ROBUST APPROXIMATION OF CHANCE CONSTRAINED OPTIMIZATION WITH POLYNOMIAL PERTURBATION

BO RAO, LIU YANG, GUANGMING ZHOU, AND SUHAN ZHONG

ABSTRACT. This paper proposes a robust approximation method for solving chance constrained optimization (CCO) of polynomials. Assume the CCO is defined with an individual chance constraint that is affine in the decision variables. We construct a robust approximation by replacing the chance constraint with a robust constraint over an uncertainty set. When the objective function is linear or SOS-convex, the robust approximation can be equivalently transformed into linear conic optimization. Semidefinite relaxation algorithms are proposed to solve these linear conic transformations globally and their convergent properties are studied. We also introduce a heuristic method to find efficient uncertainty sets such that optimizers of the robust approximation are feasible to the original problem. Numerical experiments are given to show the efficiency of our method.

1. Introduction

Many real-world decision problems can be conveniently described by optimization with uncertainties. The chance constrained optimization (CCO) is a popular framework in stochastic programming. It aims at finding the best decisions such that random constraints are satisfied with a probability greater than or equal to a specified threshold. A CCO problem is

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{s.t.} & \quad \mathbb{P}\{\xi : h(x, \xi) \geq 0\} \geq 1 - \epsilon,
\end{align*}
\]

where \(x \in \mathbb{R}^n\) is the decision variable constrained in a set \(X\), \(\xi \in \mathbb{R}^r\) is the random vector with the probability distribution \(\mathbb{P}\) supported in a given set \(S \subseteq \mathbb{R}^r\). The \(f : \mathbb{R}^n \to \mathbb{R}\) is a function in \(x\), \(h : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^{m_0}\) is a scalar or vector-valued function, and \(\epsilon \in (0, 1)\) is a given risk level. In (1.1), we use \(\mathbb{P}\{\xi : h(x, \xi) \geq 0\}\) to denote the probability of \(h(x, \xi) \geq 0\) with respect to the distribution \(\mathbb{P}\). The corresponding random constraint is called the chance constraint. When \(h\) is a scalar function, (1.1) is called an individual chance constrained problem. Otherwise, it is called a joint chance constrained problem if \(m_0 > 1\). The (1.1) is said to have a polynomial perturbation if \(m_0 > 1\). This class of CCO problems has been studied in [1, 15, 25, 28, 34]. CCO is very difficult to solve due to the complicated structure of its feasible set. In this paper, we propose an efficient robust approximation approach to handle individual CCO problems with polynomial perturbation, under some convex assumptions in the decision variables.

2020 Mathematics Subject Classification. 90C15, 90C17, 90C22, 90C59.

Key words and phrases. Chance constrained optimization, Robust optimization, Nonnegative polynomial, Semidefinite relaxation.
Chance constrained optimization has broad applications in finance [44], emergency management [13], and energy management [10, 7]. Individual CCO problems are studied in [1, 5, 36, 9]. For a comprehensive introduction to the topic, we refer to monographs [45, 14] and references therein. It is very challenging to solve a general CCO problem. A major reason is that the feasible set usually does not have a convenient characterization in computations. Approximation approaches are often used to solve CCO problems. Scenario approximation is given in [8, 35]. Sample average approximation (SAA) is given in [11, 33, 44]. Condition value at risk (CVaR) approximation is given in [36, 47]. Bernstein approximation is given in [36]. DC approximation is given in [16, 22]. Smooth and nonsmooth approximations are given in [24, 9].

In this paper, we propose a robust approximation approach for individual CCO problems with polynomial perturbation. Assume \( f, h \) in (1.1) are scalar polynomials and that \( h \) is affine in \( x \). Let \( d \) denote the highest degree of \( h(x, \xi) \) in \( \xi \). One can use a scalar matrix \( A \) and vector \( b \) to represent
\[
(1.2) \quad h(x, \xi) = (Ax + b)^T[\xi]_d
\]
as a vector inner product, where \([\xi]_d := (1, \xi_1, \ldots, \xi_r, \xi_1^2, \ldots, \xi_r^d)^T\) is the monomial vector in \( \xi \) of the highest degree \( d \) ordered alphabetically. Since the chance constraint is nonlinear in \( \xi \), the feasible set of (1.1) is hard to characterize computationally. Instead, it is much simpler to find an uncertainty set \( U \subseteq \mathbb{R}^r \) such that for every \( x \in X \),
\[
h(x, \xi) \geq 0, \forall \xi \in U \quad \Rightarrow \quad \mathbb{P}\{\xi : h(x, \xi) \geq 0\} \geq 1 - \epsilon.
\]
This implies a robust approximation of (1.1) as
\[
(1.3) \quad \left\{ \begin{array}{l}
\min_{x \in X} f(x) \\
\text{s.t.} \quad h(x, \xi) \geq 0, \forall \xi \in U.
\end{array} \right.
\]
We refer to [2, 5, 6, 23, 31, 49] for relevant work on the robust approximation approach. Let \( \mathcal{P}(U) \) denote the set of polynomials in \( \xi \) that are nonnegative on \( U \). Then we can equivalently reformulate (1.3) into
\[
(1.4) \quad \left\{ \begin{array}{l}
\min_{x \in X} f(x) \\
\text{s.t.} \quad h(x, \xi) = (Ax + b)^T[\xi]_d \in \mathcal{P}(U).
\end{array} \right.
\]
In the above, the membership constraint means \( h(x, \xi) \), as a polynomial in \( \xi \), belongs to \( \mathcal{P}(U) \).

When \( f \) is affine in \( x \) and \( X \) has a semidefinite representation, the (1.4) is a linear conic optimization problem involving a Cartesian product of semidefinite and nonnegative polynomial cones. When \( f \) is a general polynomial, (1.4) can be further relaxed into linear conic optimization by introducing new variables to represent nonlinear monomials. Such a relaxation is tight if \( f(x) \) and \( X \) are given by SOS-convex polynomials \( ^1 \). Under the archimedeaness and some general assumptions, the transformed linear conic optimization can be solved globally by Moment-Sum-of-Squares (SOS) relaxations [37, 39].

---

\(^1\)A polynomial \( f \in \mathbb{R}[x] \) is said to be SOS-convex if there exists a matrix-polynomial \( F(x) \) such that its Hessian matrix \( \nabla^2 f = F(x)^T F(x) \).
For convenience, we assume the uncertainty set in (1.3) has the form

\[ U = \{ \xi \in \mathbb{R}^r : \Gamma - (\xi - \mu)^T \Lambda^{-1} (\xi - \mu) \geq 0 \}, \]

where \( \Gamma > 0 \) is a scalar, \( \mu \in \mathbb{R}^r \) is a real vector and \( \Lambda \in \mathbb{R}^{r \times r} \) is a positive definite matrix. In this paper, \( \mu \) and \( \Lambda \) are simply chosen as the mean value vector and covariance matrix of the random variable. In numerical experiments, they are either given directly or computed from samples. The scalar \( \Gamma \) is used to describe the size of \( U \). The selection of \( \Gamma \) can be tricky. It should be big enough such that the optimal solution of (1.3) is feasible to the original problem (1.1). Meanwhile, a relatively smaller \( \Gamma \) is usually preferred for an efficient approximation. In computations, we use a heuristic iterative method to find good \( \Gamma \). We first determine an initial value by the quantile estimation approach, and then update based on the probability of constraint violation of computed solutions. The quality of a robust approximation is usually dependent on the selection of uncertainty sets, see related works in [17, 18, 19, 30, 5]. Simple sets like polyhedrons and ellipsoids are often selected as uncertainty sets. The size of an uncertainty set can be determined by prior probability bounds [5, 31], by posterior probability bounds [5, 31], or by computed solutions [30].

**Contribution.** This paper focuses on the individual chance constrained optimization problem with polynomial perturbation. We find an efficient robust approximation of CCO by replacing the chance constraint with a robust constraint on an ellipsoidal uncertainty set. We propose semidefinite relaxation algorithms to solve the robust approximation globally. Analysis of convergent properties for these algorithms is carried out. In addition, a heuristic method is presented to find good uncertainty sets. Numerical experiments are given to show the efficiency of our approach. The main contributions of this paper are summarized as follows.

- We propose a robust approximation approach for individual chance constrained optimization with polynomial perturbation. The approximation is constructed by replacing the “hard” chance constraint with a “tractable” robust constraint over a convenient uncertainty set. Assume the chance constraint is linear in the decision variable. We show the approximation can be transformed into linear conic optimization with nonnegative polynomial cones. The transformation is equivalent under some convex assumptions.

- We propose efficient semidefinite relaxation algorithms to solve these linear conic transformations. Under some general assumptions, these algorithms can converge to the global optimizers of the robust approximations.

- In addition, we introduce a heuristic method to find good uncertainty sets in robust approximations. We determine a preferred size of uncertainty set based on sampling information and corresponding computed candidate solutions. Numerical experiments are presented to show the efficiency of our method.

The rest of this paper is organized as follows. Section 2 reviews some basics for polynomial and moment optimization. Section 3 presents an algorithm for solving the robust approximation problem with a linear objective. Section 4 studies the case where the objective function is an SOS-convex polynomial. Section 5 discusses the
construction of uncertainty sets. Section 4 performs some numerical experiments and an application. Conclusions are summarized in Section 7.

2. Preliminaries

**Notation.** The symbol $\mathbb{R}$ (resp., $\mathbb{R}^+, \mathbb{N}$) denotes the set of real numbers (resp., nonnegative real numbers, nonnegative integers). The superscript $^T$ denotes the transpose of a vector or matrix. The $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ denotes the $i$th unit vector in $\mathbb{R}^n$. For a vector $v \in \mathbb{R}^n$ and a scalar $R > 0$, $B(v, R)$ denotes the closed ball centered at $v$ with radius $R$. A symmetric matrix $W \in \mathbb{R}^{n \times n}$ is said to be positive semidefinite if $v^T W v \geq 0$ for all $v \in \mathbb{R}^n$, denoted as $W \succeq 0$. If $v^T W v > 0$ for all nonzero vectors $v$, then $W$ is said to be positive definite, denoted as $W > 0$. Let $\xi := (\xi_1, \cdots, \xi_r)$. The $\mathbb{R}[\xi]$ stands for the ring of polynomials in $\xi$ with real coefficients and $\mathbb{R}[\xi]_d \subseteq \mathbb{R}[\xi]$ contains all polynomials of degrees at most $d$. For a polynomial $p$, $\deg(p)$ denotes its degree. For a tuple $p := (p_1, \cdots, p_m)$ of polynomials, $\deg(p)$ denotes the highest degree of all $p_i$, i.e., $\deg(p) = \max\{\deg(p_1), \cdots, \deg(p_m)\}$. For a degree $d$, $[\xi]_d$ denotes the monomial vector with degrees up to $d$ and ordered alphabetically, i.e.,

$$\xi_d := (1 \quad \xi_1 \quad \cdots \quad \xi_r \quad \xi_1^2 \quad \xi_1 \xi_2 \quad \cdots \quad \xi_r^d)^T.$$  

Denote the power set $\mathbb{N}_d := \{\alpha := (\alpha_1, \cdots, \alpha_r) \in \mathbb{N}^r : \alpha_1 + \cdots + \alpha_r \leq d\}$. For $t \in \mathbb{R}$, $\lceil t \rceil$ denotes the smallest integer that is no less than $t$. For a set $X \subseteq \mathbb{R}^n$, $1_X(\cdot)$ denotes the characteristic function of $X$, i.e.,

$$1_X(x) := \begin{cases} 1, & \text{if } x \in X, \\ 0, & \text{if } x \notin X. \end{cases}$$

2.1. SOS and nonnegative polynomials. A polynomial $\sigma \in \mathbb{R}[\xi]$ is said to be *sum-of-squares* (SOS) if there are some polynomials $s_1, \cdots, s_k \in \mathbb{R}[\xi]$ such that $\sigma := s_1^2 + \cdots + s_k^2$. We use $\Sigma[\xi]$ to denote the set of all SOS polynomials in $\mathbb{R}[\xi]$ and denote $\Sigma[\xi]_d := \Sigma[\xi] \cap \mathbb{R}[\xi]_d$ for each degree $d$. For a polynomial $p \in \mathbb{R}[\xi]$, it is said to be *SOS-convex* if its Hessian matrix $\nabla^2 p$ is SOS, i.e., $\nabla^2 p = V(\xi)^T V(\xi)$ for a matrix-polynomial $V(\xi)$, and it is said to be *SOS-concave* if $-p$ is SOS-convex.

Let $g = (g_1, \ldots, g_m)$ be a tuple of polynomials. The quadratic module of $g$ is a polynomial cone defined as

$$QM[g] := \Sigma[\xi] + g_1 \cdot \Sigma[\xi] + \cdots + g_m \cdot \Sigma[\xi].$$

For an order $k$ such that $2k \geq \deg(g)$, the $k$th order truncated quadratic module of $g$ is given by

$$QM[g]_{2k} := \Sigma[\xi]_{2k} + g_1 \cdot \Sigma[\xi]_{2k-\deg(g_1)} + \cdots + g_m \cdot \Sigma[\xi]_{2k-\deg(g_m)}.$$  

For every $k$, we have the containment relation

$$QM[g]_{2k} \subseteq QM[g]_{2k+2} \subseteq QM[g].$$

Let $U := \{\xi \in \mathbb{R}^r \mid g(\xi) \geq 0\}$. Denote the nonnegative polynomial cone on $U$

$$\mathcal{P}(U) := \{p \in \mathbb{R}[\xi] : p(\xi) \geq 0, \forall \xi \in U\},$$

and its $d$th degree truncation $\mathcal{P}_d(U) = \mathcal{P}(U) \cap \mathbb{R}[\xi]_d$. Clearly, $QM[g]$ is a subset of $\mathcal{P}(U)$. If there exists $R > 0$ such that $R^2 - \|\xi\|_2^2 \in QM[g]$, then $U$ must be
compact and \( QM[g] \) is said to be archimedean. Under the archimedean assumption on \( QM[g] \), if a polynomial \( p \in \mathbb{R}[\xi] \) is positive on \( U \), then we have \( p \in QM[g] \). The conclusion is called Putinar’s Positivstellensatz \([46]\).

2.2. Moment and localizing matrices. Let \( \mathbb{R}^{N_{2k}} \) be the space of real vectors that are indexed by \( \alpha \in \mathbb{N}_{2k}^d \). A real vector \( y := (y_0)_{\alpha \in \mathbb{N}_{2k}^d} \) is called truncated multi-sequence (tms) of degree \( 2k \). It gives a Riesz functional \( \mathcal{L}_y \) acting on \( \mathbb{R}[^2\!{}_{2k}] \) as

\[
\mathcal{L}_y \left( \sum_{\alpha \in \mathbb{N}_{2k}^d} p_\alpha \xi^\alpha \right) := \sum_{\alpha \in \mathbb{N}_{2k}^d} p_\alpha y_\alpha.
\]

For \( p \in \mathbb{R}[^2\!{}_{2k}] \) and \( y \in \mathbb{R}^{N_{2k}} \), we write that

\[
\langle p, y \rangle := \mathcal{L}_y(p).
\]

For a degree \( d \), a tms \( y \in \mathbb{R}^{N_d} \) is said to admit a measure \( \mu \) supported in a set \( U \) if \( y_\alpha = \int_U \xi^\alpha d\mu \) for every \( \alpha \in \mathbb{N}_{d}^r \). Denote

\[
\mathcal{R}_d(U) := \{ y \in \mathbb{R}^{N_d} : y \text{ admits a measure supported in } U \}.
\]

When \( U \) is compact, \( \mathcal{R}_d(U) \) is a closed convex cone that is dual to \( \mathcal{P}_d(U) \), written as \( \mathcal{R}_d(U) = \mathcal{P}_d(U)^\ast \). Let \( g \in \mathbb{R}[\xi] \) with \( \text{deg}(g) \leq 2k \). The \( k \)th order localizing matrix of \( g \) and \( y \) is the symmetric matrix \( L^{(k)}_g(y) \) such that

\[
\text{vec}(a_1)^T L^{(k)}_g(y) \text{vec}(a_2) = \mathcal{L}_y(ga_1a_2)
\]

for all \( a_1, a_2 \in \mathbb{R}[\xi] \) with degrees at most \( k - \lceil \text{deg}(g)/2 \rceil \), where \( \text{vec}(a_1) \) denotes the vector of coefficients of \( a_1 \). When \( g = 1 \) is the constant polynomial, \( L^{(k)}_1(y) \) becomes the \( k \)th order moment matrix

\[
M_k[y] := L^{(k)}_1[y].
\]

For instance, when \( r = 2, k = 2 \) and \( g = 2 - \xi_1^2 - \xi_2^2 \),

\[
L^{(2)}_g[y] = \begin{bmatrix} 2y_{00} - y_{20} - y_{02} & 2y_{10} - y_{30} - y_{12} & 2y_{01} - y_{21} - y_{03} \\ 2y_{10} - y_{30} - y_{12} & 2y_{20} - y_{40} - y_{22} & 2y_{11} - y_{31} - y_{13} \\ 2y_{01} - y_{21} - y_{03} & 2y_{11} - y_{31} - y_{13} & 2y_{22} - y_{42} - y_{24} \end{bmatrix}.
\]

Moment and localizing matrices can be used to represent dual cones of truncated quadratic modules. For an order \( k \), define the tms cone

\[
\mathcal{S}[g]_{2k} := \{ y \in \mathbb{R}^{N_{2k}} : M_k[y] \succeq 0, L^{(k)}_g[y] \succeq 0 \}.
\]

The \( \mathcal{S}[g]_{2k} \) is closed convex for each \( k \). In particular, it is the dual cone of \( QM[g]_{2k} \) \([43]\) Theorem 2.5.2]. That is,

\[
\mathcal{S}[g]_{2k} = (QM[g]_{2k})^\ast := \left\{ y \in \mathbb{R}^{N_{2k}} : \langle p, y \rangle \geq 0 \ \forall p \in QM[g]_{2k} \right\}.
\]

In the above, the superscript \(^\ast\) is the notation for dual cone.
3. CCO WITH A LINEAR OBJECTIVE

Assume (1.1) is an individual CCO problem with polynomial perturbation as in (1.2). Consider the relatively simple case that \( f \) is a linear function, i.e., \( f(x) = c^T x \). The (1.1) becomes

\[
\begin{align*}
\min_{x \in X} & \quad c^T x \\
\text{s.t.} & \quad \mathbb{P}\{\xi : (Ax + b)^T [\xi]_d \geq 0\} \geq 1 - \epsilon,
\end{align*}
\]

where \( A \) is a scalar matrix, \( b, c \) are scalar vectors and \( [\xi]_d \) is the monomial vector as in (2.1). Let \( \mu \) denote the mean value and \( \Lambda \) denote the covariance matrix of \( \xi \).

For a proper size \( \Gamma > 0 \), define the uncertainty set

\[
U = \{\xi \in \mathbb{R}^n : \Gamma - (\xi - \mu)^T \Lambda^{-1}(\xi - \mu) \geq 0\}.
\]

We transform the chance constrain in (1.1) into the robust constraint

\[
(Ax + b)^T [\xi]_d \geq 0, \quad \forall \xi \in U.
\]

It can also be interpreted as \((Ax + b)^T [\xi]_d\), as a polynomial in \( \xi \), is nonnegative on \( U \). Recall that nonnegative polynomial cone \( \mathcal{P}(U) \) as in [22]. The robust approximation (1.3) can be written as

\[
\begin{align*}
f_{\text{min}} := \min_{x \in X} & \quad c^T x \\
\text{s.t.} & \quad (Ax + b)^T [\xi]_d \in \mathcal{P}_d(U),
\end{align*}
\]

where \( \mathcal{P}_d(U) \) is the \( d \)-th degree truncation of \( \mathcal{P}(U) \). The (3.3) is a linear conic optimization problem with a nonnegative polynomial cone. Since \( \mathcal{P}_d(U) \) is hard to characterize in computations, we consider solving (3.3) by semidefinite relaxations with quadratic modules.

### 3.1. A semidefinite relaxation algorithm

Let \( k_0 := \max \{\lfloor d/2 \rfloor, 1\} \) and denote

\[
g(\xi) := \Gamma - (\xi - \mu)^T \Lambda^{-1}(\xi - \mu).
\]

Clearly, \( QM[g] \) is archimedean since \( U \) is always compact for given \( \Gamma \). We can use truncated quadratic modules \( QM[g]_{2k} \) to give good approximations of \( \mathcal{P}(U) \).

Indeed, by [43] Proposition 8.2.1, we have

\[
\int(\mathcal{P}_d(U)) \subseteq \bigcup_{k=1}^{\infty} QM[g]_{2k} \cap \mathbb{R}[\xi]_d \subseteq \mathcal{P}_d(U).
\]

The \( k \)-th order SOS approximation of (3.3) is

\[
\begin{align*}
f_{k}^{\text{sos}} := \min_{x \in \mathbb{R}^n} & \quad c^T x \\
\text{s.t.} & \quad (Ax + b)^T [\xi]_d \in QM[g]_{2k} \cap \mathbb{R}[\xi]_d,
\end{align*}
\]

Recall the dual cone of \( QM[g]_{2k} \) is the tms cone \( \mathcal{S}(g)_{2k} \) as in (23). Let \( X^* \) denote the dual cone of \( X \). We can get the dual problem of (3.4):

\[
\begin{align*}
f_{k}^{\text{mom}} := \max_{(y,z)} & \quad -b^T y \\
\text{s.t.} & \quad c - A^T y \in X^*,
\end{align*}
\]

where

\[
y = z|_d := (z_\alpha)_{\alpha \in \mathbb{N}_0^n}, \quad z \in \mathcal{S}(g)_{2k}.
\]
Since \( U \) is compact, the \( \mathcal{R}_d(U) \) is closed. Then \( QM[g]_{2k} \subseteq \mathcal{P}(U) \) implies
\[
(3.6) \quad \mathcal{R}_d(U) := (\mathcal{P}_d(U))^* \subseteq (QM[g]_{2k} \cap \mathbb{R}[\xi]_d)^*.
\]
So we say (3.5) is the \( k \)th order relaxation of the dual problem of (3.3). Under the strong duality, the (3.3) can be solved by (3.4) if the relaxation (3.5) is tight.

**Algorithm 3.1.** For the robust approximation (3.3), let \( k = k_0 \). Then do the following:

**Step 1:** Solve (3.4)–(3.5) for optimizers \( x^* \) and \( (y^*, z^*) \) respectively.

**Step 2:** If (3.4)–(3.5) have no duality gap and there exists \( t \in [k_0,k] \) such that
\[
(3.7) \quad \text{rank } M_t[z^*] = \text{rank } M_{t-1}[z^*],
\]
then stop and output the optimizer \( x^* \) and the optimal value \( f^* = c^T x^* \). Otherwise, let \( k = k + 1 \) and go to Step 1.

When \( X, X^* \) have semidefinite representations, the primal-dual pair (3.4)–(3.5) are semidefinite programs. They can be solved globally with efficient computational methods like interior point methods. The rank condition in (3.7) is called flat truncation [40]. It is a sufficient condition for the primal-dual pair (3.4)–(3.5) is tight. The result is proved in the following theorem.

**Theorem 3.2.** Assume \( x^* \) and \( (y^*, z^*) \) are optimizers of (3.4)–(3.5) respectively at some order \( k \). If \( y^* \in \mathcal{R}_d(U) \) and there is no duality gap between (3.4)–(3.5), i.e., \( f^{\text{pos}}_k = f^{\text{mom}}_k \), then \( x^* \) is also a minimizer for (3.3).

**Proof.** Since \( U \) in (3.2) is compact, \( \mathcal{R}_d(U) = \mathcal{P}_d(U)^* \). The dual problem of (3.3) is
\[
(3.8) \quad (D) : \begin{cases} 
\max_y & -b^T y \\
\text{s.t.} & c - A^T y \in X^*, \\
y & \in \mathcal{R}_d(U).
\end{cases}
\]
Let \( f^{\text{max}} \) denote the optimal value of (D). Since (3.5) is a relaxation of (D), we have \( f^{\text{max}} \leq f^{\text{max}}_k \) for each \( k \). If the optimizer \( y^* \in \mathcal{R}_d(U) \) at some order \( k \), then \( y^* \) is also feasible for (D), thus
\[
f^{\text{mom}}_k = -b^T y^* \leq f^{\text{max}}_k.
\]
It implies that \( f^{\text{mom}}_k = f^{\text{max}} \) and that \( y^* \) is the optimizer of (D). Let \( f^{\text{min}} \) denote the optimal value of (3.3). By the weak duality between (3.3) and (D), we further have
\[
f^{\text{mom}}_k = f^{\text{max}} \leq f^{\text{min}} \leq f^{\text{pos}}_k.
\]
Suppose \( f^{\text{pos}}_k = f^{\text{mom}}_k \). The above inequality implies
\[
-b^T y^* = f^{\text{min}} = f^{\text{pos}}_k = c^T x^*.
\]
The \( x^* \) is feasible for (3.3). So it is also a minimizer of (3.3).  \( \square \)
The above theorem guarantees Algorithm 3.1 returns the correct optimizer and optimal value of (3.3) if it terminates in finite loops.

**Corollary 3.3.** If Algorithm 3.1 terminates in finite loops, then the output \( x^* \) and \( f^* \) are the correct optimizer and the optimal value of (3.3).

**Proof.** Let \( k \) be the order in the terminating loop. The termination criteria requires \( f_{k}^{\text{sos}} = f_{k}^{\text{mom}} \) and the rank condition (3.7) holds. By (3.7), the truncated tms \( z^*|_t \) admits a measure supported in \( U \). Since \( t \geq k_0 \) and \( y^* = z^*|_{k_0} \), we must have \( y^* \in \mathcal{A}_d(U) \). Then the conclusion is implied by Theorem 3.2. \( \square \)

### 3.2. Convergent properties

We study convergent properties of Algorithm 3.1. Major results from [37, 39] are used in our discussions, which include insightful analysis of linear optimization with moment and nonnegative polynomial cones.

We first discuss the asymptotic convergence of Algorithm 3.1. Recall that \( h(x, \xi) = (Ax + b)^T \xi |_d \). For convenience, we denote the coefficient vector

\[
(3.9) \quad h(x) := Ax + b.
\]

The optimization problem (3.3) is said to be strictly feasible if there exists \( \hat{x} \in \text{int}(X) \) such that \( h(\hat{x}, \xi) > 0 \) for all \( \xi \in U \). At the relaxation order \( k \), let \((y^{(k)}, z^{(k)})\) denote the optimizer of (3.5) and let \( f_k^{\text{sos}}, f_k^{\text{mom}} \) denote the optimal values of (3.4)–(3.5) respectively.

**Theorem 3.4.** Assume \( X \) is a convex set with a semidefinite representation. Suppose (3.3) is strictly feasible and its dual problem is feasible. Then for all \( k \) sufficiently large, \((y^{(k)}, z^{(k)})\) exists and that

\[
f_k^{\text{sos}} = f_k^{\text{mom}} \to f_{\text{min}} \quad \text{as} \quad k \to \infty.
\]

**Proof.** Suppose \( \hat{x} \) is a strictly feasible point of (3.3). Then there exists \( \epsilon_0 > 0 \) such that \( h(\hat{x}, \xi) > \epsilon_0 \) on \( U \). Since \( U \) is compact, there exists \( \delta > 0 \) sufficiently small such that \( B(\hat{x}, \delta) \subseteq X \) and

\[
\forall x \in B(\hat{x}, \delta) : \quad h(x, \xi) > \epsilon_0 \quad \text{on} \quad U \quad \text{as} \quad \text{a polynomial in} \quad \xi.
\]

Since \( QM[g] \) is archimedean, by [12, Theorem 6], there exists \( N_0 > 0 \) such that \( h(x, \xi) \in QM[g]_{2k} \) for every \( x \in B(\hat{x}, \delta) \) and for all \( k \geq N_0 \). It implies that \( \hat{x} \) is also a strictly feasible point of (3.4) for all \( k \geq N_0 \). Given the dual problem of (3.3) is feasible, its relaxation (3.5) is also feasible. Hence for all \( k \) big enough, there is a strong duality between (3.4)–(3.5) and (3.3) is solvable with an optimizer \((y^{(k)}, z^{(k)})\).

For an arbitrarily given \( \epsilon_1 \in (0, 1) \), there exists a feasible point \( \bar{x} \) of (3.3) such that

\[
f_{\text{min}} \leq c^T \bar{x} < f_{\text{min}} + \epsilon_1.
\]

Denote \( \bar{x} := (1 - \epsilon_1)\bar{x} + \epsilon_1 \hat{x} \).

It is strictly feasible for (3.3) (see as in [3, Proposition 1.3.1]). By previous arguments, we also have \( \bar{x} \) being feasible for (3.3) when \( k \) is large enough. In this case,

\[
f_k^{\text{sos}} \leq c^T \bar{x} = (1 - \epsilon_1)c^T \bar{x} + \epsilon_1 c^T \hat{x}
\]

\[
< (1 - \epsilon_1)(f_{\text{min}} + \epsilon_1) + \epsilon_1 c^T \hat{x}.
\]
Note $f_{k+1}^{\min} \leq f_{k+1}^{sos} \leq f_k^{sos}$ for all $k$. Since $\epsilon_1$ can be arbitrarily small, we can conclude that $f_k^{sos} \rightarrow f_{\min}$ as $k \rightarrow \infty$.

The above conclusion still holds if $X$ is a convex set determined by SOS-concave polynomials. We refer to Section 4 for more details. In addition, if we make some further assumptions on the optimizer(s) of (3.3), the finite convergence of Algorithm 3.1 can be reached.

**Theorem 3.5.** Under all the assumptions of Theorem 3.4, suppose (3.3) is solvable with an optimizer $x^*$ such that $h(x^*, \cdot) \in QM[g]_{2k}$ for all $k > k_1$, and

$$
(3.10) \begin{cases}
\min_{\xi \in \mathbb{R}^n} & h(x^*, \xi) = (Ax^* + b)^T \xi \\
\text{s.t.} & g(\xi) \geq 0
\end{cases}
$$

has only finitely many critical points $\zeta$ satisfying $h(x^*, \zeta) = 0$. Then Algorithm 3.1 will terminate in finite loops.

**Proof.** Under given conditions, the optimization problem (3.3) and its dual (3.8) are both solvable with optimizers $x^*$ and $y^*$ respectively. The strong duality holds that

$$0 = -b^Ty^* - c^Tx^* = -(Ax^* + b)^Ty^* - (c - A^Ty^*)^Tx^*.$$ 

Since $c - A^Ty^* \in X^*$, $h(x^*, \cdot) \in \mathcal{P}_d(U)$ and $y^* \in \mathcal{P}_d(U)$, we must have

$$(Ax^* + b)^Ty^* = 0, \quad (c - A^Ty^*)^Tx^* = 0.$$ 

It implies that (3.10) has the optimal value 0, which is achieved at every point in the support of the measure admitted by $y^*$. The (3.10) is a polynomial optimization problem. Its $k$th order Moment-SOS relaxation pair is

$$\gamma_k := \max \gamma \quad \text{s.t.} \quad h(x^*, \xi) - \gamma \in QM[g]_{2k},$$

$$\gamma_k^* := \min (Ax^* + b)^Tv \quad \text{s.t.} \quad v_0 = 1, \quad v \in \mathcal{P}[g]_{2k},$$

where $\mathcal{P}[g]_{2k}$ is given in (3.8). Suppose $h(x^*, \cdot) \in QM[g]_{2k}$ for all $k \geq k_1$ and there are only finitely many critical points $\zeta$ of (3.10) such that $h(x^*, \zeta) = 0$. For all $k$ that is large enough, the relaxation pair (3.11)–(3.12) is tight, i.e., $\gamma_k = \gamma_k^* = 0$, and every minimizer of (3.12) has a flat truncation [40, Theorem 2.2].

Let $k \geq k_1$ be sufficiently large. The (3.5) is solvable with the optimizer $(y^{(k)}, z^{(k)})$. If $(z^{(k)})_0 > 0$, then $z^{(k)}/(z^{(k)}_0)$ is feasible for (3.12) and it satisfies

$$(Ax^* + b)^T(\frac{z^{(k)}}{z^{(k)}_0}) = 0.$$ 

This is because $x^*$ is also a minimizer of (3.4) and there is no duality gap between (3.4)–(3.5). The above implies $z^{(k)}/(z^{(k)}_0)$ is an optimizer (3.12), thus it has a flat truncation and satisfies the rank condition (3.7). If $(z^{(k)})_0 = 0$, then $\text{vec}(1)^TM_k[z^{(k)}]\text{vec}(1) = 0$, where $\text{vec}(1)$ denotes the coefficient vector of the scalar polynomial that is constantly one. Since $M_k[z^{(k)}] \succeq 0$, we must have $M_k[z^{(k)}]\text{vec}(1) = 0$. By [29, Lemma 5.7], $M_k[z^{(k)}]\text{vec}(\xi^n) = 0$ for all $|n| \leq k - 1$. Then for every $\alpha = \beta + \eta$ with $|\beta|, |\eta| \leq k - 1$,

$$(z^{(k)})_\alpha = \text{vec}(\xi^\beta)^TM_k[z^{(k)}]\text{vec}(\xi^n) = 0.$$
It implies $z^{(k)}_{2k-2}$ is a zero vector, which is naturally flat. Therefore, the rank condition (3.7) holds for $k$ sufficiently large. \qed

## 4. CCO with SOS-Convexity

In this section, we study a more general kind of individual CCO with polynomial perturbation. In (1.1), assume $f(x)$ is an SOS-convex polynomial and the constraining set

$$X = \{ x \in \mathbb{R}^n : u_1(x) \geq 0, \ldots, u_{m_1}(x) \geq 0 \},$$

where each $u_i$ is an SOS-concave polynomial. Let $U$ be a selected uncertainty set as in (3.2). We repeat the robust approximation (1.3) as

$$u_i(x) \geq 0, \quad \text{for each } i = 1, \ldots, m_1.$$

When $f$, $u_i$ are nonlinear, Algorithm 3.1 cannot be directly applied to solve (1.1). Interestingly, we can always relax (4.1) into a linear conic optimization problem with nonnegative polynomial cones and semidefinite constraints. Let

$$d_1 := \left[ \frac{1}{2} \max \{ \deg(f), \deg(u_1), \ldots, \deg(u_{m_1}) \} \right].$$

We can relax (4.1) into the following problem:

$$\begin{align*}
\min_{(x,w)} & \quad \langle f, w \rangle \\
\text{s.t.} & \quad (Ax + b)^T [x] \in \mathcal{P}_d(U), \\
& \quad \langle u_i, w \rangle \geq 0, \quad i = 1, \ldots, m_1, \\
& \quad M_{d_1}[w] \succeq 0, \quad w_0 = 1, \\
& \quad x = \pi(w), \quad x \in \mathbb{R}^n, \quad w \in \mathbb{R}^{N_{d_1}},
\end{align*}$$

where $\pi : \mathbb{R}^{N_{d_1}} \to \mathbb{R}^n$ is the projection map defined by

$$\pi(w) := (w_{e_1}, \ldots, w_{e_n}), \quad w \in \mathbb{R}^{N_{d_1}}.$$

The relaxation is tight under SOS-convex assumptions.

**Theorem 4.1.** Suppose $f, -u_1, \ldots, -u_{m_1}$ are all SOS-convex polynomials. The optimization problems (4.1)–(4.2) are equivalent in the sense that they have the same optimal value and a feasible point $(x^*, w^*)$ is a minimizer of (4.2) if and only if $x^* = \pi(w^*)$ is a minimizer of (4.1).

**Proof.** Let $f_0, f_1$ denote the optimal values of (4.1)–(4.2) respectively. The $f_1 \leq f_0$ since (4.2) is a relaxation of (4.1). Suppose $x^* \in \mathbb{R}^n$ is a feasible point of (4.1). Since each $-u_i$ is SOS-convex, by Jensen’s inequality [27], we have

$$-u_i(x^*) = -u_i(\pi(w^*)) \leq \langle -u_i, w^* \rangle \leq 0.$$

So $\hat{x} = \pi(w^*)$ must also be feasible for (4.1). Also, since $f$ is SOS-convex, we have

$$f_0 \leq f(\hat{x}) = f(\pi(w^*)) \leq \langle f, \hat{w} \rangle.$$

Then $f_0 \leq f_1$, and since the above inequality holds for every feasible point of (4.2). So $f_0 = f_1$ and $x^* = \pi(w^*)$ is a minimizer of (4.1) if $(x^*, w^*)$ is an optimizer of (4.2). Conversely, if $x^*$ is a minimizer of (4.1), then $w^* = [x^*]_{2d_1} \in \text{feasible set for (4.2)}$, and $(x^*, w^*)$ is an optimizer of (4.2). \qed
As in Subsection 3.1 we construct the kth order SOS approximation of (4.2) with the truncated quadratic module of $g$ as follows

$$
\min_{(x, w)} \langle f, w \rangle \\
\text{s.t.} \quad (Ax + b)^T \xi \in QM[g]\{2k, \\
\langle u_i, w \rangle \geq 0, i = 1, \ldots, m_1, \\
M_{d_1}[w] \succeq 0, \quad w_0 = 1, \\
x = \pi(w), \quad x \in \mathbb{R}^n, \quad w \in \mathbb{R}^{N_{2d_1}}.
$$

Then we consider the dual problem of (4.2). Let $y \in \mathcal{R}_d(U)$, $\tau \in \mathbb{R}$, $\lambda = (\lambda_1, \ldots, \lambda_{m_1}) \in \mathbb{R}^{m_1}$ and $Q \succeq 0$. The Lagrange function of (4.2) is

$$
\mathcal{L}(w; y, \tau, \lambda, Q) = \langle f, w \rangle - \langle \pi(w) + b, y \rangle - \tau(w_0 - 1)
$$

Denote $q(x) = f(x) - y^T Ax - \lambda^T u(x) - \tau$. One can simplify

$$
\mathcal{L}(w; y, \tau, \lambda, Q) = \langle q, w \rangle - \langle M_{d_1}[w], Q \rangle + \tau - \langle b, y \rangle.
$$

For $\mathcal{L}$ to be bounded from below for all tms $w$, we must have $\langle q, w \rangle = \langle M_{d_1}[w], Q \rangle$. Note that $M_{d_1}[w]$ can be decomposed as

$$
M_{d_1}[w] = \sum_{\alpha \in \mathbb{N}^{2d_1}} w_\alpha C_\alpha,
$$

where each $C_\alpha$ is a symmetric matrix such that $[x]_{d_1}[x]_{d_1}^T := \sum_{\alpha \in \mathbb{N}^{2d_1}} x^\alpha C_\alpha$. Then $\langle q, w \rangle = \langle M_{d_1}[w], Q \rangle$ can be reformulated as

$$
q(x) = \left\langle \sum_{\alpha \in \mathbb{N}^{2d_1}} x^\alpha C_\alpha, Q \right\rangle = [x]_{d_1}^T Q [x]_{d_1}, \quad \forall x \in \mathbb{R}^n.
$$

Let $u = (u_1, \ldots, u_{m_1})$. Since $Q \succeq 0$, the above implies that $q(x)$ is an SOS polynomial in $x$, i.e.,

$$
q(x) = f(x) - y^T Ax - \lambda^T u(x) - \tau \in \Sigma[x]_{2d_1}.
$$

Then the dual problem of (4.2) is

$$
\max_{(\tau, \lambda, y)} \quad \tau - \langle b, y \rangle \\
\text{s.t.} \quad f(x) - y^T Ax - \lambda^T u(x) - \tau \in \Sigma[x]_{2d_1}, \\
y \in \mathcal{R}_d(U), \quad \lambda \in \mathbb{R}^{m_1}, \quad \tau \in \mathbb{R}.
$$

The dual problem of (4.3) is the kth order relaxation of the above problem, written as

$$
\max_{(\tau, \lambda, y, z)} \quad \tau - \langle b, y \rangle \\
\text{s.t.} \quad f(x) - y^T Ax - \lambda^T u(x) - \tau \in \Sigma[x]_{2d_1}, \\
y = z|_{d_1}, \quad z \in \mathcal{R}[g]_{2k}, \quad \tau \in \mathbb{R}, \quad \lambda \in \mathbb{R}^{m_1}.
$$

Since $\Sigma[x]_{2d_1}$ is semidefinite representable, (4.3) and its dual (4.5) can be solved effectively by some semidefinite programming techniques. Then we summarize an algorithm for solving the robust approximation (4.2) defined with SOS-convex polynomial.
Algorithm 4.2. For the robust approximation (3.3), let $k = k_0$. Then do the following:

**Step 1:** Solve (4.3) for a minimizer $(x^*, w^*)$ and solve (4.5) for a maximizer $(\tau^*, \lambda^*, y^*, z^*)$.

**Step 2:** If (4.3) and (4.5) have no duality gap and there exists an integer $t \in [k_0, k]$ such that (3.7) holds, then stop and output the optimizer $x^* = \pi(w^*)$ and the optimal value $f(x^*)$. Otherwise, let $k = k + 1$ and go to Step 2.

By Theorem 4.1, the output optimizer $x^*$ in Step 2 is also a minimizer of (4.1). Since (4.1) is equivalent to the linear conic optimization problem (4.2), the convergent properties of Algorithm 4.2 can be proved with similar arguments as in Subsection 3.2.

5. Size of uncertainty set

In this section, we discuss how to choose the size of the uncertainty set $U$. For convenience, denote the polynomial

\begin{equation}
\Gamma(\xi) := (\xi - \mu)^T \Lambda^{-1}(\xi - \mu).
\end{equation}

Then we have $U = \{\xi \in \mathbb{R}^r \mid \Gamma(\xi) \leq \Gamma\}$ and (1.3) can be expressed as

$$
(P_\Gamma) : \min_{x \in X} f(x) \quad \text{s.t.} \quad h(x, \xi) \geq 0, \quad \text{if } \Gamma(\xi) \leq \Gamma.
$$

Clearly, $(P_\Gamma)$ has a smaller feasible set with bigger $\Gamma > 0$. We want to find a small $\Gamma$ such that the optimizer of $(P_\Gamma)$ is feasible for the original CCO. Assume the optimizer of (1.1) is active at the chance constraint. We introduce a heuristic method to compute the preferred set size.

Define the $(1 - \epsilon)$-quantile of $\Gamma(\xi)$ as

\begin{equation}
\hat{\Gamma} := \inf \{\Gamma : \mathbb{P}\{\xi : \Gamma(\xi) \leq \Gamma\} \geq 1 - \epsilon\}.
\end{equation}

When $\Gamma \geq \hat{\Gamma}$, every feasible point of $(P_\Gamma)$ must also be feasible for (1.1). So $\hat{\Gamma}$ gives an upper bound for the set size that we want. However, it is very difficult to solve $\hat{\Gamma}$ analytically. In computations, it can be estimated using the quantile estimation method introduced by Hong et al in [23]. Suppose $\xi^{(1)}, \ldots, \xi^{(N)}$ are independent and identically distributed (i.i.d.) samples of $\xi$. Up to proper reordering, we may have

\begin{equation}
\Gamma(\xi^{(1)}) \leq \Gamma(\xi^{(2)}) \leq \cdots \leq \Gamma(\xi^{(N)}).
\end{equation}

**Theorem 5.1.** [23, Lemma 3] Assume $\xi^{(1)}, \ldots, \xi^{(N)}$ are i.i.d. samples of $\xi$ satisfying (3.3). For given $\epsilon \in (0, 1)$ and $\beta \in [0, 1)$, the $\Gamma(L^*)$ gives an upper bound on the $(1 - \epsilon)$-quantile of $\Gamma(\xi)$ with at least $(1 - \beta)$-confidence level, where

\begin{equation}
L^* := \min \left\{L \in \mathbb{N} : \sum_{i=0}^{L-1} \binom{N}{i} (1 - \epsilon)^i \epsilon^{N-i} \geq 1 - \beta, \ L \leq N \right\}.
\end{equation}

If (5.4) is unsolvable, then none of $\Gamma(\xi^{(i)})$ is a valid confidence upper bound.
Suppose \( x^\ast(\Gamma) \) is the optimizer of \((P_\Gamma)\). The probability \( \mathbb{P}\{\xi : h(x^\ast(\Gamma), \xi) < 0\} \) can be estimated by sample averages; i.e.,

\[
(5.5) \quad p_{\text{vio}}(x^\ast(\Gamma)) := \frac{1}{N} \sum_{i=1}^{\hat{N}} 1_{(-\infty,0)}(h(x^\ast(\Gamma), \xi^{(i)})),
\]

where \( 1_{(-\infty,0)}(\cdot) \) denotes the characteristic function of \((\infty,0)\) and \( \xi^{(1)}, \ldots, \xi^{(\hat{N})} \) are i.i.d. samples of \( \xi \).

Assume \( \hat{N} \) is big enough. Let \( \rho \in (0, \epsilon) \) be a selected small tolerance. Since the optimizer of \((1.1)\) is active at the chance-constraint, we want to determine a small \( \Gamma \) such that

\[
|p_{\text{vio}}(x^\ast(\Gamma)) - \epsilon| \leq \rho.
\]

If \( p_{\text{vio}}(x^\ast(\Gamma)) < \epsilon - \rho \), then \( x^\ast(\Gamma) \) is feasible to \((1.1)\) but it is expected to find a better robust approximation with a smaller \( \Gamma \). If \( p_{\text{vio}}(x^\ast(\Gamma)) > \epsilon + \rho \), then \( x^\ast(\Gamma) \) is not feasible for \((1.1)\) and we need to select a larger \( \Gamma \) such that the optimizer of \((1.3)\) is feasible to \((1.1)\). We can repeat this process for several times until a proper \( \Gamma \) is found such that \( p_{\text{vio}}(x^\ast(\Gamma)) \) is equal or close enough to \( \epsilon \). This idea was used in Li and Floudas [31, 49]. We summarize it as a heuristic algorithm as follows.

**Algorithm 5.2.** For the CCO problem \((1.1)\), choose small scalars \( \beta > 0, \rho > 0 \), select sample sizes \( N, \hat{N} (\hat{N} \gg N) \) and let \( \Gamma_L = 0 \) and \( l = 1 \). Then do the following

**Step 1:** Generate \( N + \hat{N} \) i.i.d. samples of \( \xi \) following the distribution \( \mathbb{P} \). Select \( N \) of them to be labeled with \( \xi^{(i)} \) and ordered by \((5.3)\). Denote the rest samples by \( \{\xi^{(i)}\}^{\hat{N}}_{i=1} \).

**Step 2:** Determine \( L^* \) in \((5.4)\) with \( \beta \). If \((5.4)\) is unsolvable, update \( \beta := 2\beta \). Set \( \Gamma_1 := \Gamma(\xi^{(L^*)}) \) and \( \Gamma_U := \Gamma(\xi^{(L^*)}) \). Then construct the uncertainty set \( U \) as in \((3.2)\) with set size \( \Gamma = \Gamma_l \).

**Step 3:** Solve \((1.3)\) for a minimizer \( x^\ast(\Gamma) \) and estimate the probability of constraint violation \( p_{\text{vio}}(x^\ast(\Gamma)) \) as in \((5.5)\). If \( |p_{\text{vio}}(x^\ast(\Gamma)) - \epsilon| \leq \rho \), then stop and output the set size \( \Gamma \). Otherwise, go to the next step.

**Step 4:** If \( p_{\text{vio}}(x^\ast(\Gamma)) < \epsilon \), update \( \Gamma_U := \Gamma \). Otherwise, update \( \Gamma_L := \Gamma \). Let \( l := l + 1 \), update \( \Gamma_l := \frac{1}{2}(\Gamma_L + \Gamma_U) \) and go back to Step 3.

### 6. Numerical experiments

In this section, we represent numerical experiments to show the efficiency of our method. The computations are implemented in MATLAB R2019a, running on a desktop computer with 4.0GB RAM and Intel(R) Core(TM) i3-4160 CPU using software Gloptipoly 3 [24], Yalmip [32], and SeDuMi [48]. All associated samples are generated by MATLAB commands following the specified distribution. For each example, we set the confidence level \( \beta = 0.05 \), the sample sizes \( N = 100 \) and \( \hat{N} = 10^6 \), and the tolerance parameter \( \rho = 10^{-6} \), unless stated otherwise. We use \( \Gamma^* \) to denote the terminated uncertainty set size in Algorithm 5.2. The \( f^* \) and \( x^* \) are used to represent the optimal value and optimizer of \((1.3)\) with the set size \( \Gamma^* \). For neatness, we only display four decimal digits for numerical results.
6.1. Efficiency of Algorithm 5.2 We give an example to show the efficiency of Algorithm 5.2.

Example 6.1. Consider the individual CCO problem

\[
\begin{align*}
\min_{x \in \mathbb{R}^3} & \quad f(x) = x_1 + x_2 + x_3 \\
\text{s.t.} & \quad \mathbb{P}\{\xi : h(x, \xi) \geq 0\} \geq 1 - \epsilon, \\
& \quad x_1 - 2x_2 + 2x_3 \geq 2,
\end{align*}
\]

where \(\xi\) follows multivariate Gaussian distribution with mean and covariance

\[
\mu = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} 2 & 1 & 0.5 \\ 1 & 2 & 0.4 \\ 0.5 & 0.4 & 3 \end{bmatrix},
\]

and the random constraining function

\[
h(x, \xi) = (3x_1 + 2x_2 + 2x_3)\xi_1^4 + (x_1 + 2x_2 + 2x_3 - 3)\xi_2^2\xi_3^2 + (x_1 - 2x_2)\xi_1^2\xi_2
\]
\[
+ (x_2 + 3x_3)\xi_2 + (3x_2 + x_3)\xi_3 + (2x_1 + 4x_2 + x_3).
\]

We make numerical experiments for different \(\epsilon, \beta\) and \(N\). For each pair \((\beta, N)\), we made 100 sampling instances.

(i). First, we explore the performance of the initial set size \(\Gamma(L^*)\) implied by Theorem 5.1. For each instance, we generate an uncertainty set with the initial size bound. The violation probability \(p_{vio}(U) := \mathbb{P}\{\xi : \xi \notin U\}\) can be evaluated analytically with Gaussian distributions. When \(p_{vio}(U) > \epsilon\), we say the instance is unsuitable. Denote \# as the number of unstable instances and

\[
p_{vio}(U) : \text{average of } p_{vio}(U), \quad p_{vio}^* : \text{standard deviation of } p_{vio}(U)
\]

among these 100 instances. We report the computational results in Table 1.

| \(\epsilon = 0.05\) | \(\epsilon = 0.01\) |
|---|---|
| \(\beta\) | \(N\) | \(p_{vio}(U)\) | \(p_{vio}^*(U)\) | \# | \(N\) | \(p_{vio}(U)\) | \(p_{vio}^*(U)\) | \# |
| 0.01 | 90 | 0.0103 | 0.0106 | 0 | 459 | 0.0023 | 0.0024 | 2 |
| | 500 | 0.0286 | 0.0075 | 1 | 1000 | 0.0031 | 0.0018 | 0 |
| | 1000 | 0.0348 | 0.0050 | 0 | 5000 | 0.0069 | 0.0011 | 0 |
| | 5000 | 0.0429 | 0.0030 | 3 | 10000 | 0.0080 | 0.0009 | 1 |
| | 10000 | 0.0451 | 0.0022 | 3 | | | | |
| 0.05 | 59 | 0.0166 | 0.0182 | 6 | 299 | 0.0032 | 0.0030 | 4 |
| | 500 | 0.0338 | 0.0091 | 5 | 500 | 0.0042 | 0.0028 | 4 |
| | 1000 | 0.0387 | 0.0052 | 3 | 1000 | 0.0053 | 0.0024 | 5 |
| | 5000 | 0.0451 | 0.0032 | 8 | 5000 | 0.0081 | 0.0013 | 8 |
| | 10000 | 0.0464 | 0.0019 | 3 | 10000 | 0.0086 | 0.0009 | 8 |

It is easy to observe that for both risk levels, \(p_{vio}(U)\) goes approaching \(\epsilon\) as \(N\) increases, while the number of unsuitable cases remains small. Indeed, even if the sample
size $N$ is small, the obtained uncertainty set is often suitable in our computational results.

(ii). Then, we apply Algorithm 5.2 to determine a better set size than $\Gamma(\xi(L^*))$ in (i). For each experiment, $p_{\text{vio}}(x^*(\Gamma))$ is estimated with $\tilde{N} = 10^6$ samples of $\xi$. Denote the optimal value of (1.3) by $f^I$ and $f^T$ respectively for the initial set size and the terminating set size. We write

$$\bar{f}_I : \text{average of } f_I, \quad f^I: \text{standard deviation of } f_I$$

and $\bar{f}_T, f^T$ similarly. The computational results are reported in Table 2. By applying Algorithm 5.2, the approximation quality is improved as the terminated optimal value average $\bar{f}_T$ is near half of the initial value $\bar{f}_I$. On the other hand, the difference among $f^T$ for different $(\beta, N)$ is tiny, which encourages us to use relatively smaller sample sizes to construct uncertainty sets.

### 6.2. Robust approximation of CCO.

We give some examples of individual CCO problems with linear objective functions. For a given uncertainty set, we apply Algorithm 3.1 to solve the robust approximation (1.3).

| $\epsilon$ | $\beta$ | $N$ | $\bar{f}_I$ | $f^I$ | $\bar{f}_T$ | $f^T$ |
|---|---|---|---|---|---|---|
| 0.01 | 500 | 2.4227 | 0.0211 | 1.2843 | 5.1423 $\cdot 10^{-4}$ |
| 0.05 | 500 | 2.4090 | 0.0224 | 1.2843 | 5.0555 $\cdot 10^{-4}$ |
| 0.01 | 5000 | 2.3813 | 0.0048 | 1.2844 | 4.8448 $\cdot 10^{-4}$ |
| 0.05 | 5000 | 2.3785 | 0.0041 | 1.2844 | 4.7893 $\cdot 10^{-4}$ |
| 0.01 | 50000 | 2.3546 | 0.0023 | 1.2845 | 4.6700 $\cdot 10^{-4}$ |
| 0.05 | 50000 | 2.3510 | 0.0024 | 1.2845 | 4.6144 $\cdot 10^{-5}$ |
| 0.01 | 100000 | 2.3504 | 0.0019 | 1.2846 | 4.5902 $\cdot 10^{-5}$ |
Example 6.2. Consider the individual CCO problem
\[
\min_{x \in \mathbb{R}^3} f(x) = 2x_1 + 3x_2 + x_3 \\
\text{s.t.} \quad \mathbb{P}\{\xi : h(x, \xi) \geq 0\} \geq 0.75, \\
\quad 4 - x_1 - x_2 - x_3 \geq 0, \\
\quad 2 - x_1 + 2x_2 - x_3 \geq 0,
\]
where the random constraining function
\[
h(x, \xi) = (-3x_1 + 2x_2)\xi_1^4 + (x_1 + 3x_3 + 1)\xi_2^4 + (-3x_2 + 2x_3 + 3)\xi_1^2\xi_2 \\
\quad + (x_1 + 2x_3)\xi_2^2\xi_3 + (2x_1 + x_2 - 2x_3),
\]
and \(\xi_1, \xi_2\) and \(\xi_3\) are independent and identically distributed random variables with the uniform distribution on [0, 2]. The mean vector \(\mu\) and covariance matrix \(\Lambda\) of \(\xi\) are
\[
\mu = \mathbf{1}_{3 \times 1}, \quad \Lambda = \frac{1}{3} I_3,
\]
where \(\mathbf{1}\) denotes the matrix of all ones and \(I_n\) denotes the \(n\)-dimensional identity matrix. By Theorem 5.1, we computed the initial set size and the probability violation
\[
\Gamma_1 = 4.4388, \quad p_{\text{vio}}(x^*(\Gamma_1)) = 0.0012.
\]
It took 24.1242 seconds for Algorithm 5.2 to terminate after 21 loops, where Algorithm 3.1 stops at the initial loop in each inner iteration. In the terminating loop, we get
\[
\Gamma^* = 1.5387, \quad p_{\text{vio}}(x^*) = 0.2500, \\
f^* = -1.6382, \quad x^* = (0.0531, -0.4513, -0.1781)^T.
\]

Example 6.3. Consider the individual CCO problem
\[
\min_{x \in \mathbb{R}^4} f(x) = x_1 + 2x_2 + 3x_3 + x_4 \\
\text{s.t.} \quad \mathbb{P}\{\xi : h(x, \xi) \geq 0\} \geq 0.80, \\
\quad [8 + 3x_2 - 4x_4 \quad 5 + x_3 \quad 2x_2 - 3x_4 \\
\quad 5 + x_3 \quad 10 + 2x_2 \quad -x_1 - 3x_2 + 3x_3 \\
\quad 2x_2 - 3x_4 \quad -x_1 - 3x_2 + 3x_3 \quad 3 + 3x_1 + 8x_4] \succeq 0,
\]
where the constraining random function
\[
h(x, \xi) = (x_1 + 3x_3 + 3x_4)\xi_1^5 + (-x_2 + 3x_3 + 2x_4)\xi_2^5 + (x_1 + 3x_2 + 2x_4 + 2) \\
\xi_3^2\xi_4^2 + (2x_1 + 2x_2 + x_4 - 5)\xi_3\xi_4\xi_5 + (x_1 + x_2 + 2x_3 - x_4),
\]
and each \(\xi_i\) independently follows \(t\)-distribution with degrees of freedom \(\nu = 3\). The mean vector \(\mu\) and covariance matrix \(\Lambda\) of \(\xi\) are \((0\text{ is the zero matrix})\)
\[
\mu = \mathbf{0}_{5 \times 1}, \quad \Lambda = 3I_5.
\]
By Theorem 5.1, we computed the initial set size and the probability violation
\[
\Gamma_1 = 9.0544, \quad p_{\text{vio}}(x^*(\Gamma_1)) = 0.0155.
\]
It took 40.8692 seconds for Algorithm 5.2 to terminate after 23 loops, where Algorithm 3.1 stops at the initial loop in each inner iteration. In the terminating loop, we get
\[
\Gamma^* = 1.2693, \quad p_{\text{vio}}(x^*) = 0.2000, \\
f^* = 0.7784, \quad x^* = (5.1776, -2.0061, 0.4772, -1.8185)^T.
\]
Example 6.4. Consider the individual CCO problem
\[
\min_{x \in \mathbb{R}^3} f(x) = -2x_1 - 3x_2 + x_3 \\
\text{s.t. } \mathbb{P}\{\xi : h(x, \xi) \geq 0\} \geq 0.90, \\
2 - x_1 + 2x_2 - x_3 \geq 0, \\
1 - x_1^2 - x_2^2 - x_3^2 \geq 0,
\]
where the random constraining function
\[
h(x, \xi) = (3x_1 - 6x_2)\xi_1^3\xi_2^2 + (x_1 - x_3)\xi_1^2\xi_2 + (x_1 + 3)\xi_1^2
\]
\[+(x_3 + 2)\xi_2^2 + 3x_2\xi_1 - 4x_3\xi_2,
\]
and \(\xi_1\) and \(\xi_2\) are two independent random variables that follow the exponential distribution with parameter \(\lambda = 1\) and \(\bar{\lambda} = 2\), respectively. The mean vector \(\mu\) and the covariance matrix \(\Lambda\) of \(\xi\) are
\[
\mu = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}.
\]
By Theorem 5.1 we computed the initial set size and the probability violation
\[
\Gamma_1 = 5.3688, \quad p_{vio}(x^*(\Gamma_1)) = 0.0000.
\]
It took 18.0012 seconds for Algorithm 5.2 to terminate after 20 loops, where Algorithm 5.1 stops at the initial loop in each inner iteration. In the terminating loop, we get
\[
\Gamma^* = 1.2693, \quad p_{vio}(x^*) = 0.1000, \\
f^* = -3.5249, \quad x^* = (0.7656, 0.5576, -0.3208)^T.
\]

Example 6.5. Consider the individual CCO problem
\[
\min_{x \in \mathbb{R}^2} f(x) = x_1 + 2x_2 \\
\text{s.t. } \mathbb{P}\{\xi : h(x, \xi) \geq 0\} \geq 0.95, \\
3 + 2x_1 - x_2 \geq 0, \\
1 - x_1 + x_2 \geq 0,
\]
where the random constraining function
\[
h(x, \xi) = x_1\xi_1^4 + 3x_2\xi_1^4 + 2x_1\xi_1\xi_2 + (3x_1 - 3x_2)\xi_2^2
\]
\[+(x_2 + 3)\xi_1 + (-x_1 + x_2 - 2)\xi_2 + (3x_1 + 4x_2),
\]
and the random vector \(\xi\) follows an empirical probability measure supported by 1000 samples. Those samples were randomly generated from the standard Gaussian distribution. The mean \(\mu\) and covariance matrix \(\Lambda\) of \(\xi\) are
\[
\mu = \begin{bmatrix} 0.0676 \\ 0.0132 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} 0.9887 & -0.0057 \\ -0.0057 & 0.9848 \end{bmatrix}.
\]
By Theorem 5.1 we computed the initial set size and the probability violation with \(N = 1000\) samples
\[
\Gamma_1 = 6.4948, \quad p_{vio}(x^*(\Gamma_1)) = 0.0000.
\]
It took 3.0562 seconds for Algorithm 5.2 to terminate after 11 loops, where Algorithm 5.1 stops at the initial loop in each inner iteration. In the terminating loop, we get
\[
\Gamma^* = 0.7548, \quad p_{vio}(x^*) = 0.0500, \\
f^* = 1.0895, \quad x^* = (1.0298, 0.029)^T.
\]
The $x^*$ is the optimizer of $\{1, 2, 3\}$ where the uncertainty set $U$ has the size $\Gamma^*$. Interestingly, $U$ with the size $\Gamma^*$ only contains 303 samples of $\xi$, i.e.,

$$\mathbb{P}\{\xi : \xi \in U\} = 0.303 < 0.95 = \mathbb{P}\{h(x^*, \xi) \geq 0\}.$$  

The probability difference implies that $\mathbb{P}\{\xi : \xi \in U\} \geq 1 - \epsilon$ is sufficient but not necessary for the chance constraint in (1.1) to hold. In other words, there exists $U$ such that $\mathbb{P}\{\xi : \xi \in U\} < 1 - \epsilon$ but $\mathbb{P}\{h(x^*, \xi) \geq 0\} \geq 1 - \epsilon$ for the optimizer $x^*$ of (1.3).

Then we give some examples of individual CCO problems defined with SOS-convex polynomials. For a given uncertainty set, we apply Algorithm 4.2 to solve the robust approximation $\{1, 2, 3\}$.

**Example 6.6.** Consider the individual CCO problem

$$\begin{align*}
\min_{x \in \mathbb{R}^5} & \quad f(x) = 4x_1^4 + 6x_2^2 + x_3 + 3x_4 + x_5 \\
\text{s.t.} & \quad \mathbb{P}\{\xi : h(x, \xi) \geq 0\} \geq 0.90, \\
& \quad u_1(x) = 8 - x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_5^2 \geq 0, \\
& \quad u_2(x) = 10 - 3x_1^4 - 6x_2^2 - 2x_3^2 + 6x_4 - 3x_5 \geq 0,
\end{align*}$$

where the random constraining function

$$h(x, \xi) = (3x_1 + 2x_2 + 2x_4)\xi_1^4 + (x_2 - 2x_4 + 2x_5)\xi_1^2\xi_2^2 + (x_3 - 3x_4)\xi_1^2\xi_4 + (3x_2 - x_3 - 3x_5)\xi_3 + (2x_2 - 3x_5)\xi_4 + (2x_1 + 4x_2 + x_3 - 5x_4 - 10x_5),$$

and the random vector $\xi$ follows $t$-distribution with freedom degree, location and the scale matrix

$$\begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \end{bmatrix}, \quad \bar{\Lambda} = \begin{bmatrix} 4 & 2 & 0 & 1 \\ 2 & 3 & 0 & 1 \\ 0 & 0 & 2 & 3 \\ 1 & 1 & 3 & 6 \end{bmatrix}.$$  

Then the mean and covariance of $\xi$ are

$$\mu = \bar{\mu}, \quad \Lambda = \frac{\bar{\nu}}{\bar{\nu} - 2}\bar{\Lambda}.$$  

It is easy to verify that $f$, $-u_1$ and $-u_2$ are all SOS-convex polynomials. By Theorem 4.1, we computed the initial set size with $N = 300$ samples and the probability violation

$$\Gamma_1 = 48.6056, \quad p_{vio}(x^*(\Gamma_1)) = 0.0000.$$  

It took 21.2167 seconds for Algorithm 4.2 to terminate after 21 loops, where Algorithm 4.2 stops at the initial loop in each inner iteration. In the terminating loop, we get

$$\Gamma^* = 3.2416, \quad p_{vio}(x^*) = 0.0100, \quad f^* = -3.7496, \quad x^* = (0.5603, -0.0467, -1.3485, -0.7668, -0.5080)^T.$$  

**Example 6.7.** Consider the individual CCO problem

$$\begin{align*}
\min_{x \in \mathbb{R}^3} & \quad f(x) = 6x_1^4 + x_2^3 + 3x_4^2 + 5x_2 \\
\text{s.t.} & \quad \mathbb{P}\{\xi : h(x, \xi) \geq 0\} \geq 0.85, \\
& \quad u_1(x) = 11 - (x_1 + x_2)^4 - 2x_3^4 - (3x_3 - x_4)^2 + 5x_2 \geq 0, \\
& \quad u_2(x) = 6 - (x_2 - x_3)^2 - 3x_2x_3 + 4x_3 + 3x_1 \geq 0,
\end{align*}$$
where the random constraining function

\[ h(x, \xi) = (8x_1 + x_2 + 6x_3)\xi_1^4 + (x_3 - 2x_4)\xi_2^2\xi_4^2 + (x_2 + 3x_4)\xi_2\xi_3^2 + (x_1 - 2x_2 + 3x_3)\xi_1\xi_2 + (2x_1 + 3x_3 + 1)\xi_3\xi_4 + (8x_1 - 4x_2 - 2x_3), \]

and the random variables \( \xi_i, i = 1, 2, 3, 4 \) are independent to each other. Assume \( \xi_1 \) is a beta distribution with shape parameters \( \bar{\alpha} = \bar{\beta} = 2 \), \( \xi_2 \) is a gamma distribution with shape parameter \( \bar{k} = 2 \) and scale parameter \( \bar{\theta} = 1 \), \( \xi_3 \) is a chi-squared distribution with degrees of freedom 3, and \( \xi_4 \) is a chi-squared distribution with degrees of freedom 4. Then \( \xi \) has mean and covariance

\[
\begin{bmatrix}
\mu_1 \\
\mu_2 \\
\mu_3 \\
\mu_4 \\
\end{bmatrix}, \quad \Lambda =
\begin{bmatrix}
\frac{1}{2} & 0 & 0 & 0 \\
0 & 2 & 0 & 0 \\
0 & 0 & 6 & 0 \\
0 & 0 & 0 & 8 \\
\end{bmatrix}.
\]

It is easy to verify that \( f, -u_1, \) and \(-u_2\) are all SOS-convex polynomials. By Theorem 5.1, we computed the initial set size with \( N = 300 \) samples and the probability violation

\[ \Gamma_1 = 8.1934, \quad p_{vio}(x^*(\Gamma_1)) = 0.0000. \]

It took 36.2564 seconds for Algorithm 5.2 to terminate after 23 loops, where Algorithm 5.2 stops at the initial loop in each inner iteration. In the terminating loop, we get

\[
\Gamma^* = 0.2918, \quad p_{vio}(x^*) = 0.1500,
\]

\[ f^* = -8.2948, \quad x^* = (0.5030, -1.7362, 0.0185, -0.0240)^T. \]

Next, we give an example of individual CCO problems with an application background in portfolio selection problems.

**Example 6.8. (VaR Portfolio Optimization)** An investor intends to invest in 4 risky assets. The goal is to get the minimal loss level such that the probability of a bigger loss does not exceed a preferred risk level \( \epsilon \). Value-at-Risk (VaR) portfolio optimization can be used to model this problem. It has the form

\[
\begin{align*}
\min_{(x_0, x)} \quad & x_0 \\
\text{s.t.} \quad & \mathbb{P}\left\{ \xi : x_0 \geq \sum_{i=1}^{4} x_ir_i(\xi) \right\} \geq 1 - \epsilon, \\
& \sum_{i=1}^{4} x_i = 1, \quad x_0 \in \mathbb{R}, \\
& x = (x_1, x_2, x_3, x_4) \in \mathbb{R}^4_+, \\
\end{align*}
\]

where \( \epsilon \) is a preferred risk level, and each function \( r_i(\xi) \) stands for the return rate function of the \( i \)-th asset. Assume \( \xi = (\xi_1, \xi_2, \xi_3) \) and

\[
\begin{align*}
r_1(\xi) &= 0.5 + \xi_1^2 - \xi_2^2\xi_3^2 + \xi_4^4, \\
r_2(\xi) &= -1 + \xi_2^2 + \xi_4^4 - \xi_1^2\xi_3^2, \\
r_3(\xi) &= 0.8 + \xi_3^2 - \xi_1\xi_2 + \xi_4^4, \\
r_4(\xi) &= 0.5 + \xi_3 - \xi_1\xi_2\xi_3 + \xi_2^2\xi_3^2.
\end{align*}
\]

In this application, we consider \( \xi_1, \xi_2, \) and \( \xi_3 \) are independently distributed random variables. Suppose \( \xi_1 \) follows the beta distribution with shape parameters \( \bar{\alpha} = \bar{\beta} = 4 \), \( \xi_2 \) follows the log-normal distribution with parameters \( \bar{\mu} = 0, \bar{\sigma} = 1 \) and \( \xi_3 \)
follows the log-normal distribution with parameters \( \mu = -1, \sigma = 1 \). Then \( \xi \) has the mean vector and covariance matrix

\[
\mu = \begin{bmatrix} \frac{1}{2} \\ \sqrt{e} \end{bmatrix}, \quad \Lambda = \begin{bmatrix}
\frac{1}{2} & 0 & 0 \\
0 & e^2 - e & 0 \\
0 & 0 & 1 - \frac{1}{e}
\end{bmatrix}.
\]

For a given uncertainty set and for different risk levels, i.e., \( \epsilon \in \{0.05, 0.20, 0.35\} \), we apply Algorithm 3.1 to solve the robust approximation problem (1.3). The computational results are reported in Table 3, where \( \Gamma_1 \) denotes the initial set size implied by Theorem 5.2, \( f_I \) denotes the optimal value of (1.3) with the initial set size.

| \( \epsilon \) | 0.05   | 0.20   | 0.35   |
|-----------------|--------|--------|--------|
| \( \Gamma_1 \)  | 8.6725 | 3.9047 | 3.7130 |
| \( \Gamma^* \)  | 0.5703 | 0.3137 | 0.1191 |
| \( p_{vio}(x^*(\Gamma_1)) \) | \( 3.0600 \cdot 10^{-4} \) | 0.0011 | 0.0031 |
| \( p_{vio}(x^*) \) | 0.0500 | 0.2000 | 0.3500 |
| \( f_I \)       | -0.5340 | -0.5364 | -0.5365 |
| \( f^* \)       | -0.5598 | -0.6642 | -0.8127 |
| \( x^* \)       | \begin{bmatrix}
0.3909 \\
0.0751 \\
0.3515 \\
0.1826
\end{bmatrix}, \quad \begin{bmatrix}
0.1417 \\
0.0788 \\
5.3332 \cdot 10^{-9} \\
0.7795
\end{bmatrix}, \quad \begin{bmatrix}
3.3121 \cdot 10^{-9} \\
0.1523 \\
4.9504 \cdot 10^{-9} \\
0.8477
\end{bmatrix} \]
| Loop            | 18     | 20     | 21     |
| Time(seconds)    | 17.2824 | 18.6632 | 19.2371 |

7. Conclusion and Discussions

This paper focuses on individual CCO problems of polynomials with the chance constraint being affine in decision variables. A robust approximation method is proposed to transform such polynomially perturbed CCO problems into linear conic optimization with nonnegative polynomial cones. When the objective function is linear and the constraining set \( X \) has a semidefinite representation or the objective function is SOS-convex and defining functions for \( X \) are all SOS-concave, semidefinite relaxation algorithms, i.e., Algorithm 3.1 and Algorithm 4.2 are proposed to solve these robust approximations globally. Convergent analysis of these algorithms is carried out. In addition, discussions are made to construct efficient uncertainty sets of the robust approximation. A heuristic method is introduced to find good set sizes. Numerical examples, as well as an application for portfolio optimization, are given to show the efficiency of our approach.

There is promising future work for this paper. For instance, can we extend this method for joint CCO problems with polynomial perturbation? How do we solve more general polynomial individual CCO problems without convex assumptions?
How to analyze the gap between the robust approximation and the original problem? These questions motivate our ongoing exploration.

References

[1] Ahmed, S.: Convex relaxations of chance constrained optimization problems. *Optim. Lett.*, 8, 1—12 (2014).
[2] Ben-Tal, A., Nemirovski, A.: Robust solutions of linear programming problems contaminated with uncertain data. *Math. Program.*, 88, 411–424 (2000)
[3] Bertsekas, D.: *Convex Optimization Theory*. Athena Scientific, Nashua (2009)
[4] Bertsimas, D., Gupta, V., Kallus, N.: Data-driven robust optimization. *Math. Program.*, 167, 235–292 (2018)
[5] Bertsimas, D., Hertog, D., Pauphilet, J.: Probabilistic guarantees in robust optimization. *SIAM J. Optim.*, 31, 2893–292 (2021)
[6] Bertsimas, D., Sim. M.: The price of robustness. *Oper. Res.*, 52, 35–53 (2004)
[7] Bienstock, D., Chertkov, M., Harnett, S.: Chance-constrained optimal power flow: Risk-aware network control under uncertainty. *SIAM Rev.*, 56, 461–495 (2014)
[8] Calafiore, G.C., Campi, M.C.: The scenario approach to robust control design. *IEEE Trans. on Autom. Control*, 51, 742–753 (2006)
[9] Cao, Y., Victor, M.: A sigmoidal approximation for chance-constrained nonlinear programs. Preprint at https://arxiv.org/abs/2004.02402 (2020)
[10] Charnes, A., Cooper, W. W., Symonds, G.H.: Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Manag. Sci.* 4, 235–263 (1958)
[11] Cheng, J., Gicquel, C., Lisser, A.: Partial sample average approximation method for chance constrained problems. *Optim. Lett.*, 13, 657–672 (2019)
[12] Curto, R., Fialkow, L.: Truncated K-moment problems in several variables. *J. Operator Theory*, 54, 189-226 (2005)
[13] Deng, C., Yang, L.: Sample average approximation method for chance constrained stochastic programming in transportation model of emergency management. *Syst. Eng. Procedia.*, 5, 137–143 (2012)
[14] Dentecheva, D.: Optimization models with probabilistic constraints. In: Shapiro, A., Dentcheva, D., Ruszcyzyński, A. (eds) *Lectures on Stochastic Programming: Modeling and Theory*, Third Edition, pp. 81-149. SIAM, Philadelphia (2021)
[15] Feng, C., Dabbene, F., Lagoa, C. M.: A kinship function approach to robust and probabilistic optimization under polynomial uncertainty. *IEEE Trans. on Autom. Control*, 56, 1509–1523 (2011)
[16] Feng, C., Dabbene, F., Lagoa, C.M.: A smoothing function approach to joint chance-constrained programs. *J. Optim. Theory Appl.*, 163, 181-199 (2014)
[17] Guzman, Y., Matthews, L., Floudas, C.: New a priori and a posteriori probabilistic bounds for robust counterpart optimization: I. Unknown probability distributions. *Comput. Chem. Eng.*, 84, 568-598 (2016).
[18] Guzman, Y., Matthews, L., Floudas, C.: New a priori and a posteriori probabilistic bounds for robust counterpart optimization: II. A priori bounds for known symmetric and asymmetric probability distributions. *Comput. Chem. Eng.*, 101, 279–311 (2017).
[19] Guzman, Y., Matthews, L., Floudas, C.: New a priori and a posteriori probabilistic bounds for robust counterpart optimization: III. Exact and near-exact a posteriori expressions for known probability distributions. *Comput. Chem. Eng.*, 103,116–143 (2017).
[20] Helton, J.W., Nie, J.: Semidefinite representation of convex sets. *Math. Program.*, 122, 21-64 (2010)
[21] Henrion, D., Lasserre, J., Löfberg, J.: GloptiPoly 3: moments, optimization and semidefinite programming. *Optim. Method Softw.*, 24, 761-779 (2009)
[22] Hong, L., Yang, Y., Zhang, L.: Sequential convex approximations to joint chance constrained programs: A Monte Carlo approach. *Oper. Res.*, 59, 617-630 (2011)
[23] Hong, L., Huang, Z., Lam, H.: Learning-based robust optimization: Procedures and statistical guarantees. *Manag. Sci.*, 67, 3447-3467 (2021)
[24] Kannan, R., Luedtke, J.R.: A stochastic approximation method for approximating the efficient frontier of chance-constrained nonlinear programs. *Math. Prog. Comp.*, 13, 705-751 (2021)

[25] Lagoa, C.M., Li, X., Sznajer, M.: Probabilistically constrained linear programs and risk-adjusted controller design. *SIAM J. Optim.*, 15, 938–951 (2005).

[26] Lasserre, J.: Global optimization with polynomials and the problem of moments. *SIAM J. Optim.* 11, 796–817 (2001)

[27] Lasserre, J.: Convexity in semialgebraic geometry and polynomial optimization. *SIAM J. Optim.* 19, 1995-2014 (2009)

[28] Lasserre, J. B.: Representation of chance-constraints with strong asymptotic guarantees. *Control Syst. Lett.*, 1, 50-55 (2017).

[29] Laurent, M.: Sums of squares, moment matrices and optimization over polynomials. In: Putinar, M., Sullivant, S. (eds.) *Emerging Applications of Algebraic Geometry*, pp. 157-270. Springer, New York (2009)

[30] Li, Z., Tang, Q., Floudas, C.: A comparative theoretical and computational study on robust counterpart optimization: II. Probabilistic guarantees on constraint satisfaction. *Ind. Eng. Chem. Res.*, 51, 6769-6788 (2012)

[31] Li, Z., Tang, Q., Floudas, C.: A comparative theoretical and computational study on robust counterpart optimization: III. Improving the quality of robust solutions. *Ind. Eng. Chem. Res.* 53, 13112-13124 (2014)

[32] Lofberg, J.: YALMIP: a toolbox for modeling and optimization in MATLAB. *2004 IEEE International Conference on Robotics and Automation* (IEEE Cat. No.04CH37508). 284-289 (2005)

[33] Luedtke, J., Ahmed, S.: A sample approximation approach for optimization with probabilistic constraints. *SIAM J. Optim.*, 19, 674-699 (2008)

[34] Jasour, A. M.; Aybat, N. S.; Lagoa, C. M.: Semidefinite programming for chance constrained optimization over semialgebraic sets. *SIAM J. Optim.* 25, 1411—1440 (2015)

[35] Nemirovski, A., Shapiro, A.: Scenario approximations of chance constraints. In: Calafiore, G., Dabbene, F. (eds.) *Probabilistic and Randomized Methods for Design under Uncertainty*, pp. 3-47. Springer, London (2006)

[36] Nemirovski, A., Shapiro, A.: Convex approximations of chance constrained programs. *SIAM J. Optim.*, 17, 969-996 (2007)

[37] Nie, J.: The $A$-truncated $K$-moment problem. *Found. Comput. Math.*, 14, 1243-1276 (2014)

[38] Nie, J.: Optimality conditions and finite convergence of Lasserre’s hierarchy. *Math. Program.*, 146, 97-121 (2014)

[39] Nie, J.: Linear optimization with cones of moments and nonnegative polynomials. *Math. Program.*, 153, 247-274 (2015)

[40] Nie, J.: Certifying convergence of Lasserre’s hierarchy via flat truncation. *Math. Program.*, 142, 485-510 (2013)

[41] Nie, J., Ranestad, K.: Algebraic degree of polynomial optimization. *SIAM J. Optim.*, 20, 485-502 (2009)

[42] Nie, J., Schweighofer, M.: *On the complexity of Putinar’s Positivstellensatz*. *J. Complexity.*, 23, 135-150 (2007)

[43] Nie, J., *Moment and Polynomial Optimization*. SIAM, 2023.

[44] Pagnoncelli, B.K., Ahmed, S., Shapiro, A.: Sample average approximation method for chance constrained programming: theory and applications. *J. Optim. Theory Appl.*, 142, 399-416 (2009)

[45] Prékopa, A.: Probabilistic programming. In: Ruszczynski, A., Shapiro, A. (eds.) *Handbooks in Operations Research and Management Science*, vol. 10, pp. 267-351. Elsevier (2003)

[46] Putinar M.: Positive polynomials on compact semi-algebraic sets. *Indiana Univ. Math. J.*, 42, 969-984 (1993)

[47] Rockafellar, R., Uryasev, S.: Optimization of conditional value-at-risk. *Journal of risk* 2, 21-42 (2000)
[48] Sturm, J.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Method Softw.*, 11, 625-653 (1999)

[49] Yuan, Y., Li, Z., Huang, B.: Robust optimization approximation for joint chance constrained optimization problem. *J. Glob. Optim.*, 67, 805-827 (2017)

Bo Rao, School of Mathematics and Computational Sciences, Xiangtan University, Hunan, China

*Email address*: raob@smail.xtu.edu.cn

Liu Yang, Guangming Zhou, School of Mathematics and Computational Sciences, Xiangtan University, Hunan, China

Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan University, Hunan, China

*Email address*: yangl410@xtu.edu.cn, zhougm@xtu.edu.cn

Suhan Zhong, Department of Mathematics, Texas A&M University, College Station, Texas, USA, 77843

*Email address*: suzhong@tamu.edu