Robust Geometric Programming is co-NP hard*

André Chassein† and Marc Goerigk

Fachbereich Mathematik, Technische Universität Kaiserslautern, Germany

Abstract

Geometric Programming is a useful tool with a wide range of applications in engineering. As in real-world problems input data is likely to be affected by uncertainty, Hsiung, Kim, and Boyd introduced robust geometric programming to include the uncertainty in the optimization process. They also developed a tractable approximation method to tackle this problem. Further, they pose the question whether there exists a tractable reformulation of their robust geometric programming model instead of only an approximation method. We give a negative answer to this question by showing that robust geometric programming is co-NP hard in its natural posynomial form.

Keywords Robust optimization; Geometric Programming; Complexity; co-NP hardness

1 Introduction

Geometric programming can cover constraints that might be neither linear nor convex and, hence, it can be used to model problems that cannot be formulated by standard optimization tools such as linear or convex programs. Despite the possibly difficult constraints in a geometric program, it can be solved efficiently by transforming it to a convex optimization problem using variable transformation. For a general overview of geometric programming and its applications we refer to the survey of Ecker [4].

In the field of classic optimization one assumes to have perfect knowledge of the input parameters that are used in the description of the problem. For real-world problems this assumption can be violated by, e.g., erroneous measurements or wrong forecasts. A solution that is computed under the assumption of wrong input parameters can yield a bad performance in reality. Hence, researchers have tried to incorporate the aspect that data is afflicted with uncertainty into the model formulation of the optimization problem. Well-known such approaches are due to Ben-Tal and Nemirovski [1], and Bertsimas and

*Effort sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-13-1-3066. The U.S Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright notation thereon.

†Corresponding author. Email: chassein@mathematik.uni-kl.de

Sim [3]. For more information about robust optimization we refer to the surveys of Bertsimas, Brown, and Caramanis [2], and of Goerigk and Schöbel [6].

To introduce uncertainty into the optimization problem, one defines an uncertainty set that describes the possible realizations of the input parameters of the model. The classic approach of robust optimization is to find a solution that is feasible under all possible realizations of the input data.

Hsiung, Kim, and Boyd applied this general method to geometric programming and defined the robust geometric program [7]. They note that the problem complexity is unknown. In this note we study the computational complexity of the robust geometric programming reformulation if it is provided in its natural posynomial form, thus providing an answer to this open question. Using a reduction from the complement of the partition problem, we show that geometric programming is co-NP hard.

2 Robust Geometric Programming

A robust geometric program in convex form is defined as

$$\begin{array}{lll} \text{minimize} & c^T y & \\ \text{subject to} & \operatorname{lse}(A_i y + b_i) \leq 0 & i = 1, \ldots, m \\ & Gy + h = 0 & \end{array}$$

where $A_i \in \mathbb{R}^{K^i \times n}$, $b_i \in \mathbb{R}^{K^i}$ for $i = 1, \ldots, m$, $c \in \mathbb{R}^n$, $G \in \mathbb{R}^{\ell \times n}$, and $h \in \mathbb{R}^\ell$ denote the input parameters, and $y \in \mathbb{R}^n$ is the vector of decision variables. The convex *log-sum-exp* function lse: $\mathbb{R}^k \to \mathbb{R}$ is defined as

$$\operatorname{lse}(z_1, \ldots, z_k) = \log(e^{z_1} + \ldots + e^{z_k}).$$

In robust optimization it is assumed that the input parameter of the problem are not given exactly. A common approach to allow variability in the input data is to parameterize it with respect to some vector $u \in \mathbb{R}^L$ that belongs to a fixed uncertainty set $\mathcal{U}$. The same method is used in [7]. The authors assume that the problem data (A_i, b_i) depends affinely on the vector of uncertain parameters u.

$$(\tilde{A}_i(u), \tilde{b}_i(u)) = \left(A_i^0 + \sum_{j=1}^{L} u_j A_i^j,\ b_i^0 + \sum_{j=1}^{L} u_j b_i^j \right),\ u \in \mathcal{U} \subset \mathbb{R}^L.$$

The robust geometric program in convex form ($RGP-con$) is then given by

$$\begin{array}{lll} \text{minimize} & c^T y & \\ \text{subject to} & \sup\limits_{u \in \mathcal{U}} \operatorname{lse}(\tilde{A}_i(u) y + \tilde{b}_i(u)) \leq 0 & i = 1, ..., m \\ & Gy + h = 0 & \end{array}$$

The authors of [7] suggest two different uncertainty sets: A polyhedral uncertainty set $\mathcal{U} = \{u \in \mathbb{R}^L \mid Du \leq d\}$, where $d \in \mathbb{R}^K$ and $D \in \mathbb{R}^{K \times L}$, and an ellipsoidal uncertainty set $\mathcal{U} = \{\overline{u} + P\rho \mid ||\rho||_2 \leq 1, \rho \in \mathbb{R}^L\}$, where $\bar{u} \in \mathbb{R}^L$ and $P \in \mathbb{R}^{L \times L}$. In this paper, we consider polyhedral uncertainty sets. The authors note that it is an open question whether ($RGP-con$) has a tractable reformulation.

Note that geometric programs appear in practice in their posynomial and not in their convex form. Hence, we discuss tractability issues for problems in their posynomial form. The difficulty with the convex form is that a problem given in posynomial form with rational input data leads in general to a convex problem with irrational data. An exact description in convex form with bounded input size is, therefore, impossible.

We first transform the problem to a robust geometric program in posynomial form. This can be done in two steps. First by replacing each variable $y_l = log(x_l)$ and then taking the exponential function on both sides of every (in)equality and on the objective function.

An affine equality constraint

$$G_j^T y + h_j = 0$$

with G_j being the j^{th} row of G is transformed to a monomial equality constraint

$$f_j(x) = e^{h_j} \prod_{l=1}^{n} x_l^{G_{jl}} = 1.$$

Note that the objective function is also transformed to a monomial function $f_0(x)$. An inequality constraint of the form

$$\text{lse}(\tilde{A}_i(u)y + \tilde{b}_i(u)) \leq 0$$

is transformed to a posynomial inequality constraint with parameter u

$$f_i(u, x) = \sum_{k=1}^{K^i} \left(e^{\tilde{b}_i^k(u)} \prod_{l=1}^{n} x_l^{\tilde{A}_{il}^k(u)} \right) \leq 1.$$

The robust geometric program in posynomial form ($RGP - pos$) has, therefore, the form

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & \sup_{u \in \mathcal{U}} f_i(u, x) \leq 1 & i = 1, \ldots, m \\ & f_j(x) = 1 & j = 1, \ldots, p \end{array}$$

3 Complexity Result

In this section we show that the decision version of robust geometric programming in posynomial form ($Dec - RGP - pos$) is co-NP hard.

Definition 3.1. ($Dec - RGP - pos$)
Given a constant k. Does there exists a solution $x \in \mathbb{R}^n$ such that $f_0(x) \leq k$, $sup_{u \in \mathcal{U}} f_i(u, x) \leq 1 \ \forall i = 1, \ldots, m$, and $f_j(x) = 1 \ \forall j = 1, \ldots, p$?

For the reduction we use the complement of the partition problem ($co - PART$), which is known to be a co-NP complete problem, as partition is a NP complete problem [5].

Definition 3.2. ($co-PART$)
Given N numbers $a_1,...,a_N \in \mathbb{N}$. Does it hold for all subsets $S \subset \{1,...,N\}$ of these N items that $\sum_{i\in S} a_i \neq \sum_{j\notin S} a_j$?

Theorem 3.3. *($Dec-RGP-pos$) is co-NP hard.*

Proof. To show that the decision version of ($RGP-pos$) is co-NP hard, we have to find a reduction from another co-NP hard problem. This means that we have to construct for every instance Π_{co-P} of ($co-PART$) an instance Π_{RGP} of ($Dec-RGP-pos$) in polynomial time (with respect to the size of Π_{co-P}) such that Π_{RGP} is a yes−instance if and only if Π_{co-P} is a yes−instance.

Given an instance Π_{co-P} with N natural numbers $a_1,...,a_N \in \mathbb{N}$, we define the following robust geometric program in posynomial form:

$$\begin{aligned}
\text{minimize} \quad & 0 \\
\text{subject to} \quad & \sup_{u\in\mathcal{U}} \left(\sum_{l=1}^{N} x^{u_l} + x^{-u_l} \right) \leq N\left(2+\frac{1}{2}\right) - \frac{1}{2N(a_{max}+1)} \\
& x = 2
\end{aligned}$$

where $\mathcal{U} = \{u \in \mathbb{R}^N \mid -1 \leq u_l \leq 1 \; \forall\, l = 1,\ldots,N \wedge \sum_{l=1}^{N} u_l a_l = 0\}$ and $a_{max} = \max_{l=1,...,N} a_l$. We complete the description of Π_{RGP} by setting $k = 0$. Note that this problem is indeed an instance of ($Dec-RGP-pos$), and the uncertainty set is a polyhedron. It is straightforward to check that the definition of Π_{RGP} can be done in polynomial time (with respect to the size of Π_{co-P}). We need to show that Π_{RGP} is a yes−instance if and only if Π_{co-P} is a yes−instance.

We start with the easier direction. Assume that Π_{co-P} is a no−instance. Then we have to show that Π_{RGP} is also a no−instance. If Π_{co-P} is a no−instance there must exist a subset $S \subset \{1,\ldots,N\}$ such that $\sum_{i\in S} a_i = \sum_{j\notin S} a_j$. Define the vector $\hat{u}$ as follows, $\hat{u}_l = 1$ if $l \in S$ and $\hat{u}_l = -1$ otherwise $\forall\, l = 1,\ldots,N$. By construction, $\hat{u} \in \mathcal{U}$. Therefore, we get

$$\begin{aligned}
\sup_{u\in\mathcal{U}} \left(\sum_{l=1}^{N} x^{u_l} + x^{-u_l} \right) &= \sup_{u\in\mathcal{U}} \left(\sum_{l=1}^{N} 2^{u_l} + 2^{-u_l} \right) \\
&\geq \sum_{l=1}^{N} \left(2^{\hat{u}_l} + 2^{-\hat{u}_l}\right) \\
&= \sum_{l=1}^{N} \left(2^{1} + 2^{-1}\right) \\
&= N\left(2+\frac{1}{2}\right) \\
&> N\left(2+\frac{1}{2}\right) - \frac{1}{2N(a_{max}+1)}
\end{aligned}$$

Hence, the geometric program is infeasible and Π_{RGP} is clearly a no−instance.

It is left to show that from Π_{RGP} being a no−instance follows that Π_{co-P} is also a no−instance. Assume that Π_{RGP} is a no−instance. This means that the robust geometric program is infeasible. Hence, there must exists a $\tilde{u} \in \mathcal{U}$ such that

$$\sum_{l=1}^{N} \left(2^{\tilde{u}_l} + 2^{-\tilde{u}_l}\right) > N\left(2 + \frac{1}{2}\right) - \frac{1}{2N(a_{max}+1)}.$$

Assume that there exists an index l^* such that $|\tilde{u}_{l^*}| < 1 - \frac{1}{N(a_{max}+1)}$. Note that the function $s : [-1,1] \to \mathbb{R}$, $s(\alpha) = 2^{\alpha} + 2^{-\alpha}$ is maximal for $\alpha \in \{-1,1\}$ and that $s(\alpha) \leq 2 + 0.5|\alpha|\ \forall\ \alpha \in [-1,1]$. Therefore, we get a contradiction as

$$\begin{aligned}
\sum_{l=1}^{N} \left(2^{\tilde{u}_l} + 2^{-\tilde{u}_l}\right) &= \sum_{l \neq l^*} \left(2^{\tilde{u}_l} + 2^{-\tilde{u}_l}\right) + 2^{\tilde{u}_{l^*}} + 2^{-\tilde{u}_{l^*}} \\
&\leq (N-1)\left(2 + \frac{1}{2}\right) + 2^{\tilde{u}_{l^*}} + 2^{-\tilde{u}_{l^*}} \\
&\leq (N-1)\left(2 + \frac{1}{2}\right) + 2 + \frac{1}{2}|\tilde{u}_{l^*}| \\
&< (N-1)\left(2 + \frac{1}{2}\right) + 2 + \frac{1}{2}\left(1 - \frac{1}{N(a_{max}+1)}\right) \\
&= N\left(2 + \frac{1}{2}\right) - \frac{1}{2N(a_{max}+1)}
\end{aligned}$$

Hence, we know that $|\tilde{u}_l| \geq 1 - \frac{1}{N(a_{max}+1)}\ \forall\ l = 1, \ldots, N$. Next we define the vector $\hat{u} = sgn(\tilde{u})$, where sgn is the multidimensional extension of the signum function, as well as $\delta = \hat{u} - \tilde{u}$. Note that $|\delta_l| \leq \frac{1}{N(a_{max}+1)}\ \forall\ l = 1, \ldots, N$.

$$\begin{aligned}
\left|\sum_{l=1}^{N} \hat{u}_l a_l\right| &= \left|\sum_{l=1}^{N} (\tilde{u}_l + \delta_l) a_l\right| \\
&\leq \left|\sum_{l=1}^{N} \tilde{u}_l a_l\right| + \left|\sum_{l=1}^{N} \delta_l a_l\right| \\
&\leq \sum_{l=1}^{N} |\delta_l| a_l \qquad (\text{as } \tilde{u} \in \mathcal{U}) \\
&\leq \sum_{l=1}^{N} \frac{a_l}{N(a_{max}+1)} < 1
\end{aligned}$$

As $\hat{u} \in \mathbb{Z}^N$ and $a_l \in \mathbb{N}\ \forall l = 1, \ldots, N$, we know that $\sum_{l=1}^{N} \hat{u}_l a_l \in \mathbb{Z}$. Therefore, we can conclude that $\sum_{l=1}^{N} \hat{u}_l a_l = 0$. This gives a partition $S = \{l \mid \hat{u}_l = 1\}$ with $\sum_{i \in S} a_i = \sum_{j \notin S} a_j$. Hence, Π_{co-P} is a no$-$instance. This finishes the proof. □

Note that we used only a sinlge posynomial constraint for the reduction in the proof of Theorem 3.3. Additionaly, only the exponents of the posynomial constraint where affected by uncertainty.

Remark 3.4. *(Dec−RGP−pos) is co-NP hard even if only a single posynomial constraint is part of the problem and if all coefficients of the posynomial are certain.*

References

[1] Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[2] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[3] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

[4] Joseph G. Ecker. Geometric programming: Methods, computations and applications. *SIAM review*, 22(3):338–362, 1980.

[5] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[6] Marc Goerigk and Anita Schöbel. Algorithm engineering in robust optimization. Technical report, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Universität Göttingen, 2013. Submitted.

[7] Kan-Lin Hsiung, Seung-Jean Kim, and Stephen Boyd. Tractable approximate robust geometric programming. *Optimization and Engineering*, 9(2):95–118, 2008.