( \lambda^{(k)} \in \mathbb{R}^q \) is a non-adjustable variable. Recall that

\[
W = \left\{\left(w_0, w\right) : w_0 + f^*(w) \leq 0\right\}.
\]

17
which generally has infinitely many extreme points.

An approach to find a small and efficient set of scenarios for two-stage fixed-recourse robust constraints is suggested by Hadjiyiannis et al. (2011). For any feasible solution $\hat{x}$ and linear decision rule $\lambda = \hat{u} + \hat{V}w + \hat{r}w_0$, we find scenarios that are worst-case for the constraints in (4). We then hope that these scenarios are also worst-case for the actual optimal solution $x^*, \lambda^*$ of (4). For our problem, this means that we obtain scenarios

$$\bar{w} = \arg \max_{\left[w_0 \atop w\right] \in W} \left\{ d^T \left(\hat{u} + \hat{V}w + \hat{r}w_0\right) + b(\hat{x})^T w + w_0\right\},$$

as well as the worst-case scenarios from (4b). An extension proposed by Zhen et al. (2017) is to use these $L + 1$ scenarios to also obtain scenarios $\zeta^{(1)}, \ldots, \zeta^{(L+1)}$ by solving

$$\zeta^{(k)} = \arg \max_{\zeta \in U} \left\{ (A(\hat{x})\zeta + b(\hat{x}))^T \bar{w}^{(k)}\right\}.$$

We note that similarly to this approach, we can also obtain worst-case scenarios $\bar{w}$ based on a linear decision rule solving (8). The resulting approximation of (1) will, however, likely be worse than the one based on (10). For more details, we refer to the papers by Hadjiyiannis et al. (2011) and Zhen et al. (2017).

References


