

# ROBUST DESIGNS VIA GEOMETRIC PROGRAMMING

ALI SAAB\*, EDWARD BURNELL†, AND WARREN W. HOBURG‡

**Abstract.** An approximate formulation of a robust geometric program (RGP) as a convex program is proposed. Interest in using geometric programs (GPs) to model complex engineering systems has been growing, and this has motivated explicitly modeling the uncertainties fundamental to engineering design optimization. RGPs provide a framework for modelling and solving GPs while representing their uncertainties as belonging to an uncertainty set. The RGP methodologies presented here are based on reformulating the GP and then robustifying it with methods from robust linear programming. These new methodologies, along with previous ones from the literature, are used to robustify two aircraft design problems, and the results of these different methodologies are compared and discussed.

**Key words.** Geometric Programming, Posynomials, Robust Optimization, Uncertainty Set, Robust Linear Programming, Safe Approximation, Aircraft, Conic Optimization

**AMS subject classifications.** 90C90, 90C99

**1. Introduction.** A geometric program is a log-convex optimization problem of the form

$$(1) \quad \begin{aligned} & \text{minimize} && f_0(\mathbf{u}) \\ & \text{subject to} && f_i(\mathbf{u}) \leq 1, i = 1, \dots, m_p \\ & && h_i(\mathbf{u}) = 1, i = 1, \dots, m_e \end{aligned}$$

where each  $f_i$  is a *posynomial* and each  $h_i$  is a *monomial*. In its **natural form**, a monomial  $h(\mathbf{u})$  is a function of the form

$$h(\mathbf{u}) = e^b \prod_{j=1}^n u_j^{a_j}$$

where  $\mathbf{a}$  is a row vector in  $\mathbb{R}^n$ ,  $\mathbf{u}$  is a column vector in  $\mathbb{R}_+^n$ , and  $b$  is in  $\mathbb{R}$ . A posynomial  $f(\mathbf{u})$  is the sum of  $K \in \mathbb{Z}^+$  monomials

$$f(\mathbf{u}) = \sum_{k=1}^K e^{b_{kj}} \prod_{j=1}^n u_j^{a_{kj}}$$

where  $\mathbf{a}_k$  is a row vector in  $\mathbb{R}^n$ ,  $\mathbf{u}$  is a column vector in  $\mathbb{R}_+^n$ , and  $b_k$  is in  $\mathbb{R}$  [7]. A logarithmic change of the variables  $x_j = \log(u_j)$  would convert the monomials and posynomials into their **exponential form**. A monomial  $h_i(\mathbf{x})$  is a function of the form

$$h_i(\mathbf{x}) = e^{\mathbf{a}_i \mathbf{x} + b_i}$$

where  $\mathbf{a}_i$  is a row vector in  $\mathbb{R}^n$ ,  $\mathbf{x}$  is a column vector in  $\mathbb{R}^n$ , and  $b_i$  is in  $\mathbb{R}$ . A posynomial  $f_i(\mathbf{x})$  is the sum of  $K_i \in \mathbb{Z}^+$  monomials

$$f_i(\mathbf{x}) = \sum_{k=1}^{K_i} e^{\mathbf{a}_{ik} \mathbf{x} + b_{ik}}$$

where  $\mathbf{a}_{ik}$  is a row vector in  $\mathbb{R}^n$ ,  $\mathbf{x}$  is a column vector in  $\mathbb{R}^n$ , and  $b_{ik}$  is in  $\mathbb{R}$ .

---

\*Department of Computations for Design and Optimization, Massachusetts Institute of Technology, Cambridge, MA (saab@mit.edu).

†Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA (eburn@mit.edu).

‡Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA (whoburg@mit.edu).

Solving large-scale GPs became extremely efficient and more reliable [7] after the development of interior point methods [22] [19], prompting engineers to use GP formulations for the modeling and design optimization of complex engineering systems. Recent GP-compatible applications have been found in fields as varied as digital circuit optimization [8], analog circuit design [12], communication systems [11], chemical process control [26], environmental quality control [14], statistics [21], and antenna and aircraft design [1] [15].

While these GPs can be solved to global optimality for fixed parameter values, many of those values are uncertain during the engineering design process, either because of intrinsic uncertainty (e.g. wind velocity), or unpredictability at that stage of the design process (e.g. the final assembled weight of an airplane’s avionics). Designers, seeking a solution they can trust enough to manufacture, account for these uncertainties by making their models conservative. This conservativeness almost always takes the form of specifying worst-case values and margins for all uncertain constraints and parameters [15]. Such an approach exaggerates the impact of uncertainties, leading to sub-optimality which is undesired. From a designer’s perspective, however, this unwanted conservativeness is their only way to ensure robust results. New techniques that can represent design uncertainties less conservatively are thus able to directly improve engineering solutions and present new design opportunities.

Robust optimization has been used to efficiently solve optimization problems under uncertainty [25]. In contrast to stochastic optimization [6] [24], probability density functions are replaced in robust optimization by uncertainty sets. In the late 1990s, Ben-Tal, Nemirovski, and El Ghaoui proved the tractability of robust linear, quadratic, and second order conic programming problems [3]. Later, Bertsimas et al. derived a tractable approximation of a subset of robust conic optimization problems [4], and discussed the importance and means of choosing an uncertainty set to best describe the problem while preserving tractability [2].

Robust geometric programming is co-NP hard in its natural posynomial form [10], but an interesting tractable formulation for an approximate robust geometric program was presented by Hsiung, Kim, and Boyd [18]. They proposed a bivariate safe approximation for the posynomial constraints, replacing each posynomial constraint with a set of two-term (“bivariate”) posynomial constraints that can be easily approximated using piecewise-linear functions. Although it simplifies the problem into a linear optimization problem, the piecewise-linear approximation of every posynomial leads to a considerable number of additional constraints. Moreover, because it decouples all but two monomials in each posynomial, it yields an effective uncertainty set larger than the specified one and thus more conservative. Our methods seek to improve on [18] by reducing the number of constraints while more accurately approximating the uncertainty set to achieve better (less conservative) solutions.

In this article we use three novel methodologies to construct a tractable approximation of a robust geometric program while drastically reducing the number of constraints compared to [18]. Our first approach deals with each monomial while separating it from its posynomial. Our second approach assumes that only the coefficients  $b$  are uncertain and deals with posynomials as linear constraints. Our third approach accounts for dependency between monomials while letting both the coefficients  $b$  and the exponents  $\mathbf{a}$  be uncertain.

In section 2 of this article, we define robust geometric programming and derive its robust counterparts. Section 3 introduces a simple formulation, our most conservative proposal, which is then refined in Section 4 with the concept of monomial partitioning. Section 5 briefly reviews the piecewise-linearization of two term posynomials in [18],

an essential tool for our work. Section 6 discusses different methodologies for dealing with larger posynomials. In section 7 we apply all methodologies, both new and old, to several previously published GPs for aircraft design and discuss the results. Finally, section 8 states our contributions and conclusions.

**2. Robust Geometric Programming.** The data in a geometric program is usually prone to uncertainties that could either have a probabilistic description or are known to belong to an uncertainty set. Robust geometric programming assumes the latter, and solves the problem for the worst case scenario by finding the best solution that is feasible to all possible realizations from the uncertainty set. This section introduces different forms of GPs, derives intractable RGP formulations, and then states the tractable approximation presented in [18].

**2.1. Other Forms of a Geometric Program.** By replacing each monomial equality constraint  $h(\mathbf{x}) = 1$  by the two monomial inequality constraints  $h(\mathbf{x}) \leq 1$  and  $\frac{1}{h(\mathbf{x})} \leq 1$ , we arrive at the **inequality form** of a GP

$$(2) \quad \begin{aligned} \min \quad & \sum_{k=1}^{K_0} e^{\mathbf{a}_{0k}\mathbf{x} + b_{0k}} \\ \text{s.t.} \quad & \sum_{k=1}^{K_i} e^{\mathbf{a}_{ik}\mathbf{x} + b_{ik}} \leq 1 \quad \forall i \in 1, \dots, m \end{aligned}$$

Without loss of generality, the objective function will be assumed deprived of any data (i.e. replacing it with its epigraph form). Let  $\mathbf{A}$  be the exponents of a GP with components  $\mathbf{a}_{ik}$ , and  $\mathbf{b}$  be the coefficients of a GP with components  $b_{ik}$ .

Also relevant is the **convex form** of a GP, where a logarithm is applied to the objective function and to the constraints.

$$(3) \quad \begin{aligned} \min \quad & \log(f_0(\mathbf{x})) \\ \text{subject to} \quad & \log\left(\sum_{k=1}^{K_i} e^{\mathbf{a}_{ik}\mathbf{x} + b_{ik}}\right) \leq 0 \quad \forall i \in 1, \dots, m \end{aligned}$$

Note that a GP in its convex form is a convex program whose dual is the entropy maximization problem [17].

A final interesting form of a GP is the **categorized form**, which will be introduced later in Section 4, where the constraints of a GP are categorized into different sets according to the number of monomial terms in each posynomial constraint.

**2.2. Robust Counterparts.** As mentioned before, a formulation immune to uncertainty in the system's data should be derived. The data ( $\mathbf{A}$  and  $\mathbf{b}$ ) will be contained in an uncertainty set  $\mathcal{U}$ , where  $\mathcal{U}$  is parameterized affinely by perturbation vector  $\boldsymbol{\zeta}$  as follows

$$(4) \quad \mathcal{U} = \left\{ [\mathbf{A}; \mathbf{b}] = [\mathbf{A}^0; \mathbf{b}^0] + \sum_{l=1}^L \zeta_l [\mathbf{A}^l; \mathbf{b}^l] \right\}$$

where  $\boldsymbol{\zeta}$  belongs to a conic perturbation set  $\mathcal{Z} \in \mathbb{R}^L$  parameterized by  $\mathbf{F}$ ,  $\mathbf{G}$ ,  $\mathbf{h}$  and  $\mathbf{K}$  such that

$$(5) \quad \mathcal{Z} = \left\{ \boldsymbol{\zeta} \in \mathbb{R}^L : \exists \mathbf{u} \in \mathbb{R}^k : \mathbf{F}\boldsymbol{\zeta} + \mathbf{G}\mathbf{u} + \mathbf{h} \in \mathbf{K} \right\}$$

where  $\mathbf{K}$  is a Regular cone in  $\mathbb{R}^N$  with a non-empty interior if it is not polyhedral,  $\mathbf{F} \in \mathbb{R}^{N \times L}$ ,  $\mathbf{G} \in \mathbb{R}^{N \times k}$ , and  $\mathbf{h} \in \mathbb{R}^N$ .

Accordingly, the robust counterpart of the uncertain geometric program in (2) is

$$(6) \quad \begin{aligned} \min \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & \sum_{k=1}^{K_i} e^{\mathbf{a}_{ik}(\boldsymbol{\zeta})\mathbf{x} + b_{ik}(\boldsymbol{\zeta})} \leq 1 \quad \forall i \in 1, \dots, m \quad \forall \boldsymbol{\zeta} \in \mathcal{Z} \end{aligned}$$

$$(7) \quad \begin{array}{ll} \min & f_0(\mathbf{x}) \\ \text{subject to} & \max_{\zeta \in \bar{\mathcal{Z}}} \left\{ \sum_{k=1}^{K_i} e^{\mathbf{a}_{ik}(\zeta)\mathbf{x} + b_{ik}(\zeta)} \right\} \leq 1 \quad \forall i \in 1, \dots, m \end{array}$$

**2.3. Two-Term Formulation.** The current state-of-the-art in approximating RGP is found in [18]. They present an interesting tractable formulation of an approximate robust geometric program which by replacing each posynomial with a set of two-term posynomials arrives at the following formulation

$$\begin{aligned}
(8) \quad & \min f_0(\boldsymbol{x}) \\
& \text{s.t. } \max_{\zeta \in Z} \left\{ e^{\boldsymbol{a}_{i1}(\zeta)\boldsymbol{x} + b_{i1}(\zeta)} + e^{t_1} \right\} \leq 1 \quad \forall i : K_i \geq 3 \\
& \max_{\zeta \in Z} \left\{ e^{\boldsymbol{a}_{ik}(\zeta)\boldsymbol{x} + b_{ik}(\zeta)} + e^{t_k} \right\} \leq e^{t_{k-1}} \quad \forall i : K_i \geq 4 \\
& \hspace{17cm} \forall k \in 2, \dots, K_i - 2 \\
& \max_{\zeta \in Z} \left\{ \sum_{k=K_i-1}^{K_i} e^{\boldsymbol{a}_{ik}(\zeta)\boldsymbol{x} + b_{ik}(\zeta)} \right\} \leq e^{t_{K_i-2}} \quad \forall i : K_i \geq 3 \\
& \max_{\xi \in Z} \left\{ \sum_{k=1}^{K_i} e^{\boldsymbol{a}_{ik}(\xi)\boldsymbol{x} + b_{ik}(\xi)} \right\} \leq 1 \quad \forall i : K_i \leq 2
\end{aligned}$$

As an example, a GP

$$\begin{array}{ll} \min & f \\ \text{s.t.} & \max \{M_1 + M_2 + M_3 + M_4\} \leq 1 \\ & \max \{M_5 + M_6\} \leq 1 \end{array}$$

$$\begin{array}{ll} \min & f \\ \text{s.t.} & \max \{M_1 + e^{t_1}\} \leq 1 \\ & \max \{M_2 + e^{t_2}\} \leq e^{t_1} \\ & \max \{M_3 + M_4\} \leq e^{t_2} \\ & \max \{M_5 + M_6\} \leq 1 \end{array}$$

*This manuscript is for review purposes only.*

The formulation in (8) decouples every monomial in a posynomial except for the last two, yielding a conservative solution. If  $\mathbf{P} = \{i : K_i \geq 3\}$ ,  $\mathbf{N} = \{i : K_i = 2\}$ , and  $\mathbf{M} = \{i : K_i = 1\}$ , and if  $r$  is the number of piecewise-linear terms used to approximate a two-term posynomial, then the number of constraints for the approximate tractable problem in (8) is

$$r \sum_{k \in \mathbf{P}} (K_i - 1) + r|\mathbf{N}| + |\mathbf{M}|$$

**3. Simple Conservative Formulation.** This section introduces our first novel approximation of the robust program in (7). While this formulation is the simplest to implement, it is also the most conservative and completely decouples the monomials of each posynomial.

**DEFINITION 3.1.** A solution  $\mathbf{x}$  to a geometric program is said to be feasible only if  $\mathbf{x}$  satisfies all constraints of that geometric program.

**DEFINITION 3.2.** Let  $\mathcal{C}$  be a constraint on  $\mathbf{x}$ , and let  $\mathcal{S}$  be a set of constraints on  $\mathbf{x}$  and some additional variables  $\mathbf{y}$ .  $\mathcal{S}$  is said to be a safe approximation for  $\mathcal{C}$  if the  $\mathbf{x}$  component of every feasible solution  $(\mathbf{x}, \mathbf{y})$  is  $\mathcal{S}$ -feasible to  $\mathcal{C}$  [3].

For example, if  $f(x) \leq g(x) \forall x$ , then  $g(x) \leq 1$  is a safe approximation of  $f(x) \leq 1$ .

One way to approach the intractability of the dual-form problems in (7) is to replace each constraint with a tractable safe approximation. The fact that

$$\max_{\zeta \in \mathcal{Z}} \sum_{k=1}^{K_i} e^{\mathbf{a}_{ik}(\zeta)\mathbf{x} + b_{ik}(\zeta)} \leq \sum_{k=1}^{K_i} \max_{\zeta \in \mathcal{Z}} e^{\mathbf{a}_{ik}(\zeta)\mathbf{x} + b_{ik}(\zeta)}$$

suggests replacing the intractable constraints by

$$(9) \quad \sum_{k=1}^{K_i} \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathbf{a}_{ik}(\zeta)\mathbf{x} + b_{ik}(\zeta)} \right\} \leq 1 \quad \forall i \in 1, \dots, m$$

From (9), a suggested safe approximation of (7) is

$$(10) \quad \begin{aligned} & \min \quad f_0(\mathbf{x}) \\ & \text{subject to} \quad \sum_{k=1}^{K_i} \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathbf{a}_{ik}(\zeta)\mathbf{x} + b_{ik}(\zeta)} \right\} \leq 1 \quad \forall i \in 1, \dots, m \end{aligned}$$

By adding some dummy variables, the above can be rewritten as

$$(11) \quad \begin{aligned} & \min \quad f_0(\mathbf{x}) \\ & \text{subject to} \quad \sum_{k=1}^{K_i} e^{t_{ik}} \leq 1 \quad \forall i \in 1, \dots, m \\ & \quad \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathbf{a}_{ik}(\zeta)\mathbf{x} + b_{ik}(\zeta)} \right\} \leq e^{t_{ik}} \quad \forall i \in 1, \dots, m \quad \forall k \in 1, \dots, K_i \end{aligned}$$

And finally, in convex form

$$(12) \quad \begin{aligned} & \min \quad \log(f_0(\mathbf{x})) \\ & \text{subject to} \quad \log\left(\sum_{k=1}^{K_i} e^{t_{ik}}\right) \leq 0 \quad \forall i \in 1, \dots, m \\ & \quad \max_{\zeta \in \mathcal{Z}} \left\{ \mathbf{a}_{ik}(\zeta)\mathbf{x} + b_{ik}(\zeta) \right\} \leq t_{ik} \quad \forall i \in 1, \dots, m \quad \forall k \in 1, \dots, K_i \end{aligned}$$

Note that dummy variables need not be introduced for constraints with  $K_i = 1$ .

The convex constraints in (12) are deprived of data; uncertainty is present only in the linear constraints. As a result, techniques from robust linear programming (Appendix A) can be used to robustify the uncertain constraints. Under a polyhedral uncertainty set these robust linear constraints are equivalent to a set of linear constraints, allowing the approximate RGP to be transformed into a GP. A similar transformation can also be done for coefficients in an elliptical uncertainty set.

If the exponents are also in an elliptical uncertainty set, the transformed program is no longer a GP but becomes a cone program with exponential- and second-order-cone (SOC) constraints. The authors have used programs such as *ECOS* [13] and *SCS* [23] to solve such problems; they can also be solved as a GP by further approximating SOC constraints with GP-compatible softmax functions [16].

**DEFINITION 3.3.** *Two monomials are directly dependent if both depend on the same perturbation element  $\zeta_i$ . e.g.  $M_1 = e^{\mathbf{a}_1(\zeta_1)\mathbf{x}+b_1}$  and  $M_2 = e^{\mathbf{a}_3(\zeta_1,\zeta_2)\mathbf{x}+b_3}$  are directly dependent as they share the perturbation element  $\zeta_1$ .*

**DEFINITION 3.4.** *Two monomials are indirectly dependent if they are both dependent on a third monomial. e.g. with  $M_1 = e^{\mathbf{a}_1(\zeta_1)\mathbf{x}+b_1}$ ,  $M_2 = e^{\mathbf{a}_1(\zeta_2)\mathbf{x}+b_2}$ , and  $M_3 = e^{\mathbf{a}_3(\zeta_1,\zeta_2)\mathbf{x}+b_3}$ ,  $M_1$  and  $M_2$  are indirectly dependent because both share a perturbation element with (and are hence directly dependent on)  $M_3$ .*

**DEFINITION 3.5.** *Two monomials are independent if they are neither directly nor indirectly dependent.*

**DEFINITION 3.6.** *Two coefficients are consistently dependent if they both increase or both decrease as each perturbation element varies. e.g. if  $b_{11} = b_{11}^0 + \zeta_1$  and  $b_{12} = b_{12}^0 + 3\zeta_1 - \zeta_2$ , then  $b_{11}$  and  $b_{12}$  are dependent in a consistent way because their coefficients for  $\zeta_1$  have the same sign.*

Despite the decoupling of monomials, this formulation is exactly equivalent to (7) if each posynomial satisfies at least one of the following conditions:

- $C_1$ : Only one monomial in the posynomial has uncertain parameters
- $C_2$ : All monomials in the posynomial are either independent or dependent by a shared uncertain monomial factor, e.g.

$$p = e^{a_1(\zeta)\mathbf{x}_1+a_2\mathbf{x}_2+b_1(\zeta)} + e^{a_5\mathbf{x}_1+b_3} + e^{a_1(\zeta)\mathbf{x}_1+a_3\mathbf{x}_3+b_1(\zeta)} + e^{a_1(\zeta)\mathbf{x}_1+a_4\mathbf{x}_2+b_1(\zeta)} \\ = e^{a_1(\zeta)\mathbf{x}_1+b_1(\zeta)} (e^{a_2\mathbf{x}_2} + e^{a_3\mathbf{x}_3} + e^{a_4\mathbf{x}_2}) + e^{a_5\mathbf{x}_1+b_3}$$

- $C_3$ : The perturbation set is independent, e.g.  $\mathcal{Z} = \{\zeta \in \mathbb{R}^L : \|\zeta\|_\infty \leq \Gamma\}$ . Moreover, the monomials in the posynomial are either independent, or their coefficients are consistently dependent.

Specifically, if the  $i^{th}$  posynomial satisfies  $C_1$ ,  $C_2$ , or  $C_3$  then its maximum under perturbation will always equal the sum of the maximums of each decoupled monomial:

$$\max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k=1}^{K_i} e^{\mathbf{a}_{ik}(\zeta)\mathbf{x}+b_{ik}(\zeta)} \right\} = \sum_{k=1}^{K_i} \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathbf{a}_{ik}(\zeta)\mathbf{x}+b_{ik}(\zeta)} \right\}$$

Even when applied to a geometric program whose every posynomial does not satisfy  $C_1$ ,  $C_2$ , or  $C_3$ , this formulation has the advantage of requiring only a small number of GP or conic constraints. If  $\mathbf{P} = \{i : K_i \geq 3\}$ ,  $\mathbf{N} = \{i : K_i = 2\}$ , and  $\mathbf{M} = \{i : K_i = 1\}$  then the total number of constraints in (12) is

$$\sum_{k \in \mathbf{P}} (K_i + 1) + 3|\mathbf{N}| + |\mathbf{M}|$$

which is

$$(r-1)\sum_{k \in \mathbf{P}}(K_i-3) + 2(r-2)|\mathbf{P}| + (r-3)|\mathbf{N}|$$

constraints fewer than in the two-term formulation of [18]. Specifically, this decoupled-monomial formulation will always have fewer constraints than (8) if 3 or more piecewise sections are used, which will almost always be the case; three linear sections form a poor approximation of a two-term posynomial. Furthermore, this formulation is only more conservative than that in (8) if the number of sections is high enough and one of the approximated two-term posynomials captures an active dependence.

**4. Equivalent Intermediate Formulation.** Our focus is now on modifying the methodology of Section 3 to make better approximations of (7). This section presents an enhanced formulation of (12) that is equivalent to the robust counterparts in (7) but with smaller, easier to handle posynomial constraints.

In order to divide the posynomial into smaller posynomials, consider the set  $\mathbf{I}_i = \{1, 2, \dots, K_i\}$  associated with the  $i^{\text{th}}$  constraint, and define the equivalence relation  $\mathcal{R}$  (see Appendix C for the definition of an equivalence relation) given by

$$(13) \quad \mathcal{R} = \{k_1 \sim k_2 \iff e^{\mathbf{a}_{ik_1}(\zeta)\mathbf{x}+b_{ik_1}(\zeta)} \text{ and } e^{\mathbf{a}_{ik_2}(\zeta)\mathbf{x}+b_{ik_2}(\zeta)} \text{ are dependent}\}$$

$\mathcal{R}$  splits  $\mathbf{I}_i$  into equivalence classes  $S_{i,1}, S_{i,2}, \dots, S_{i,N_e^i}$ ,  $N_e^i \leq K_i$ . Because these classes are completely independent it must be the case that

$$\max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k=1}^{K_i} e^{\mathbf{a}_{ik}(\zeta)\mathbf{x}+b_{ik}(\zeta)} \right\} = \sum_{j=1}^{N_e^i} \max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k \in S_{i,j}} e^{\mathbf{a}_{ik}(\zeta)\mathbf{x}+b_{ik}(\zeta)} \right\}$$

The robust counterparts of (7) are thus equivalent to

$$(14) \quad \begin{aligned} \min \quad & f_0(x) \\ \text{s.t.} \quad & \sum_{j=1}^{N_e^i} e^{t_{ij}} \leq 1 \quad \forall i \in 1, \dots, m \\ & \max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k \in S_{i,j}} e^{\mathbf{a}_{ik}(\zeta)\mathbf{x}+b_{ik}(\zeta)} \right\} \leq e^{t_{ij}} \quad \forall i \in 1, \dots, m \\ & \quad \quad \quad \forall j \in 1, \dots, N_e^i \end{aligned}$$

Our main concern is now the larger posynomials, which motivate the introduction of a **categorized form** of a robust GP, in which constraints are categorized into three sets. The set of monomial constraints is labeled by  $\mathbf{M} \equiv \{(i, j) : |S_{i,j}| = 1\}$ , two-term posynomials by  $\mathbf{N} \equiv \{(i, j) : |S_{i,j}| = 2\}$ , and constraints with three or more posynomials are labeled by  $\mathbf{P} \equiv \{(i, j) : |S_{i,j}| \geq 3\}$ . Further defining  $S_{i,j}^k$  as the  $k^{\text{th}}$  element of  $S_{i,j}$ , the **categorized form** is

$$(15) \quad \begin{aligned} \min \quad & f_0(x) \\ \text{s.t.} \quad & \sum_{j=1}^{N_e^i} e^{t_{ij}} \leq 1 \quad \forall i \in 1, \dots, m \\ & \max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k \in S_{i,j}} e^{\mathbf{a}_{ik}(\zeta)\mathbf{x}+b_{ik}(\zeta)} \right\} \leq e^{t_{ij}} \quad \forall (i, j) \in \mathbf{P} \\ & \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathbf{a}_{iS_{i,j}^1}(\zeta)\mathbf{x}+b_{iS_{i,j}^1}(\zeta)} + e^{\mathbf{a}_{iS_{i,j}^2}(\zeta)\mathbf{x}+b_{iS_{i,j}^2}(\zeta)} \right\} \leq e^{t_{ij}} \\ & \quad \quad \quad \forall (i, j) \in \mathbf{N} \\ & \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathbf{a}_{iS_{i,j}^1}(\zeta)\mathbf{x}+b_{iS_{i,j}^1}(\zeta)} \right\} \leq e^{t_{ij}} \quad \forall (i, j) \in \mathbf{M} \end{aligned}$$

Figure 1 presents an illustrative example of this partitioning.

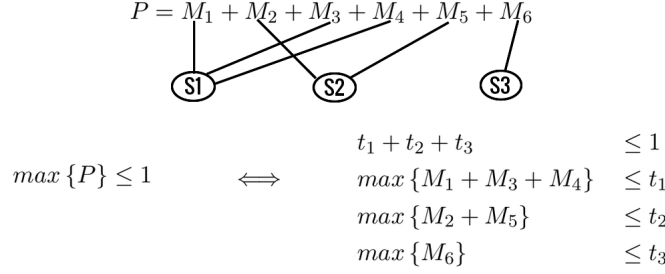


Fig. 1: Example of partitioning a large posynomial into smaller posynomials.

This partitioning will not be of much benefit when all monomials in a posynomial are uncertain and dependent on each other. However, when most monomials are certain, or when some are independent, a large posynomial can be effectively reduced into several smaller ones, or even into monomials. In practice, most GPs for engineering design are only sparsely dependent, and so this formulation provides a significant decrease (often by more than half) in the number of constraints.

**5. Robust Two Term Posynomials[Review].** We will now review work done in [18] on using piecewise-linear functions to approximate two-term posynomials.

Consider the convex function  $\phi(x) = \log(1 + e^x)$ . The unique best piecewise-linear convex lower approximation of  $\phi$  with  $r$  terms is defined as

$$(16) \quad \underline{\phi}_r = \begin{cases} 0 & \text{if } x \in (-\infty, x_1] \\ \underline{a}_i x + \underline{b}_i & \text{if } x \in [x_i, x_{i+1}], i = 1, 2, \dots, r-2 \\ x & \text{if } x \in [x_{r-1}, \infty) \end{cases}$$

such that

$$\begin{aligned} x_1 &< x_2 < \dots < x_{r-1} \\ 0 &= \underline{a}_0 < \underline{a}_1 < \underline{a}_2 < \dots < \underline{a}_{r-2} < \underline{a}_{r-1} = 1 \\ 1 &= \underline{a}_i + \underline{a}_{r-i-1} \quad \forall i \in \{0, 1, \dots, r-1\} \\ \underline{b}_i &= \underline{b}_{r-i-1} \quad \forall i \in \{1, \dots, r-2\} \\ \underline{b}_0 &= \underline{b}_{r-1} = 0 \end{aligned}$$

Moreover,  $\exists \tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{r-2} \in \mathbf{R}$  satisfying

$$x_1 < \tilde{x}_1 < x_2 < \tilde{x}_2 < \dots < x_{r-2} < \tilde{x}_{r-2} < x_{r-1}$$

such that  $\underline{a}_i x + \underline{b}_i$  is tangent to  $\phi$  at  $\tilde{x}_i$ .

The maximum approximation error  $\epsilon_r$  of this piecewise-linearization occurs at the break points  $x_1, \dots, x_{r-1}$  (for a constructive algorithm of the above coefficients, refer to [18]). This piecewise-linearization can then be used to safely approximate a two-term posynomial.



Letting  $h = \log(e^{y_1} + e^{y_2})$  be a two term posynomial in log-space, where  $y_1 = \mathbf{a}_1 \mathbf{x} + b_1$  and  $y_2 = \mathbf{a}_2 \mathbf{x} + b_2$ , the unique best  $r$ -term piecewise-linear lower approximation is

$$(17) \quad \underline{h}_r = \max\{\underline{a}_{r-1}y_1 + \underline{a}_0y_2 + \underline{b}_0, \underline{a}_{r-2}y_1 + \underline{a}_1y_2 + \underline{b}_1, \underline{a}_{r-3}y_1 + \underline{a}_2y_2 + \underline{b}_2, \dots, \underline{a}_1y_1 + \underline{a}_{r-2}y_2 + \underline{b}_{r-2}, \underline{a}_0y_1 + \underline{a}_{r-1}y_2 + \underline{b}_{r-1}\}$$

while its unique best  $r$ -term piecewise-linear upper approximation is

$$(18) \quad \overline{h}_r = \underline{h}_r + \epsilon_r$$

where  $\underline{a}_0, \underline{a}_1, \dots, \underline{a}_{r-1}$  and  $\underline{b}_0, \underline{b}_1, \dots, \underline{b}_{r-1}$  are as given in equation (16), and  $\epsilon_r$  is the maximum error between  $\phi$  and  $\underline{\phi}_r$ .

Since  $\overline{h}_r \geq h$ , then each posynomial in the set  $\mathbf{N}$  can be safely approximated by its own  $\overline{h}_r$ . Replacing the two term posynomial constraints by their piecewise-linear lower approximation will lead to a relaxed formulation, and thus the difference between the safe formulation's solution and the relaxed formulation's solution is an indication of how good an approximation is, and whether the number of piecewise-linear terms  $r$  should be further increased or not.

Because a piecewise-linear constraint can be represented as a set of linear constraints, these two-term posynomial approximations can be transformed to GP. This GP can be made less conservative by increasing the number of piecewise-linear terms.

**6. Robust Large Posynomials.** Having introduced improved techniques for monomials and two-term posynomials, we now turn to better approximations for posynomial constraints with more than two terms (those associated with  $\mathbf{P}$ ). Two novel methodologies for approximating large posynomials will be presented: the first is only able to transform posynomials whose coefficients are uncertain, while the second can transform posynomials whose coefficients and exponents are uncertain.

**6.1. Linearized Perturbations Formulation.** In the majority of engineering-design constraints, exponents are derived with certainty from physical laws or dimensional analysis, and so programs where uncertainty is only present in the coefficients are common. Let

$$\mathbf{b} = \mathbf{b}^0 + \sum_{l=1}^L \mathbf{b}^l \zeta_l$$

where  $\zeta \in \mathcal{Z}$  is as given by equation (5). The second set of constraints in equation (15) is then equivalent to

$$\begin{aligned} \max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k \in S_{i,j}} e^{\mathbf{a}_{ik} \mathbf{x} + b_{ik}^0} e^{\sum_{l=1}^L b_{ik}^l \zeta_l} \right\} &\leq e^{t_{ij}} \quad \forall (i, j) \in \mathbf{P} \\ \Leftrightarrow \max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k \in S_{i,j}} \left( \prod_{l=1}^L e^{b_{ik}^l \zeta_l} \right) e^{\mathbf{a}_{ik} \mathbf{x} + b_{ik}^0} \right\} &\leq e^{t_{ij}} \quad \forall (i, j) \in \mathbf{P} \end{aligned}$$

Thus by applying the following change of variable

$$v_{i,j}^k = e^{\mathbf{a}_{ik} \mathbf{x} + b_{ik}^0} \quad \forall k \in S_{i,j}$$

the constraint above can be rewritten as

$$(19) \quad \max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k \in S_{i,j}} \left( \prod_{l=1}^L e^{b_{ik}^l \zeta_l} \right) v_{i,j}^k \right\} \leq e^{t_{ij}} \quad \forall (i, j) \in \mathbf{P}$$

Although the constraints in (19) are linear in terms of  $\mathbf{v}_{i,j}$ , the perturbations are not affine but exponential, and should be linearized to improve tractability.

**6.1.1. Linearizing Perturbations.** The convexity of exponential perturbations (see Figure 2) implies that there exists some half-space  $[\mathbf{f}_{i,j}^k]^T \boldsymbol{\zeta} + g_{i,j}^k$  such that

$$[\mathbf{f}_{i,j}^k]^T \boldsymbol{\zeta} + g_{i,j}^k \geq \prod_{l=1}^L e^{b_{ik}^l \zeta_l} \quad \forall (i,j) \in \mathbf{P} \quad \forall k \in S_{i,j}$$

on some finite domain for  $\boldsymbol{\zeta}$ .

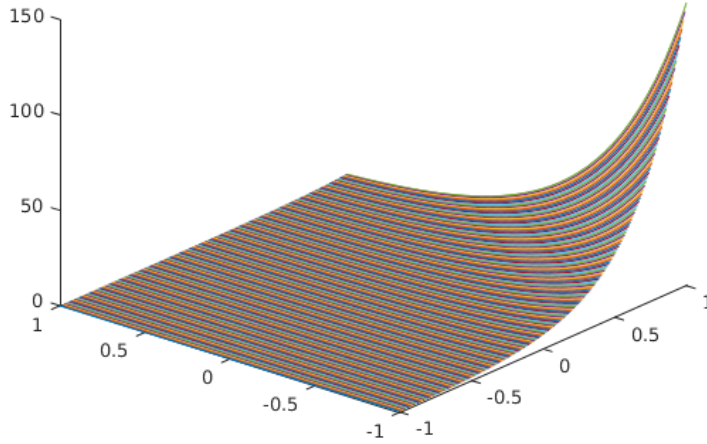


Fig. 2: Example of the convexity of exponential perturbations.

Accordingly, a possible safe approximation of the constraints in equation (19) is

$$(20) \quad \max_{\boldsymbol{\zeta} \in \mathcal{Z}} \left\{ \sum_{k \in S_{i,j}} ([\mathbf{f}_{i,j}^k]^T \boldsymbol{\zeta}) v_{i,j}^k \right\} + \sum_{k \in S_{i,j}} g_{i,j}^k v_{i,j}^k \leq e^{t_{ij}} \quad \forall (i,j) \in \mathbf{P}$$

To construct the half-space  $[\mathbf{f}_{i,j}^k]^T \boldsymbol{\zeta} + g_{i,j}^k$ , and taking into consideration the fact that  $-1 \leq \zeta_l \leq 1$  for  $l = 1, \dots, L$ , we suggest the following algorithm

1. Find the list of vertices  $\mathcal{V}$  of the unit box in  $\mathbf{R}^L$
2. Find the list of values  $\mathcal{O}$  of  $\prod_{l=1}^L e^{b_{ik}^l \zeta_l}$  at the vertices  $\mathcal{V}$ , note that the  $i^{th}$  vertex corresponds to the  $i^{th}$  value.
3. Find the maximum  $M_k$  and minimum  $m_k$  of  $\mathcal{O}$  and their corresponding vertices  $\boldsymbol{\zeta}_M$  and  $\boldsymbol{\zeta}_m$ .
4. Solve the least-squares problem

$$\min_{\mathbf{f}_{i,j}^k, g_{i,j}^k} \sqrt{\sum_{\alpha=1}^{|\mathcal{O}|} ([\mathbf{f}_{i,j}^k]^T \boldsymbol{\nu}_\alpha + g_{i,j}^k - \mathcal{O}_\alpha)^2}$$

$$\text{such that } [\mathbf{f}_{i,j}^k]^T \boldsymbol{\zeta}_M + g_{i,j}^k = M_k \text{ and } [\mathbf{f}_{i,j}^k]^T \boldsymbol{\zeta}_m + g_{i,j}^k = m_k$$

Figure 2 shows a 2D example where the half-space passes through the highest point (1, -1) and the lowest point (-1, 1).

This half-space is safe for all types of uncertainty sets, although more-specific half-spaces can be constructed for non-exponential uncertainty sets. For instance, with elliptical uncertainty sets, the vertices in the first step might be replaced by an equidistant set of samples from the surface of the uncertainty set.

**6.1.2. Signomial Programming Compatible Constraint.** Equation (20) can be transformed with robust linear programming techniques. Unfortunately, because some components of  $\mathbf{f}_{i,j}^k$  might not be positive, the resulting set of robust constraints is not always GP compatible. Appendix D defines signomial programming and clarifies how it is useful to our discussion; in short, it allows the specification and efficient solving of a difference-of-convex program with these negative coefficients.

Signomial problems generally do not need a starting point to solve, but specifying one can speed convergence, and in robust programming there are two quick and obvious candidates for a starting point: the nominal solution of the original program, or the conservative solution of the GP-compatible formulation in 3.

The number of additional possibly SP constraints needed per uncertain posynomial is equal to the number of uncertain parameters in that posynomial. Decoupling the monomials in each posynomial is equivalent to evaluating the exponential perturbations at its largest value, however, in (20), the perturbation will be evaluated on a lower point on the half space. As a result, this formulation is guaranteed to be less conservative than that of Section 3, or the same if both are exact.

**6.2. Best Pairs Formulation.** We now discuss a framework for robustifying constraints with uncertain coefficients and exponents. Consider the constraints in **P**

$$(21) \quad \max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k \in S_{i,j}} e^{\mathbf{a}_{ik}(\zeta) \mathbf{x} + b_{ik}(\zeta)} \right\} \leq e^{t_{ij}} \quad \forall (i, j) \in \mathbf{P}$$

Let  $\mathcal{P}_{i,j}$  be the set of permutations on  $S_{i,j}$ , and let  $\phi_k$  be the image of an element  $k \in S_{i,j}$  under the permutation  $\phi \in \mathcal{P}_{i,j}$ . Accordingly, (21) is equivalent to

$$(22) \quad \max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k \in S_{i,j}} e^{\mathbf{a}_{i\phi_k}(\zeta) \mathbf{x} + b_{i\phi_k}(\zeta)} \right\} \leq e^{t_{ij}} \quad \forall (i, j) \in \mathbf{P} \quad \phi \in \mathcal{P}_{i,j}$$

Our goal is to maximize pairs of monomials while trying to choose the least conservative combination. Let

$$\mathbf{a}_{iS_{i,j}^k}(\zeta) \mathbf{x} + b_{iS_{i,j}^k}(\zeta) = \mathcal{L}_{i,j}^k \quad \forall (i, j) \in \mathbf{P}$$

where  $S_{i,j}^k$  is the  $k^{th}$  element of  $S_{i,j}$ , and assume that  $|S_{i,j}|$  is even, such that

$$\max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k=1}^{|S_{i,j}|} e^{\mathcal{L}_{i,j}^k} \right\} \leq \sum_{k=1}^{|S_{i,j}|/2} \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathcal{L}_{i,j}^{\phi_{2k-1}}} + e^{\mathcal{L}_{i,j}^{\phi_{2k}}} \right\}$$

for any permutation  $\phi \in \mathcal{P}_{i,j}$ . Therefore

$$\max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k=1}^{|S_{i,j}|} e^{\mathcal{L}_{i,j}^k} \right\} \leq \min_{\phi \in \mathcal{P}_{i,j}} \left\{ \sum_{k=1}^{|S_{i,j}|/2} \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathcal{L}_{i,j}^{\phi_{2k-1}}} + e^{\mathcal{L}_{i,j}^{\phi_{2k}}} \right\} \right\}$$

and

$$(23) \quad \min_{\phi \in \mathcal{P}_{i,j}} \left\{ \sum_{k=1}^{|S_{i,j}|/2} \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathcal{L}_{i,j}^{\phi_{2k-1}}} + e^{\mathcal{L}_{i,j}^{\phi_{2k}}} \right\} \right\} \leq e^{t_{ij}} \quad \forall (i, j) \in \mathbf{P}$$

is a safe approximation of (22). From the fact that

$$\max(a + b) + \max(c + d) = \max(c + d) + \max(a + b) = \max(b + a) + \max(d + c)$$

we see that some perturbations result in identical safe approximations. The permutation set can be modified to contain differing permutations only; let  $\mathcal{P}_{i,j}$  represent the set of differing permutations. The number of differing permutations is given by

$$(24) \quad |\mathcal{P}_{i,j}| = \frac{\binom{|S_{i,j}|}{2} \binom{|S_{i,j}|-2}{2} \binom{|S_{i,j}|-4}{2} \dots \binom{4}{2}}{(|S_{i,j}|/2)!}$$

If the number of monomial terms for a given posynomial is large, then  $|\mathcal{P}_{i,j}|$  might become quite large. Therefore, we might choose to work with a subset  $\hat{\mathcal{P}}_{i,j}$  of  $\mathcal{P}_{i,j}$ , where the permutations are either chosen depending on the structure of the posynomial or randomly selected. Constraints associated with  $\mathbf{P}$  will be replaced by

$$(25) \quad \sum_{k=1}^{|S_{i,j}|/2} \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathcal{L}_{i,j}^{\phi_{2k-1}}} + e^{\mathcal{L}_{i,j}^{\phi_{2k}}} \right\} \leq e^{t_{ij}} \quad \forall (i,j) \in \mathbf{P} \quad \phi \in \hat{\mathcal{P}}_{i,j}$$

The constraints in (25) are safe approximations for those in (22), and equivalent to

$$(26) \quad \begin{aligned} \sum_{k=1}^{|S_{i,j}|/2} e^{z_{ij}^k} &\leq e^{t_{ij}} & \forall (i,j) \in \mathbf{P} \\ \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathcal{L}_{i,j}^{\phi_{2k-1}}} + e^{\mathcal{L}_{i,j}^{\phi_{2k}}} \right\} &\leq e^{z_{ij}^k} & \forall (i,j) \in \mathbf{P} \quad \forall k \in \{1, \dots, |S_{i,j}|/2\} \end{aligned}$$

Accordingly, the robust counterparts in section 2 will be safely approximated by

$$(27) \quad \begin{aligned} \min \quad & f_0(x) \\ \text{s.t.} \quad & \sum_{j=1}^{N_e^i} e^{t_{ij}} \leq 1 & \forall i \in 1, \dots, m \\ & \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathcal{L}_{i,j}^1} \right\} \leq e^{t_{ij}} & \forall (i,j) \in \mathbf{M} \\ & \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathcal{L}_{i,j}^1} + e^{\mathcal{L}_{i,j}^2} \right\} \leq e^{t_{ij}} & \forall (i,j) \in \mathbf{N} \\ & \sum_{k=1}^{|S_{i,j}|/2} e^{z_{ij}^k} \leq e^{t_{ij}} & \forall (i,j) \in \mathbf{P} \\ & \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathcal{L}_{i,j}^{\phi_{2k-1}}} + e^{\mathcal{L}_{i,j}^{\phi_{2k}}} \right\} \leq e^{z_{ij}^k} & \forall (i,j) \in \mathbf{P} \\ & & \forall k \in \{1, \dots, |S_{i,j}|/2\} \\ & & \phi \in \hat{\mathcal{P}}_{i,j} \end{aligned}$$

Each constraint in (27) is composed of monomials, two-term posynomials, or data-deprived large posynomials, and so the program is tractable.

Our final step is finding good permutations, i.e. permutations that will make our solution less conservative. For that sake, the following lemma will be utilized

LEMMA 6.1. *Considering the two optimization problems*

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathcal{S}_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \dots, n \end{aligned}$$

and

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathcal{T}_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \dots, n \end{aligned}$$

337 let  $\mathbf{x}_1$  and  $\mathbf{x}_2$  be their respective solutions. If  $\mathcal{T}_i(\mathbf{x}_1) \leq \mathcal{S}_i(\mathbf{x}_1) \quad \forall i \in \{1, 2, \dots, n\}$ , then  
 338  $f(\mathbf{x}_2) \leq f(\mathbf{x}_1)$

*Proof.*

$$\mathcal{T}_i(\mathbf{x}_1) \leq \mathcal{S}_i(\mathbf{x}_1) \leq 0 \quad \forall i \in \{1, 2, \dots, n\}$$

So  $\mathbf{x}_1$  is a feasible solution for the second optimization problem. But since  $\mathbf{x}_2$  is the optimal solution of the second problem, then

$$f(\mathbf{x}_2) \leq f(\mathbf{x}_{feasible}) \implies f(\mathbf{x}_2) \leq f(\mathbf{x}_1)$$

339

□

340 Starting from the above lemma, the algorithm below illustrates how relatively  
 341 good permutations might be chosen:

342

343

344

345

346

347

348

349

350

1. Randomly choose the permutations  $\phi$  from the set  $\hat{\mathcal{P}}_{i,j} \quad \forall (i, j) \in \mathbf{P}$
2. Solve the optimization problem in (27), and let  $\mathbf{x}_1$  be the solution
3. Repeat
  - (a)  $\forall (i, j) \in \mathbf{P}$ , select the new permutations  $\phi \in \hat{\mathcal{P}}_{i,j}$  such that  $\phi$  minimizes
 
$$\sum_{k=1}^{|S_{i,j}|/2} \max_{\zeta \in \mathcal{Z}} \left\{ e^{\mathcal{L}_{i,j}^{\phi_{2k-1}}} + e^{\mathcal{L}_{i,j}^{\phi_{2k}}} \right\} \Big|_{\mathbf{x}_i}$$
  - (b) Solve the optimization problem in (27), and let  $\mathbf{x}_i$  be the solution
  - (c) If  $\mathbf{x}_i = \mathbf{x}_{i-1}$  : break

351

352

353

354

355

Although the final solution might not be the globally least-conservative over all possible permutations it is guaranteed to be the locally least-conservative given the descent algorithm based on Lemma (6.1). The Best Pairs Formulation is guaranteed to be less-conservative than the Two-term formulation because it involves less monomial decoupling and fewer piecewise-linear approximations.

356

357

358

359

360

**7. Applications.** In this section, uncertain design problems are transformed into robust programs to demonstrate and compare the methods above. The first problem (a simple aircraft model) is used mostly to clarify the concept, while the second (a large-scale solar-aircraft design code) shows the effectiveness of the tractable robust GP-approximation methods proposed.

361

362

**7.1. Simple Flight Design.** RGP formulation ideas were implemented on a simple wing design adapted from Hoburg [15].

$$\begin{aligned}
& \min \quad D \\
& s.t. \quad C_D \geq \frac{C_{DA0}}{S} + kC_f S_{wet} + \frac{C_L^2}{\pi A e} \\
& \quad W_W \geq W_{W_2} S + \frac{W_{W_1} N_{utl} A^{1.5} \sqrt{W_0 W S}}{\tau} \\
& \quad D \geq 0.5 \rho S C_D V^2 \\
& \quad Re \leq \frac{\rho}{\mu} V \sqrt{\frac{S}{A}} \\
& \quad C_f \geq \frac{0.074}{Re^{0.2}} \\
& \quad W \leq 0.5 \rho S C_L V^2 \\
& \quad W \leq 0.5 \rho S C_{Lmax} V_{min}^2 \\
& \quad W \geq W_0 + W_W
\end{aligned} \tag{28}$$

This 8-constraint problem has two design variables ( $A, S$ ), seven dependent free variables ( $C_D, C_L, C_f, Re, W, W_W, V$ ), and the thirteen uncertain parameters in Table 1. Note that the uncertainties specified in the table are the principal-axis widths of either the box or elliptical uncertainty sets. Note that uncertainties in this problem are only in the coefficients, and that constraints in (28) are predominantly monomials and two-term posynomials, with only one posynomial of more than two terms.

The problem is first solved for different sizes ( $\Gamma$  values) of box and elliptical uncertainty sets (defined in Appendix A), where the number of piecewise-linear terms is chosen such that the relative error between the solution of the upper tractable approximation and that of the lower tractable approximation is less than 0.1%. The design variables are then fixed for each solution so that the design can be simulated for 1000 realizations of the uncertain parameters to examine average design performance.

Uncertain Parameter	Value	Description
$C_{DA0}$	$0.035 \pm 42.85\% [m^2]$	Fuselage Drag Area
$k$	$1.17 \pm 31.11\%$	Form Factor
$S_{wet}$	$2.075 \pm 3.61\%$	Wetted Area Ratio
$e$	$0.92 \pm 7.60$	Oswald Efficiency Factor
$W_{W_2}$	$60 \pm 66\% [Pa]$	Wing Weight Coefficient 2
$W_{W_1}$	$12e^{-5} \pm 60\% [\frac{1}{m}]$	Wing Weight Coefficient 1
$N_{utl}$	$3.3 \pm 33.33\%$	Ultimate Load Factor
$W_0$	$6250 \pm 60\% [N]$	Aircraft Weight Excluding Wing
$\tau$	$0.12 \pm 33.33\%$	Airfoil Thickness to Chord Ratio
$\rho$	$1.23 \pm 10\% [\frac{kg}{m^3}]$	Density of Air
$\mu$	$1.775e^{-5} \pm 4.22\% [\frac{kg}{ms}]$	Viscosity of Air
$C_{Lmax}$	$1.6 \pm 25\%$	Maximum Lift coefficient of Wing
$V_{min}$	$25 \pm 20\% [\frac{m}{s}]$	Takeoff Speed

Table 1: Uncertain parameters in the simple wing design.

We can see from Figure 3 that while probability of failure goes to zero as  $\Gamma$  increases. Descriptions of how  $\Gamma$  determines the size of uncertainty sets are in Appen-

381 dices A.1 and A.2. For a given level of risk, the average performance of designs robust  
 382 to an elliptical uncertainty set is better than that of those using a box uncertainty  
 383 set.

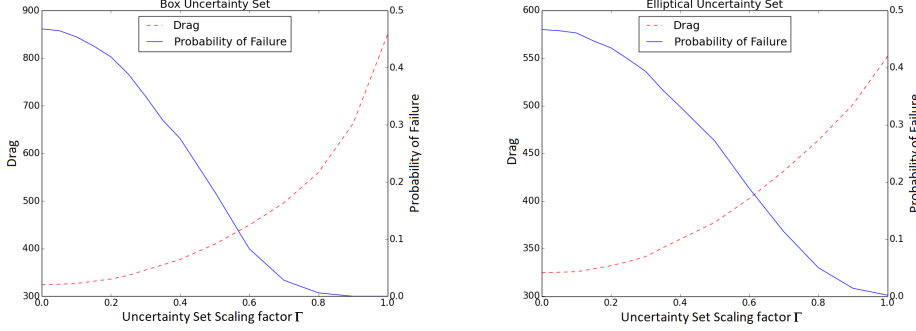


Fig. 3: Performance of the optimal robust simple wing, using the Best Pairs formulation, as a function of  $\Gamma$  for different uncertainty sets.

384 Table 2 and Figure 4 compare the results of robustifying and solving the simple  
 385 wing design problem with different methodologies. The number of piecewise-linear  
 386 terms per two-term posynomial needed to reach a 0.1% tolerance is much higher for  
 387 the Two Term formulation from [18] than for our formulations. Correspondingly, the  
 388 total number of constraints needed by the two-term formulation is also much larger.

Method	Uncertainty Set	Relative Error [%]	Number of PWL Sections	Number of Constraints
Two Term	Box	0.1	79	327
	Elliptical	0.1	70	291
Simple Conservative	Box	0	0	16
	Elliptical	0	0	16
Linearized Perturbations	Box	0	0	16
	Elliptical	0.1	31	49
Best Pairs	Box	0	0	16
	Elliptical	0.1	40	93
Deterministic	N/A	N/A	N/A	8

Table 2: Simple wing results

391 Our methodologies achieved an exact solution for the box uncertainty set as con-  
 392 dition  $C_3$  (Section 3) is satisfied, and therefore no piecewise-linearization is needed.  
 393 This explains the identical results of our three methodologies for the box uncertainty  
 394 set in Figures 4 and 5, contrasting with the more than 70 piecewise-linear sections  
 395 needed by the two-term formulation to achieve a 0.1% tolerance. Condition  $C_3$  is not  
 396 satisfied for the elliptical uncertainty set, and so here our simple method is noticeably  
 397 more conservative than those using piecewise-linear approximations.

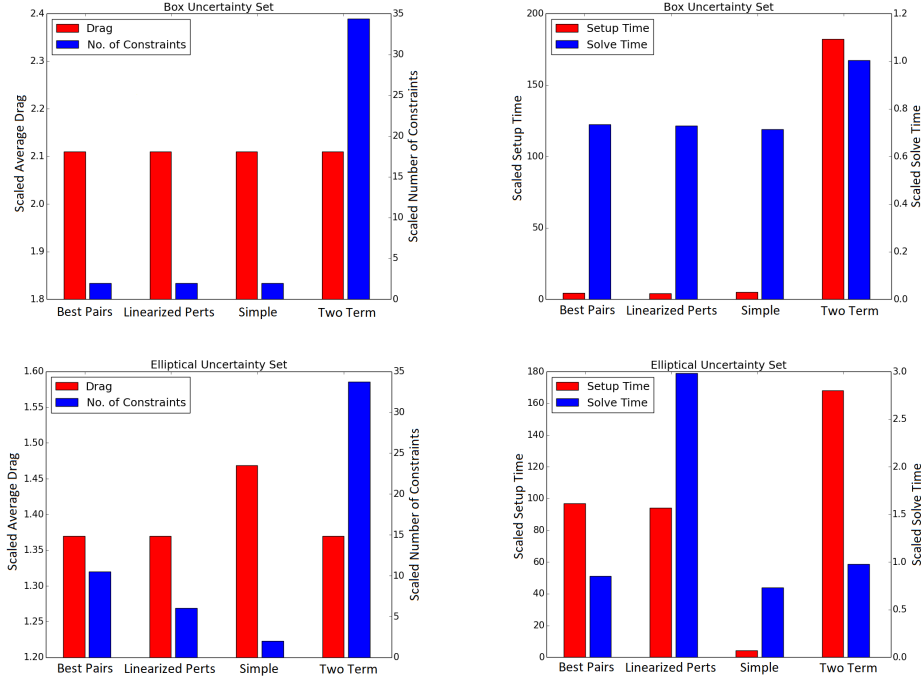


Fig. 4: Robust simple wing design results relative to the nominal design problem.

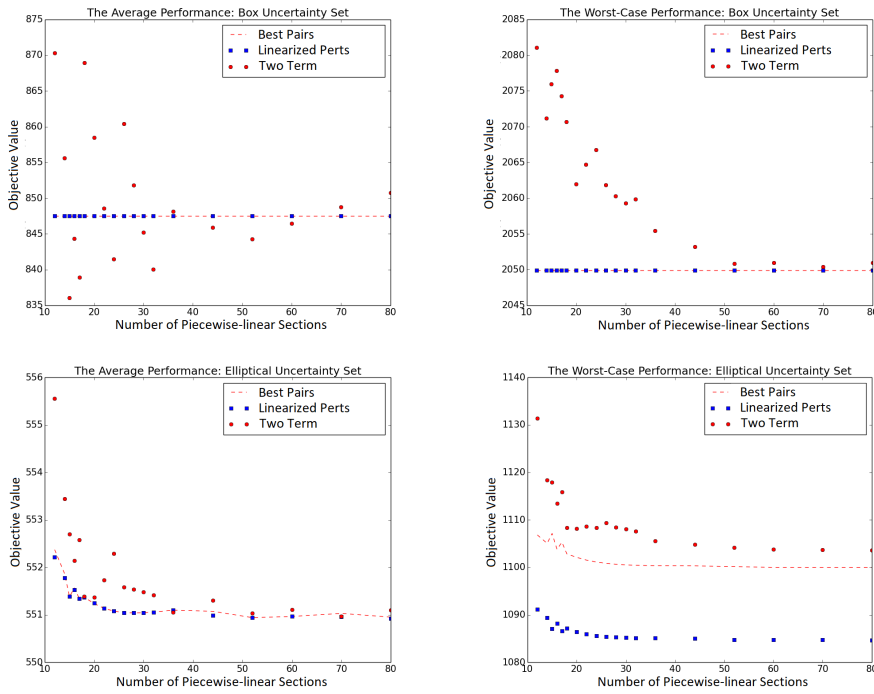


Fig. 5: The performance (objective value) of the simple wing design as a function of the number of piecewise-linear terms



**7.2. UAV Solar Model.** In previous GP modeling work, Burton presented a GP-compatible solar-powered aircraft model [9] designed to fulfill the requirements in Table 3. The nominal model is formed of 2698 constraints including 20 design variables, 875 dependent free variables, and 271 uncertain parameters. Table 4 presents a sample of the uncertain parameters in the solar aircraft design.

Mission Requirements	
Payload	10 <i>lbs</i>
Station Keeping	90% <i>winds</i>
Endurance	> 5 <i>days</i>
Season	<i>all seasons</i>
Altitude	> 4600 <i>m</i>
Latitude	$\pm 30^\circ$

Table 3: Mission requirements.

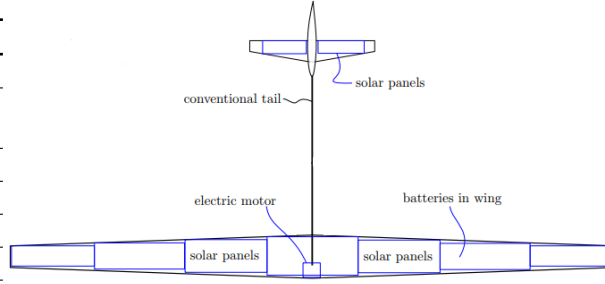


Fig. 6: Solar aircraft planform.

Uncertain Parameter	Value	Description
$\eta_m$	$0.95 \pm 7\%$	Motor Efficiency
$h_{batt}$	$350 \pm 15\% \left[\frac{Wh}{Kg}\right]$	Battery Specific Energy
$\eta_{solar}$	$0.22 \pm 15\%$	Solar Cell Efficiency
$V_{wind_{ref}}$	$100 \pm 3\% \left[\frac{m}{s}\right]$	Reference Wind Speed
$\eta_{prop}$	$0.8 \pm 10\%$	Propeller Efficiency

Table 4: A sample of the solar aircraft uncertain parameters

This problem was solved for an elliptical uncertainty set for various values of  $\Gamma$  with a maximum number of piecewise-linear terms of 50 and a desired relative error of 1% between the solution of the upper and lower tractable approximations. The average performance was then calculated from 300 realizations.

Table 5 and Figure 7 show that the number of constraints needed to formulate this approximate RGP with the Two Term formulation is extremely large even at a relative error of 2%. By comparison, the formulations presented in this paper have a relatively small number of constraints, primarily due to the partitioning of large posynomials into smaller ones.

Method	Uncertainty Set	Relative Error [%]	Number of PWL Sections	Number of Constraints
Two Term	Box	2.3	50	91,599
	Elliptical	2.2	50	91,599
Simple Conservative	Box	0	0	2,896
	Elliptical	0	0	2,896
Linearized Perturbations	Box	0	0	3,895
	Elliptical	0.9	31	2,899
Best Pairs	Box	0.7	4	2,939
	Elliptical	0.9	32	4,433
Deterministic	N/A	N/A	N/A	2,837

Table 5: Solar aircraft results.

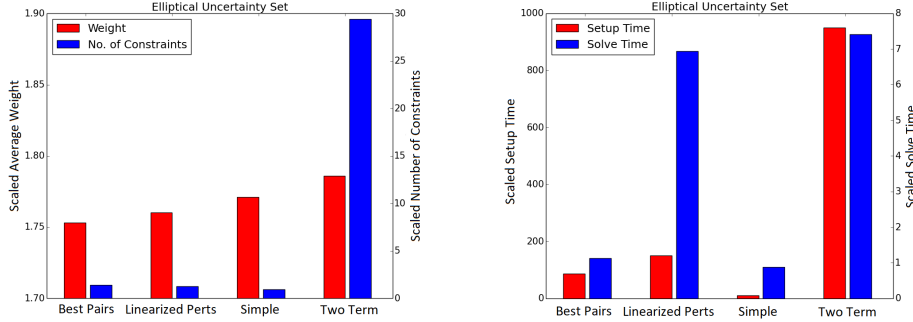


Fig. 7: Robust solar aircraft results scaled by nominal objective value, number of constraints, and timing.

Figure 8 shows that the probability of failure in the robust design goes to zero as  $\Gamma$  increases, while the nominal design fails with a probability of 0.7.

As shown in Figure 7, the programs influenced by our formulations take much less time to setup and solve than the Two Term formulation (with the exception of the the Linearized Perturbations solve), and all have better performance. Figure 9 additionally shows the fast convergence of the Best Pairs and Linearized Perturbations formulations with respect to the number of piecewise-linear terms used.

**8. Conclusion.** In this article we have described three different methodologies for creating approximate formulations of robust geometric programs. These methodologies build on each other to derive equivalent but increasingly less conservative formulations which are tractable as GPs, SPs, or conic optimization problems. Several aircraft design problems from the GP literature were robustified and a significant improvement over the state of art was shown, implying that robust programming may be a promising way to model uncertainties in engineering design. We are currently extending this work to solve Robust Signomial Programming problems.

## Appendix A. Robust Linear Programming.

This section reviews robust linear programming, a building block to formulate a tractable approximate robust geometric program. Two different examples of perturbation sets will be used for clarification: box and elliptical. Those sets will be used for our discussion throughout the article due to the fact that they are important and frequently used.

Consider the system of linear constraints

$$\mathbf{A}\mathbf{x} + \mathbf{b} \leq 0$$

where

$$\mathbf{A} \text{ is } m \times n$$

$$\mathbf{x} \text{ is } n \times 1$$

$$\mathbf{b} \text{ is } m \times 1$$

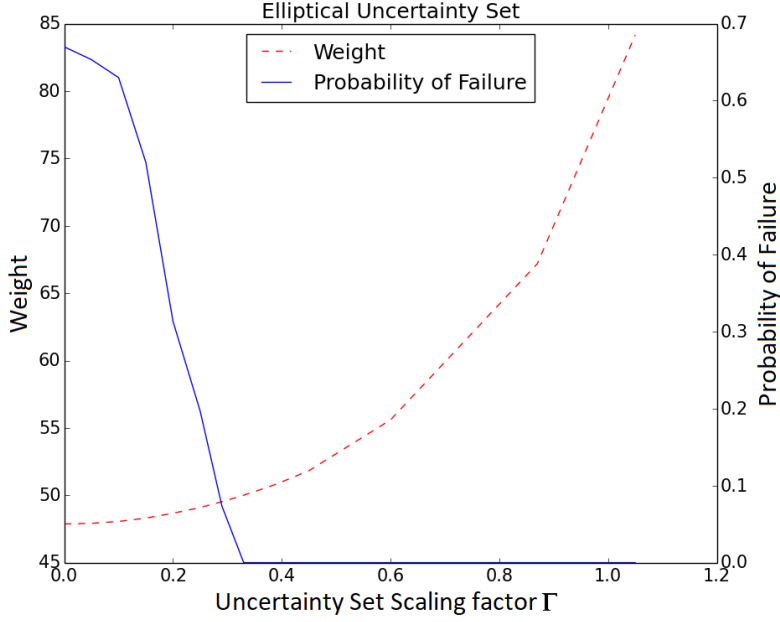


Fig. 8: Performance of the solar aircraft design based on the Best Pairs formulation, as a function of  $\Gamma$ .

Assuming that data is uncertain and is given by equations (4) and (5), the constraints should be satisfied for all values of  $\zeta$ , and thus the robust counterparts of the  $i^{th}$  linear constraint is

$$\sup_{\zeta \in \mathcal{Z}} \left\{ \sum_{l=1}^L \zeta_l ([a_i^l]^T \mathbf{x} + b_i^l) \right\} \leq -[a_i^0]^T \mathbf{x} - b_i^0$$

This is equivalent to the optimization problem

$$(29) \quad \max_{\zeta, \mathbf{u}} \left\{ \sum_{l=1}^L \zeta_l ([a_i^l]^T \mathbf{x} + b_i^l) : \mathbf{F}\zeta + \mathbf{G}\mathbf{u} + \mathbf{h} \in \mathbf{K} \right\} \leq -[a_i^0]^T \mathbf{x} - b_i^0$$

Applying the conic duality theorem, equation (29) is equivalent to

$$(30) \quad \begin{aligned} \mathbf{h}^T \mathbf{y}_i + [a_i^0]^T \mathbf{x} + b_i^0 &\leq 0 \\ \mathbf{G}^T \mathbf{y}_i &= \mathbf{0} \\ (\mathbf{F}^T \mathbf{y}_i)_l + [a_i^l]^T \mathbf{x} + b_i^l &= 0 \quad l = 1, 2, \dots, L \\ \mathbf{y}_i &\in \mathbf{K}^* \end{aligned}$$

where  $\mathbf{y}_i \in \mathbb{R}^N$ , and  $\mathbf{K}^*$  is the dual cone of  $\mathbf{K}$  (see Appendix B) [3].

**A.1. Box Uncertainty Sets.** If the perturbation set  $\mathcal{Z}$  given in equation (5) is a box uncertainty set, i.e.  $\|\zeta\|_\infty \leq \Gamma$ , then

- $\mathbf{F}\zeta = [\zeta; 0]$
- $\mathbf{G} = \mathbf{0}$ ,  $\mathbf{h} = [0_{L \times 1}; \Gamma]$

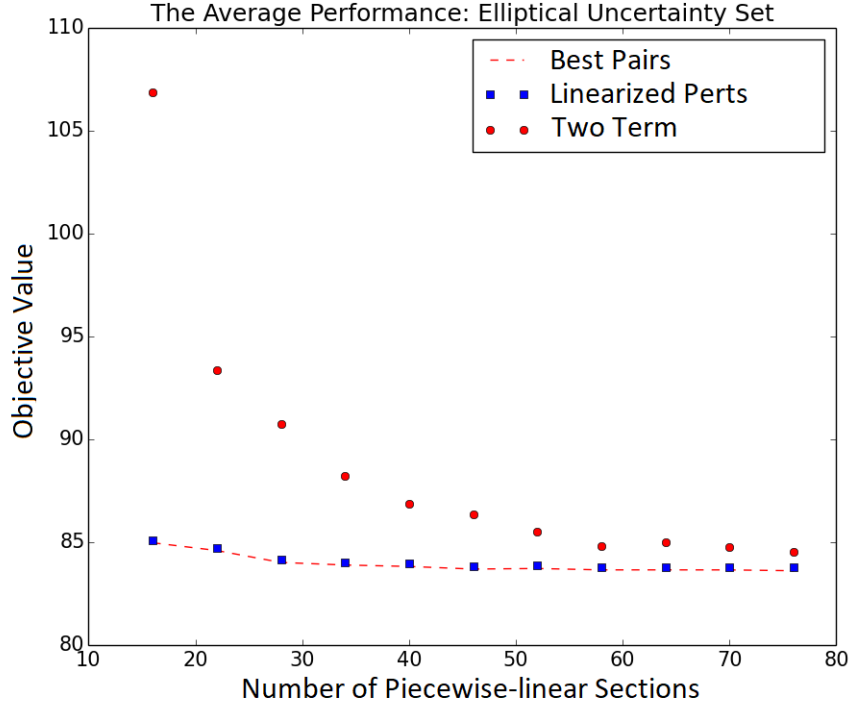


Fig. 9: The performance of the solar aircraft design as a function of the number of piecewise-linear terms for elliptical uncertainty set

- $\mathbf{K} = \{(z; t) \in \mathbb{R}^L \times \mathbb{R} : t > \|z\|_\infty\}$
- the dual cone  $\mathbf{K}^* = \{(z; t) \in \mathbb{R}^L \times \mathbb{R} : t > \|z\|_1\}$

and therefore, equation (29) is equivalent to

$$(31) \quad \Gamma \sum_{l=1}^L | -b_i^l - \mathbf{a}_i^l \mathbf{x} | + \mathbf{a}_i^0 \mathbf{x} + b_i^0 \leq 0$$

If only  $\mathbf{b}$  is uncertain, i.e.  $\mathbf{A}^l = 0 \quad \forall l = 1, 2, \dots, L$ , then equation (31) will become

$$(32) \quad \sum_{l=1}^L \mathbf{a}_i^0 \mathbf{x} + b_i^0 + \Gamma \sum_{l=1}^L |b_i^l| \leq 0$$

which is a linear constraint.

On the other hand, if  $\mathbf{A}$  is also uncertain, then equation (31) is equivalent to the following set of linear constraints

$$(33) \quad \begin{aligned} \Gamma \sum_{l=1}^L w_i^l + \mathbf{a}_i^0 \mathbf{x} + b_i^0 &\leq 0 \\ -b_i^l - \mathbf{a}_i^l \mathbf{x} &\leq w_i^l \quad \forall l = 1, \dots, L \\ b_i^l + \mathbf{a}_i^l \mathbf{x} &\leq w_i^l \quad \forall l = 1, \dots, L \end{aligned}$$

**A.2. Elliptical Uncertainty Sets.** If the perturbation set  $\mathcal{Z}$  is now an elliptical uncertainty set, i.e.  $\sum_{l=1}^L \frac{\zeta_l}{\sigma_l} \leq \Gamma$ , then

- $\mathbf{F}\zeta = [\sigma^{-1}\zeta; 0]$  with  $\sigma = \text{diag}(\sigma_1, \dots, \sigma_L)$

- $\mathbf{G} = \mathbf{0}$ ,  $\mathbf{h} = [\mathbf{0}_{L \times 1}; \Gamma]$
- $\mathbf{K} = \{(z; t) \in \mathbb{R}^L \times \mathbb{R} : t > \|z\|_2\}$
- The dual cone  $\mathbf{K}^* = \mathbf{K}$  (Lorentz or second order cone)

and therefore, equation (29) is equivalent to

$$(34) \quad \Gamma \sqrt{\sum_{l=1}^L \sigma_l^2 (-b_l^l - \mathbf{a}_l^l \mathbf{x})^2} + \mathbf{a}_i^0 \mathbf{x} + b_i^0 \leq 0$$

which is a second order conic constraint.

If only  $\mathbf{b}$  is uncertain, i.e.  $\mathbf{A}^l = \mathbf{0} \quad \forall l = 1, 2, \dots, L$ , then equation (34) will become

$$(35) \quad \mathbf{a}_i^0 \mathbf{x} + b_i^0 + \Gamma \sqrt{\sum_{l=1}^L \sigma_l^2 (b_l^l)^2} \leq 0$$

which is a linear constraint ( $\sqrt{\sum_{l=1}^L \sigma_l^2 (b_l^l)^2}$  is a constant).

## Appendix B. Cones.

**B.1. Euclidean Space.** An Euclidean space is a finite dimensional linear space over real numbers equipped with an inner product  $\langle x, y \rangle_E$ .

**B.2. Cones.** A nonempty subset  $\mathbf{K}$  of an Euclidean space is called a cone if for any  $x \in \mathbf{K}$  and  $\alpha \geq 0$   $\alpha x \in \mathbf{K}$  [5].

A cone is said to be convex cone if  $\alpha, \beta \geq 0$  and  $x, y \in \mathbf{K}$ , we have  $\alpha x + \beta y \in \mathbf{K}$  [3].

**B.3. Dual Cones.** If  $\mathbf{K}$  is a cone in an euclidean space  $\mathbf{E}$ , then the set

$$\mathbf{K}^* = \{e \in \mathbf{E} : \langle e, h \rangle_E \geq 0 \quad \forall h \in \mathbf{K}\}$$

is also a cone and is called the cone dual to  $\mathbf{K}$  [3].

**Appendix C. Equivalence Relations.** Let  $\mathcal{R}$  be some relation on a set  $\mathbf{S}$ , and let  $x, y \in \mathbf{S}$ . We say  $x \simeq y$  if  $x$  and  $y$  are related.

A relation  $\mathcal{R}$  on a set  $\mathbf{S}$  is called an equivalence relation if the following is true

- the relation is reflexive, i.e. for all  $a \in \mathbf{S}$ ,  $a \simeq a$
- the relation is symmetric, i.e. for  $a, b \in \mathbf{S}$ , if  $a \simeq b$ , then  $b \simeq a$
- the relation is transitive, i.e. for  $a, b, c \in \mathbf{S}$ , if  $a \simeq b$  and  $b \simeq c$ , then  $a \simeq c$

An equivalence relation naturally partitions a set into equivalence classes. Those classes are such that if  $a$  and  $b$  belong to the same class then  $a \simeq b$ , and if  $a$  and  $b$  are not related, then they belong to different classes.

**Appendix D. Signomial Programming.** Signomials allow us to solve non log-convex optimization problem as sequential geometric programs [20]. A signomial program has the following form

$$(36) \quad \begin{aligned} & \text{minimize} && f_0(\mathbf{x}) \\ & \text{subject to} && f_i(\mathbf{x}) \leq 1 && i = 1, \dots, m_p \\ & && g_i(\mathbf{x}) - h_i(\mathbf{x}) \leq 1 && i = 1, \dots, m_s \end{aligned}$$

where  $f_i$ ,  $g_i$ , and  $h_i$  are posynomials.

To clarify how a signomial program is useful in our discussion, consider equation (20), and assume that  $\mathcal{Z}$  is an elliptical uncertainty set. Using the knowledge from

subsection A.2, (20) is equivalent to

$$\begin{aligned}
 (37) \quad & \sum_{k \in S_{i,j}} g_{i,j}^k v_{i,j}^k + s_{i,j}^{0.5} \leq e^{t_{ij}} & \forall (i,j) \in \mathbf{P} \\
 & \sum_{l=1}^L \sigma_l s_{i,j}^l \leq s_{i,j} & \forall (i,j) \in \mathbf{P} \\
 & \left( \sum_{k \in S_{i,j}} f_{i,j}^{k,l} v_{i,j}^k \right)^2 \leq s_{i,j}^l \quad \forall l \in 1, \dots, L & \forall (i,j) \in \mathbf{P}
 \end{aligned}$$

If one of the ‘ $f$ ’s is negative, then some constraints from the third set might not be GP-compatible, but SP-compatible. The robust geometric program will be a signomial program.

**Acknowledgments.** This research was supported by Boeing. We thank Professor Karen Willcox who provided insight, expertise, and important feedback that greatly assisted this work.

## REFERENCES

- [1] A Babakhani, J Lavaei, J C Doyle, and A Hajimiri. Finding globally optimum solutions in antenna optimization problems. *2010 IEEE Antennas and Propagation Society International Symposium*, 2010.
- [2] Chaithanya Bandi and Dimitris Bertsimas. Tractable stochastic analysis in high dimensions via robust optimization. *Mathematical Programming*, 134(1):2370, 2012.
- [3] A. Ben-Tal, Laurent El Ghaoui, and Nemirovskii Arkadii Semenovich. *Robust optimization*. Princeton University Press, 2009.
- [4] Dimitris Bertsimas and Melvyn Sim. Tractable approximations to robust conic optimization problems. *Springer-Verlag*, Dec 2005.
- [5] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to linear optimization*. Athena Scientific, 1997.
- [6] John R. Birge and F. Louveaux. *Introduction to stochastic programming*. Springer Science, 2011.
- [7] Stephen Boyd, Seung-Jean Kim, Lieven Vandenberghe, and Arash Hassibi. A tutorial on geometric programming. *Optimization and Engineering*, 8(1):67127, Oct 2007.
- [8] Stephen P. Boyd, Seung-Jean Kim, Dinesh D. Patil, and Mark A. Horowitz. Digital circuit optimization via geometric programming. *Operations Research*, 53(6):899932, 2005.
- [9] Michael J Burton and Warren W Hoburg. Solar-electric and gas powered, long-endurance uav sizing via geometric programming, June 2017.
- [10] Andr Chassein and Marc Goerigk. Robust geometric programming is co-np hard. Workingpaper jimportmodel: Workingpaperimportmodel, 12 2014.
- [11] Mung Chiang. *Geometric programming for communication systems*. Now Publishers, 2005.
- [12] M. del Mar Hershenson. Cmos analog circuit design via geometric programming. In *Proceedings of the 2004 American Control Conference*, volume 4, pages 3266–3271 vol.4, June 2004.
- [13] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076, 2013.
- [14] Harvey J. Greenberg. Mathematical programming models for environmental quality control. *Operations Research*, 43(4):578622, 1995.
- [15] Warren Hoburg and Pieter Abbeel. Geometric programming for aircraft design optimization. *AIAA Journal*, 52(11):24142426, 2014.
- [16] Warren Hoburg, Philippe Kirschen, and Pieter Abbeel. Data fitting with geometric-programming-compatible softmax functions. *Optimization and Engineering*, 17(4):897918, Apr 2016.
- [17] Warren Woodrow Hoburg. *Aircraft Design Optimization as a Geometric Program*. PhD thesis, University of California, Berkeley, 2013.
- [18] Kan-Lin Hsiung, Seung-Jean Kim, and Stephen Boyd. Tractable approximate robust geometric programming. *Optimization and Engineering*, 9(2):95118, Apr 2007.
- [19] K. O. Kortanek, Xiaojie Xu, and Yinyu Ye. An infeasible interior-point algorithm for solving primal and dual geometric programs. *Mathematical Programming*, 76(1):155181, 1997.
- [20] Costas D. Maranas and Christodoulos A. Floudas. Global optimization in generalized geometric programming. *Computers & Chemical Engineering*, 21(4):351 – 369, 1997.

- 552 [21] M. Mazumdar and T. R. Jefferson. Maximum likelihood estimates for multinomial probabilities  
553 via geometric programming. *Biometrika*, 70(1):257, 1983.
- 554 [22] Yurii Nesterov and Arkadii Nemirovskii. Interior-point polynomial algorithms in convex pro-  
555 gramming. 1994.
- 556 [23] Brendan ODonoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via  
557 operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory*  
558 *and Applications*, 169(3):10421068, 2016.
- 559 [24] Andras Prekopa. *Stochastic programming*. Kluwer Acad. Publ., 2010.
- 560 [25] A. L. Soyster. Technical noteconvex programming with set-inclusive constraints and applica-  
561 tions to inexact linear programming. *Operations Research*, 21(5):11541157, 1973.
- 562 [26] Thomas Wayne Wall, Doran Greening, and R. E. D. Woolsey. Or practicesolving complex  
563 chemical equilibria using a geometric-programming based technique. *Operations Research*,  
564 34(3):345355, 1986.