Combining SOS with Branch and Bound to Isolate Global Solutions of Polynomial Optimization Problems

Brendon Colbert, Hesameddin Mohammadi, Matthew M. Peet

***Abstract*— In this paper, we combine a branch and bound algorithm with SOS programming in order to obtain arbitrarily accurate solutions to Global Polynomial Optimization (GPO) problems with bounded feasible sets. These arbitrarily accurate solutions are then fed into local gradient descent algorithms to obtain the true global optimizer. The algorithm successively bisects the feasible set and uses SOS to compute a Greatest Lower Bound (GLB) over each feasible set. For any desired accuracy, ε, we prove that the algorithm will return a point x such that $|x-y| \leq \varepsilon$ for some point with objective value $|f(y)-f(x^*)| \leq \varepsilon$ where x^* is the global optimizer. To achieve this point, x, the algorithm sequentially solves $O(\log(1/\varepsilon))$ GLB problems, each of identical polynomial-time complexity. The point x, can then be used as an accurate initial value for gradient descent algorithms. We illustrate this approach using a numerical example with several local optima and demonstrate that the proposed algorithm dramatically increases the effectiveness of standard global optimization solvers.**

I. Introduction

Global Polynomial Optimization (GPO) is defined as optimization of the form

$$
\begin{aligned}
f^* := \min_{x\in\mathbb{R}^n}. \quad & f(x) && (1)\\
\text{subject to} \quad & g_i(x) \geq 0 \quad \text{for } i=1,\cdots,s\\
& h_j(x) = 0 \quad \text{for } j=1,\cdots,t,
\end{aligned}
$$

where f, g_i, and h_i are real-valued polynomials in decision variables x. As defined in Eq. 1, the GPO problem encompasses many well-studied sub-classes including Linear Programming (LP) [1], [2], Quadratic Programming (QP) [3], Integer Programming (IP), Semidefinite Programming (SDP) and Mixed-Integer Nonlinear Programming (MINLP) [4]. Because of its generalized form, almost any optimization problem can be cast or approximately cast as a GPO, including certain NP-hard problems from economic dispatch [5], optimal power flow [6] and optimal decentralized control [7]. As applied to control theory, GPO can be used for stability analysis of polynomial dynamical systems by, e.g., verifying polytopic invariants as in [8].

When f and g_i are convex and h_i are linear, the GPO problem is convex and may be solved using barrier functions and gradient descent. When the GPO problem is not convex, there also exist special cases which are solvable. For example, in [9] the unconstrained problem was solved using Groebner bases. In the special case of $x \in \mathbb{R}^1$, the problem was solved in [10], [11]. In addition, there exist several widely used heuristics which often yield reasonably suboptimal and approximately or exactly feasible solutions to the GPO problem (e.g. [12], [13]), but which we will not discuss here in depth.

If we expand our definition of algorithm to include those with combinatorial complexity but finite termination time, then if the feasible set of the GPO problem is compact and the ideal generated by the equality constraints is radical and zero dimensional (typical for integer programming [14]), then one may use the moment approach to obtain an algorithm with finite termination time. Unfortunately, however, it has been shown that the class of problems for which these methods terminate is a strict subset of the general class of GPO problems [15] and furthermore, there are no tractable conditions verifying if the algorithm will terminate or bounds on computational complexity in the case of finite termination.

In this paper, we propose a new polynomial-time algorithm for using SOS estimates of the Greatest Lower Bound (GLB) to extract arbitrarily accurate approximate solutions. Specifically, define the feasible set

$$S := \{x \in \mathbb{R}^n : g_i(x) \geq 0,\, h_j(x) = 0\}. \qquad (2)$$

The Greatest Lower Bound (GLB) problem associated to the GPO Problem (1) is defined as

$$
\begin{aligned}
\lambda^* := \max_{\lambda\in\mathbb{R}}. \quad & \lambda && (3)\\
\text{subject to} \quad & f(x) - \lambda > 0 \quad ,\forall x \in S.
\end{aligned}
$$

The GLB and GPO problems are closely related in that $\lambda^* = f^* = f(x^*)$, but are not equivalent in that the GLB problem does not find x^*. Our approach is based on the observation that while the GLB problem does not return x^*, it can be used as the selection criteria in a bisection algorithm - See Section V. This means that an algorithm which solves the GLB problem with complexity $O(k)$ can be combined with bisection to find a point $x \in \mathbb{R}^n$ such that $|x-x^*| \leq \varepsilon$ for any ε and the resulting complexity is of order $O(\log(1/\varepsilon)k)$.

Unfortunately there is no algorithm which solves the GLB exactly in polynomial time. As a consequence, we turn to asymptotically exact approaches to solving the GLB problem such as Sum of Squares (SOS) programming [16], and its dual Moment-relaxation problem [17]. Both these approaches are well-studied and have associated MATLAB toolboxes, including SOSTOOLS [18] and Gloptipoly [19]. Both approaches yield a hierarchy of primal/dual semidefinite programs (SDPs) with increasing sequences of optimal values $\{p_k^*\}_{k\in\mathbb{N}}$ (SOS) and $\{d_k^*\}_{k\in\mathbb{N}}$ (Moment) such that $p_k^* \leq d_k^* \leq f^*$. Furthermore, under mild conditions, the algorithms are asymptotically accurate in the sense that $\lim_{k\to\infty} p_k^* = \lim_{k\to\infty} d_k^* = f^*$ [20]. Moreover, if the feasible set, S, as defined in (2), is nonempty and compact, then there exist constants c_1 and c_2, depending on polynomials f, h_i, and g_i such that $|p_k^* - f^*| \cong \frac{c_2}{\sqrt[c_1]{\log(k)}}$ [21].

Because the SOS/Moment approaches to solving the GLB problem are not exact, simple bisection algorithms based on selecting the region with lowest GLB bound may fail to converge. However, we note that the convergence results for SOS indicate that for any desired accuracy. There is an SOS solution to the GPO problem with that desired accuracy. Therefore, by using this accuracy as a tolerance in our algorithm, we can choose a trimming rule which is guaranteed to converge to a point with that desired accuracy. Specifically, in Sec. VI, we propose a modification of the traditional branch and bound algorithm. The sequence of algorithms can be briefly summarized as follows. Algorithm E_k initializes with hyper-rectangle $C_1 := \{x \,|\, (x_i - \underline{c}_{i,1})(\bar{c}_{i,1} - x_i) \geq 0, i = 1, \cdots, n\}$ such that $S \subset C_1$. Then, at iteration m, the algorithm forms two new hyper-rectangles by bisecting C_m by its longest edge $(\bar{c}_i - \underline{c}_i)$ and intersecting each of the new hyper-rectangles with S to create two new GPO problems. Then, the algorithm computes a GLB estimate of the optimal value of the new GPO problem by solving the corresponding k'th-order SOS/Moment relaxations. If λ_m^* denotes the best lower bound to f^*, obtained up to iteration m, then C_{m+1} is determined to be the existing hyper-rectangle with a smallest volume subject to the constraint that the corresponding GLB is less than $\frac{m\eta}{l} + \lambda_m^*$, where $\eta > 0$ and $l \in \mathbb{N}$ are the design parameters. Finally, after a certain number of iterations, the algorithm terminates by returning the centroid of the last hyper-rectangle.

In Sec. VII, we prove that the sequence of algorithms, E_k, will return in polynomial time a point $x_k \in \mathbb{R}^n$ that is suboptimal to the GPO problem in the following sense. If x_k is the sequence of proposed solutions produced by the sequence of algorithms E_k, we show that if the feasible set, S, of Problem (1) is bounded, then there exists a sequence of feasible points $y_k \in S$ such that $\lim_{k\to\infty} ||x_k - y_k|| = 0$ and $\lim_{k\to\infty} f(y_k) = \lim_{k\to\infty} f(x_k) = f^*$, where f^* is the optimal value of the GPO problem.

In Sec. VIII, we illustrate the effectiveness of the proposed algorithm by applying it to an example problem wherein the existing Moment based approach fails to extract a solution. Finally we conclude in Sec. IX.

II. Notation

Let $\mathbb{N}^n$ be the set of n-tuples of natural numbers. We use $\mathbb{S}^n$ and $\mathbb{S}^{n+}$ to denote the symmetric matrices and cone of positive semidefinite matrices of size $n \times n$, respectively. For any $a, b \in \mathbb{R}^n$, we denote by $C(a,b)$ the hyper-rectangle $\{x \in \mathbb{R}^n | a \leq x \leq b\}$, where $y \geq 0$ is defined by the positive orthant. For $x \in \mathbb{R}$ we denote by $y = \lceil x \rceil$ the smallest integer $y \in \mathbb{N}$ such that $y \geq x$. We use ℓ_∞ to denote the set of bounded infinite sequences. For any $k, n \in \mathbb{N}$, let $\mathbb{N}^n_{(k)} := \{b \in \mathbb{N}^n : |b|_{\ell_1} \leq k\}$, where $|b|_{\ell_1} := \sum_{i=1}^n |b_i|$. Finally, we denote the ring of multivariate polynomials with real coefficients as $\mathbb{R}[x]$.

III. Problem Statement

In this paper, we consider simplified GPO problems of the form:

$$f^* := \min_{x\in\mathbb{R}^n} \quad f(x) \tag{4}$$
$$\text{subject to} \quad g_i(x) \geq 0 \quad \text{for } i = 0, \cdots, l$$

where $f, g_i \in \mathbb{R}[x]$. The class of problems in (4) is equivalent to that in (1), where we have simply replaced every $h_i(x) = 0$ constraint with some $g_1(x) = h(x) \geq 0$ and $g_2(x) = h(x) \leq 0$. For every problem of Form (4), we define the associated feasible set $S := \{x \in \mathbb{R}^n : g_i(x) \geq 0\}$.

In this paper, we assume $S \neq \emptyset$. Note that given g_i, one may use SOS optimization combined with Positivstellensatz results [22] to determine feasibility of S.

Proposed Algorithm In this paper, we propose a GLB and Branch and Bound-based algorithm which, for any given $\varepsilon > 0$, will return some $x \in \mathbb{R}^n$ for which there exists a point $y \in S$ such that:

$$f(y) - f^* \leq \varepsilon, \qquad \text{and} \qquad \|y - x\| < \varepsilon. \tag{5}$$

Furthermore, x itself is ε-suboptimal in the sense that $|f(x) - f^*| \leq \varepsilon$ and $g_i(x) \geq -\varepsilon$.

Before defining this algorithm, however, in the following section, we describe some background on the dual SOS and Moment algorithms for generating approximate solutions of the GLB Problem.

IV. Background on Semidefinite Representations of SOS/Moment Relaxations

In this section, we describe two well-known asymptotic algorithms which are known to generate sequences of increasingly accurate suboptimal solutions to the GLB problem - namely the SOS and Moment approaches. Both these methods use Positivstellensatz results which parameterize the set of polynomials which are positive over a given semialgebraic set.

A. Sum-of-Squares Polynomials

In this subsection, we briefly define and denote sets of sums of squares of polynomials.

We denote monomials in variables $x \in \mathbb{R}^n$ as $x^\alpha := \prod_{i=1}^n x_i^{\alpha_i}$ where $\alpha \in \mathbb{N}^n$. Monomials can be ordered using various orderings on $\mathbb{N}^n$. In this paper, we use the graded lexicographical ordering. This ordering is defined inductively as follows. For $a, b \in \mathbb{N}^n$, $a \leq b$ if $\sum_{i=1}^n a_i < \sum_{i=1}^n b_i$, or $a_1 = b_1$ and $[a_2, \cdots, a_n] \leq [b_2, \cdots, b_n]$. Denote by $Z(x)$ the infinite ordered vector of all monomials, where $x^\alpha < x^\beta$ if $\alpha < \beta$. Because we have used the graded lexicographical ordering, if we restrict ourselves to the first $\binom{d+n}{d}$ elements of Z, then this is the vector of all monomials of degree d or less. We denote this truncated vector as $Z_d(x)$ and the length of Z_d as $\Lambda(d) := \binom{d+n}{d}$. Using this definition, it is clear that any polynomial can be represented as $p(x) = c^T Z_d(x)$ for some $c \in \mathbb{R}^{\Lambda(d)}$, where d is the degree of p.

The vector of monomials can also be combined with positive semidefinite matrices to completely parameterize the cone of sums-of-squares polynomials. Formally, we can denote the subset of polynomials which are the sum of squares of polynomials as

$$\Sigma_S := \{s \in \mathbb{R}[x] : s(x) = \sum_{i=1}^{l} p_i^2(x), p_i \in \mathbb{R}[x], l \in \mathbb{N}\}. \tag{6}$$

Clearly any element of Σ_S is a nonnegative polynomial. Furthermore, if $p \in \Sigma_S$ and is of degree $2d$, then there exists a positive semidefinite matrix $\Omega \in \mathbb{S}^{\Lambda(d)+}$ such that

$$p(x) = Z_d(x)^T \Omega Z_d(x).$$

Conversely, any polynomial of this form, with $\Omega \geq 0$, is SOS. This parametrization of SOS polynomials using positive semidefinite matrices will allow us to convert the GLB problem to an LMI. However, before defining this LMI approach, we must examine the question of positivity on semialgebraic subsets of $\mathbb{R}^n$.

B. Putinar's Positivstellensatz, Quadratic Modules and the Archimedean Property

Sum-of-Squares polynomials are globally non-negative. In this section, we briefly review Putinar's positivstellensatz which gives necessary conditions for a polynomial to be positive on the semiaglebraic set $S := \{x \in \mathbb{R}^n : g_i(x) \geq 0,\ i = 1,\dots,l\}$, where $S \neq \emptyset$ and is compact.

Putinar's Positivstellensatz uses the g_i which define S to deduce a cone of polynomials which are non-negative on S. This cone is the quadratic module which we define as follows.

Definition 1: Given a finite collection of polynomials $g_i \in \mathbb{R}[x]$, we define the quadratic module as

$$M := \{p | p = \sigma_0 + \sum_{i=1}^{l} \sigma_i g_i \quad \sigma_i \in \Sigma_S\},$$

and the degree-k bounded quadratic module as

$$M^{(k)} := \{p | p = \sigma_0 + \sum_{i=1}^{l} \sigma_i g_i \quad \sigma_i \in \Sigma_S \quad \deg(\sigma_i g_i) \leq k\}.$$

Clearly, any polynomial in M is non-negative on S. Furthermore, since Σ_S parameterizes M and positive semidefinite matrices parameterize Σ_S, the constraint $p \in M^{(k)}$ can be represented as an LMI. Furthermore, if the module satisfies the Archimedean property, then Putinar's Positivstellensatz states that any polynomial which is positive on S is an element of M. That is, M parameterizes the cone of polynomials positive on S.

A quadratic module M is said to be Archimedean if there exists some $p \in M$ and $R \neq 0$ such that $p(x) = R^2 - \sum_{i=1}^n x_i^2$. We say that $\{g_i\}$ is an Archimedean representation of S if the associated quadratic module is Archimedean. Note that the Archimedean property is a property of the functions g_i which then define the quadratic module and not a property of S. Specifically, if S is compact, and $\{g_i\}_{i=1^l}$ is not Archimedean, then we may construct an Archimedean representation $\{g_i\}_{i=1^{l+1}}$ where $g_{l+1}(x) = R^2 - \sum_{i=1}^n x_i^2$ for any sufficiently large $R > 0$.

C. SOS approach to solving the GLB problem

In this subsection, we briefly describe the use of SOS programming to define a hierarchy of GLB problems.

Consider the GPO Problem (4) where $\{g_i\}$ is an Archimedean representation of the feasible set, S, with associated quadratic module M. We now define the degree-unbounded version of the SOS GLB problem.

$$f^* = \lambda^* := \max_{\lambda \in \mathbb{R}}. \quad \lambda \qquad (7)$$
$$\text{subject to} \quad f(x) - \lambda \in M.$$

Since M is Archimedian, it follows that $\lambda^* = f^*$ (where f^* is as defined in (4)). Although Problem (7) is convex, for practical implementation we must restrict the degree of the SOS polynomials which parameterize M - meaning, we must restrict ourselves to optimization on $M^{(k)}$. This defines a new sequence of GLB problems as

$$p_k^* := \max_{\lambda \in \mathbb{R}}. \quad \lambda \qquad (8)$$
$$\text{subject to} \quad f(x) - \lambda \in M^{(k)}.$$

Clearly, $p_i^* \leq p_j^* \leq \lambda^*$ for any $i < j$. Additionally, it was shown in [16] that $\lim_{k\to\infty} p_k^* = p^*$. Furthermore, it was shown in [20], [21] that bounds on the convergence rate of $p_k^* \to \lambda^*$ exist as a function of g_i, f and k. Finally, the computational complexity of p_k is equivalent to that of a semidefinite program with order $(l+1)\Lambda(\lceil \frac{k}{2} \rceil)^2$ scalar variables, where l is the number of g_i in Problem (4).

V. Solving the GPO Problem using the ideal Branch and Bound

In this section we show that, given an algorithm that returns the exact solution to the GLB problem in $O(k)$ steps, we can design an algorithm that returns $x \in \mathbb{R}^n$ such that $|x - x^*| \leq \varepsilon$, in $O(\log(1/\varepsilon)k)$ steps for any desired accuracy ε.

The Ideal Branch and Bound Algorithm
At every iteration, we have an active hyper-rectangle $A(k) = [a_i, b_i]$;

1) Initialize the algorithm;
2) Bisect $A(k) = [a_i, b_i] = [a', b'] \cup [a'', b''] = A_1 \cup A_2$;
3) Compute the Greatest Lower Bound of
$$\lambda_i^* := \max_{\lambda \in \mathbb{R}}. \quad \lambda \qquad (9)$$
$$\text{subject to} \quad f(x) - \lambda > 0 \quad ,\forall x \in S \cap A_i;$$
4) If $\lambda_1^* > \lambda_2^*$, set $A(k+1) = A_1$, otherwise $A(k+1) = A_2$;
5) Goto 2 ;

At termination, we may choose any $x \in A$, which will be accurate within $|x - x^*| \leq r 2^{-k/n}$.

Let us examine these steps in more detail.

Initialize the algorithm Since the set S is compact, there exists some $r > 0$ such that $S \subset B_r(r)$. We may then initialize $A = [-r\mathbf{1}, r\mathbf{1}]$, where $\mathbf{1}$ is the vector of all ones.

Bisect Bisection of the hypercube occurs along the longest edge. Thus, after n iterations, we are guaranteed a two-fold increase in accuracy. As a result, the largest edge of the hypercube diminishes as $2^{-k/n}$.

Compute the Greatest Lower bound We assume that our solution to the GLB problem is exact. In this case, we are guaranteed that an optimizing x will always lie in A_i.

As an example, consider the simple two variate optimization problem,

$$\min_{x,y\in\mathbb{R}} \quad y \tag{10}$$
$$\text{subject to} \quad x+5\geq 0, \qquad xy-10\geq 0,$$
$$15-x-y\geq 0, \qquad x^2+3y^2-180=0.$$

Conceptually, the first eight iterations of the ideal branch and bound algorithm for this two variate optimization problem are displayed in Figure 1. In this case, the algorithm returns a sequence of nested rectangles $A(k)$. At iteration k, if the optimal feasible point x^* lies in the rectangle $A(k)$, then if $A_1(k),A_2(k)$ is the bisection of $A(k)$, and $i(k) = \arg\min\{\lambda^*(A_1(k)),\lambda^*(A_2(k))\}$, where $\lambda^*(A)$ is the greatest lower bound over domain A, then x^* is guaranteed to lie in $A_{i(k)}$. Therefore, by induction, at every iteration k, x^* is guarnteed to lie in domain $A(k)=A_{i(k)}$. Figure 1 illustrates the rectangle discarded at each iteration, the optimal point x_1^* (which lies in A_k at each iteration) and the associated optimal objective values $\lambda^*(A_i(k))$.

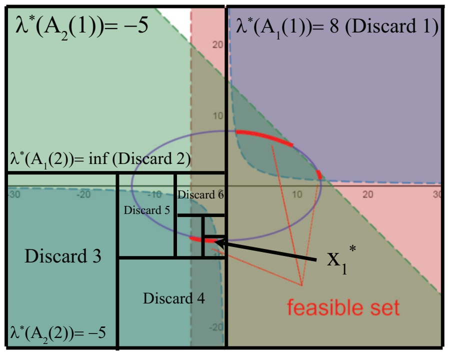


Fig. 1. The ideal Branch and Bound Algorithm applied to the two-variate optimization problem (13). Note that every iteration discards half of the active hyper-rectangle.

A. Complexity of the Ideal Branch and Bound Algorithm

In this subsection, we show that any exact solution to the GLB can be used to solve the GPO problem with arbitrary accuracy in a logarithmic number of steps using the Ideal Branch and Bound algorithm.

Suppose G is a GLB Problem of the Form (3) with solution λ^*. Define the algorithm $H: G\mapsto\lambda^*$ as $\lambda^*=H(G)$. Further suppose H has time-complexity $O(k)$, where k is a measure for the size of G. In the Ideal Branch and Bound algorithm, if we use H to perform Step (3), it is straightforward to show that for any $\varepsilon>0$, after $m=2n(c_1+\log\frac{1}{\varepsilon})$ iterations, if $A_m=[a,b]$, then $\max_i|b_i-a_i|\leq\varepsilon$ and GPO Problem (4) has a minimizer $x^*\in C(a,b)$, where c_1 depends on the size of S and n is the number of variables. Now since all the GLB problems defined in Step (3) of the algorithm are of equal size, k, the complexity of the Ideal Branch and Bound algorithm for a given ε, is $O(2nk(c_1+\log\frac{1}{\varepsilon}))$.

In this section, we considered the ideal case when the GLB can be solved exactly and we can use Branch and Bound to find an x^* such that $|x-x^*|\leq\varepsilon$ in $O(\log(\varepsilon^{-1}))$ iterations. In the following section we consider the case when SOS/Moment problems are used to solve the GLB problem and the greatest lower bound is not exact, but is guaranteed to be arbitrarily accurate depending on the choice of k.

VI. Modified Branch and Bound Algorithm

Here, we present a slightly modified branch and bound algorithm that, combined with SOS/Moment relaxations, can approximate the solution to the GPO problem to any desired accuracy.

The Modified Branch and Bound Algorithm At every iteration, we have an active hyper-rectangle $A=[a,b]$ and a set of feasible rectangles $Z=\{[a_i,b_i]\}_i$ each with associated GLB λ_i.

1) Initialize the algorithm
2) Bisect $A=[a,b]=[a',b']\cup[a'',b'']=A_1\cup A_2$
3) Compute the Greatest Lower Bound of

$$\lambda_i^* := \max_{\lambda\in\mathbb{R}} \quad \lambda \tag{11}$$
$$\text{subject to} \quad f(x)-\lambda>0 \quad ,\forall x\in S\cap A_i.$$

4) If $\lambda_i^*\leq\lambda^*+\varepsilon$, add A_i to Z.
5) Set $A=Z_i$ where Z_i is the smallest element of Z.
6) Goto 2

At termination, we choose any $x\in A$, which will be accurate within $|x-x^*|\leq r2^{-k/n}$.

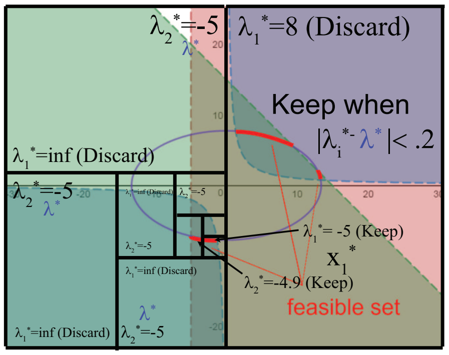


Fig. 2. The Modified Branch and Bound Algorithm applied to the two-variate optimization problem (13), where λ_i^* and λ^* are the values from line 4 of the modified branch and bound algorithm. Note that on the seventh iteration both of the hyper-rectangles are kept.

The first eight iterations of the modified branch and bound algorithm for the two variate optimization problem (10) are displayed in Figure 2. At the k-th iteration we have a hyper-rectangle A, that is bisected into $A_1(k)$ and $A_2(k)$, both with computed greatest lower bounds $\lambda^*(A_1(k))$ and $\lambda^*(A_2(k))$ respectively. In the modified branch and bound we add $A_i(k)$ to the collection of sets $Z(k)$ if $\lambda^*(A_i(k))\leq\min\{\lambda^*(E)|E\in Z(k)\}+\varepsilon$. Note that this means both $A_1(k)$ and $A_2(k)$ can be added to $Z(k)$ to obtain $Z(k+1)$ and in Figure 2 this occurs on the seventh iteration. The hyper-rectangle $A(k+1)$ to be bisected in the next iteration is then the smallest (by volume) hyper-rectangle in $Z(k+1)$.

A. Problem Definition and SOS/Moment Subroutine

Consider GPO Problem (4) and suppose the corresponding feasible set $S:=\{x\in\mathbb{R}^n : g_i(x)\geq 0\}$, is nonempty and compact with $S\subset C(a,b)$, for some $a,b\in\mathbb{R}^n$ with associated Archimedean quadratic module M.

Before defining the main sequential algorithm E_k, we will define the k^{th}-order SOS/Moment GLB subroutine, denoted

B_k, which calculates the GLB in Step (3) of the Modified Branch and Bound Algorithm.

SOS/Moment Subroutine $\lambda_{\mathbf{k}}^* = \mathbf{B_k}[\mathbf{a},\mathbf{b}]$

Given $A = [a,b]$, define the polynomials $w_i(x) := (b_i - x_i)(x_i - a_i)$. These polynomials are then used to define the modified feasible set $S \cap A$ as

$$S_{ab} := \{x \in \mathbb{R}^n : g_i(x) \geq 0,\ \forall i : 1 \leq i \leq s, \quad w_j(x) \geq 0,\ \forall j :\ 1 \leq j \leq n\}, \tag{12}$$

and the corresponding modified degree-k bounded quadratic module as

$$M_{ab}^{(k)} := \Big\{ p : p = \sum_{i=0}^{s} \sigma_i g_i + \sum_{i=s+1}^{s+n} \sigma_i w_i, \quad \sigma_i \in \Sigma_S, \quad \deg(\sigma_i g_i) \leq k,\ \deg(\sigma_i w_i) \leq k \Big\}, \tag{13}$$

where $g_0(x) = 1$. This allows us to formulate and solve the modified k-th order SOS GLB problem

$$p_k^* := \max_{\lambda \in \mathbb{R}}.\ \lambda \quad \text{subject to} \quad f(\mathbf{x}) - \lambda \in M_{ab}^{(k)} \tag{14}$$

and the corresponding dual GLB moment problem. The subroutine returns the value $\lambda_k^* = p_k^*$ such that, recall, $|p_k^* - f^*| \cong \frac{c_2}{\sqrt[c]{\log(k)}}$.

B. Formal Definition of the Modified Branch and Bound Algorithm, E_k

We now define a sequence of Algorithms E_k such that for any $k \in \mathbb{N}$, E_k takes GPO Problem (4) and returns an estimated feasible point x^*.

The Sequence of Algorithms E_k:

In the following, we use the notation $a \leftarrow b$ to indicate that the algorithm takes value b and assigns it to a. In addition parameter $0 < \eta < 1$ represents error tolerance for trimming branches and in Theorem 1 is set by the desired accuracy as $\eta < \varepsilon$. The parameter l represents the number of branch and bound loops and in Theorem 1 is set by the desired accuracy as $l > n\log_2(\frac{L\sqrt{n}}{\eta})$ where n is the number of variables and L is a bound on the radius of the feasible set.

The inputs to the following algorithm E_k are the functions $\{g_i\}$ and f, the initial hyper-rectangle such that $S \subset [a,b]$, and the design parameters η, l. The output is the estimated feasible point, x.

Algorithm E_k :

```
input: $\eta > 0$, $l \in \mathbb{N}$, $a,b \in \mathbb{R}^n$, $f, g_1, \dots, g_s \in \mathbb{R}[x]$.
output: $x \in \mathbb{R}^n$ (as an approximate solution to GPO Problem (4)).
Initialize:
    $a(0) \leftarrow a$; $b(0) \leftarrow b$; $m \leftarrow 0$; $\lambda(0) \leftarrow B_k(a(0),b(0))$;
While $(m < l)$: {
    $j^* \leftarrow \arg\min_{j \in \{0,\dots,m\}} \lambda(j)$;
    $i^* \leftarrow \arg\min_{j \in \{0,\dots,m\}} \prod_{i=1}^{n} (b(j)_i - a(j)_i)$
        subject to $\lambda(j) \leq \lambda(j^*) + \frac{m\eta}{1+l}$;        (15)
    $a^* \leftarrow a(i^*)$;    $b^* \leftarrow b(i^*)$;
    $r^* \leftarrow \arg\max_{j \in \{1,\dots,n\}} (b_j^* - a_j^*)$;  $\tilde{a} \leftarrow a^*$;  $\hat{b} \leftarrow b^*$;
    For $r$ from 1 to $n$: {
        $\tilde{b}_r \leftarrow \begin{cases} \frac{b_r^*+a_r^*}{2} & \text{if } r = r^* \\ b_r^* & \text{otherwise} \end{cases}$;  $\hat{a}_r \leftarrow \begin{cases} \frac{b_r^*+a_r^*}{2} & \text{if } r = r^* \\ a_r^* & \text{otherwise,} \end{cases}$;}
    $\tilde{\lambda} \leftarrow B_k(\tilde{a},\tilde{b})$;   $\hat{\lambda} \leftarrow B_k(\hat{a},\hat{b})$;   $m \leftarrow m+1$;
    $a(i^*) \leftarrow \tilde{a}$;   $b(i^*) \leftarrow \tilde{b}$;   $\lambda(i^*) \leftarrow \tilde{\lambda}$;
    $a(m) \leftarrow \hat{a}$;   $b(m) \leftarrow \hat{b}$;   $\lambda(m) \leftarrow \hat{\lambda}$;}
Return $x := \frac{a(l)+b(l)}{2}$;
```

In the following section we will discuss the complexity and accuracy of the sequence of Algorithms E_k.

VII. Convergence and Complexity of E_k

In this section, we first show that for any $k \in \mathbb{N}$, the greatest lower bounds obtained by the subroutine B_k increase at each iteration of the Branch and Bound loop. Next we show that for any desired accuracy, there exists a sufficiently large k, such that Algorithm E_k returns a proposed solution with that accuracy.

In the following lemma, we use Lemma 2 from the appendix to show that for any a_1, a_2, b_1, $b_2 \in \mathbb{R}^n$ such that $a_1 \leq a_2 < b_2 \leq b_1 \in \mathbb{R}^n$, the feasible set of the SOS problem solved in Subroutine $B_k(a_1,b_1)$ is contained in that of Subroutine $B_k(a_2,b_2)$.

Lemma 1: For any $k \in \mathbb{N}$ and $a \leq b \in \mathbb{R}^n$, let $M_{ab}^{(k)}$ be the modified degree-k bounded quadratic module associated to polynomials $g_1, \dots, g_s$, as defined in (13). If $\gamma \leq \alpha < \beta \leq \delta \in \mathbb{R}^n$, then $M_{\gamma\delta}^{(k)} \subset M_{\alpha\beta}^{(k)}$, for all $k \geq 2$.

Proof:

For any $j = 1, \dots, n$, let $w_{j,1}(x) := (\beta_j - x_j)(x_j - \alpha_j)$, and $w_{j,2}(x) := (\delta_j - x_j)(x_j - \gamma_j)$. Since $\gamma_j \leq \alpha_j < \beta_j \leq \delta_j$, then it is followed from Lemma 2 in the appendix that there exist $p_j, q_j, r_j \in \mathbb{R}$ such that

$$w_{j,2}(x) = p_j^2\, w_{j,1}(x) \ +\ q_j^2\, (x_j + r_j)^2.$$

Now, if $h \in M_{\gamma\delta}^{(k)}$, we will show that $h \in M_{\alpha\beta}^{(k)}$. By definition, there exist $\sigma_i, \omega_{j,2} \in \Sigma_S$ such that $h = \sum_{i=0}^{s} \sigma_i g_i + \sum_{j=1}^{n} \omega_j w_{j,2}$, where $g_0(x) = 1$. Hence, we can plug in the expression for $w_{j,2}$ to get

$$h = \sum_{i=0}^{s} \sigma_i \cdot g_i + \sum_{j=1}^{n} \omega_j \cdot (p_j^2 \cdot w_{j,1} + q_j^2 \cdot (x_j + r_j)^2)$$
$$= \underbrace{(\sigma_0 + \sum_{j=1}^{n} q_j^2 \cdot \omega_j \cdot (x_j + r_j)^2)}_{\sigma_{0\text{ new}}} + \sum_{i=1}^{s} \sigma_i \cdot g_i + \sum_{j=1}^{n} \underbrace{p_j^2\, \omega_j}_{\omega_{j\text{ new}}} \cdot w_{j,1}.$$

Clearly $\sigma_{0\,\text{new}}, \omega_{j\,\text{new}} \in \Sigma_S$. Furthermore, since $k \geq 2$, $\deg(\sigma_{0\,\text{new}}) \leq k$, and $\deg(\omega_{j\,\text{new}} \cdot w_{j,1}) \leq k$ which implies that $h \in M_{\alpha\beta}^{(k)}$. ■

Now suppose $\{g_i\}$ all have degree d or less. Then for any $k \geq d+2$ and for any hyper-rectangles $C(c,d) \subset C(a,b)$, if $\lambda_{(a,b)}$ and $\lambda_{(c,d)}$ are the solutions obtained by Subroutines $B_k(a,b)$ and $B_k(c,d)$ applied to GPO Problem (4), then Lemma 1 shows that $\lambda_{(a,b)} \leq \lambda_{(c,d)}$. Now, for a fixed $k \in \mathbb{N}$, let η and l be the design parameters of Algorithm E_k applied to GPO Problem (1). For $m = 0, \ldots, l$, let $(\lambda^*)_m := \lambda(j^*)$, where j^* is as we defined in iteration m of the loop in Algorithm E_k. Using Lemma 1, it is straightforward to show that $(\lambda^*)_m \leq (\lambda^*)_{m+1}$ for $m \leq l-1$.

In the next theorem, we will show that for any given $\varepsilon > 0$, there exist $k \in \mathbb{N}$ such that Algorithm E_k applied to GPO Problem (4) will provide a point $x \in \mathbb{R}^n$ satisfying (5).

Theorem 1: Suppose GPO Problem (4) has a nonempty and compact feasible set S. Choose $a, b \in \mathbb{R}^n$ such that $S \subset C(a,b)$. For any desired accuracy, $0 < \varepsilon < 1$, let $l > n \log_2(\frac{L\sqrt{n}}{\eta})$ and $\eta < \varepsilon$ where $L = \max_i b_i - a_i$. Then there exists a $k \in \mathbb{N}$ such that if $x = E_k(\eta, l, a, b, f, g_i)$, then there exists a feasible point $y \in S$ such that $f(y) - f^* \leq \varepsilon$ and $\|y - x\| < \varepsilon$, where f^* is the objective value of the GPO Problem (4).

Proof: Define $\mathscr{P}$ to be the set of all possible hyper-rectangles generated by the branching loop of Algorithm E_k (for any k) with number of branches bounded by l. The vertices of all elements of $\mathscr{P}$ clearly lie on a grid with spacings $\frac{|a_i, b_i|}{2^l}$. Therefore, the cardinality $|\mathscr{P}|$ is finite and bounded as a function of l, a and b. It has been shown that for any $C_\alpha := C(e,f) \in \mathscr{P}$, there exists a $k_\alpha \in \mathbb{N}$ such that for any $k' \geq k_\alpha$, the solution of Subroutine $B_{k'}(e,f)$ is accurate with the error tolerance $\frac{\eta}{1+l}$. Now define $k := \max\{k_\alpha \mid C_\alpha \in \mathscr{P}\}$.

Will now show that Algorithm E_k returns a point x with the desired accuracy. First, we show that Algorithm E_k generates exactly l nested hyper-rectangles. The proof is by induction on m.

For $m = 0, \ldots, l-1$, let $(a)_m := a(i^*)$, $(b)_m := b(i^*)$, $(C)_m := C((a)_m, (b)_m)$, $(\lambda)_m := B_k((a)_m, (b)_m)$, $(\tilde{a})_m := \tilde{a}$, $(\tilde{b})_m := \tilde{b}$, $(\hat{a})_m := \hat{a}$, $(\hat{b})_m := \hat{b}$, $(\tilde{C})_m := C((\tilde{a})_m, (\tilde{b})_m)$, $(\hat{C})_m := C((\hat{a})_m, (\hat{b})_m)$ and $(\lambda^*)_m := \lambda(j^*)$ where i^*, j^*, $\tilde{a}$, $\tilde{b}$, $\hat{a}$ and $\hat{b}$ are defined as in iteration m of Algorithm E_k.

We use induction on m to show that for all $m \leq l$:

$$(C)_m \subset (C)_{m-1}.$$

The base case $m = 0$ is trivial. For the inductive step, first note that $(\lambda^*)_m \leq f^*$ for all $m \leq l$ and $(\lambda^*)_1 \leq \cdots \leq (\lambda^*)_l$. The latter is obtained from Lemma 1 and the former is because at each iteration, $S \subset \bigcup_{i=0}^m C(a(i), b(i))$. Constraint (15) at iteration m, implies that

$$B_k((a)_m, (b)_m) \leq (\lambda^*)_m + \frac{m\eta}{l+1}. \tag{16}$$

Now we will show that again, Constraint (15) at iteration $m+1$ is satisfied at least by one of $(\tilde{C})_m$ and $(\hat{C})_m$. Suppose this is not true. Then we can write

$$\begin{aligned}(\lambda^*)_{m+1} &< B_k((\tilde{a})_m, (\tilde{b})_m) - \frac{(m+1)\eta}{1+l}, \\ (\lambda^*)_{m+1} &< B_k((\hat{a})_m, (\hat{b})_m) - \frac{(m+1)\eta}{1+l}.\end{aligned} \tag{17}$$

Now, since $(\lambda^*)_{m+1} \geq (\lambda^*)_m$, Eq. (17) implies

$$\begin{aligned}(\lambda^*)_m &< B_k((\tilde{a})_m, (\tilde{b})_m) - \frac{(m+1)\eta}{1+l}, \\ (\lambda^*)_m &< B_k((\hat{a})_m, (\hat{b})_m) - \frac{(m+1)\eta}{1+l}.\end{aligned} \tag{18}$$

Using Eq. (18) and Eq. (16) one can write

$$\begin{aligned}B_k((a)_m, (b)_m) &< B_k((\tilde{b})_m, (\tilde{b})_m) - \eta/l, \\ B_k((a)_m, (b)_m) &< B_k((\hat{a})_m, (\hat{b})_m) - \eta/l.\end{aligned} \tag{19}$$

This contradicts the fact that all $B_k((\tilde{a})_m, (\tilde{b})_m)$, $B_k((\hat{a})_m, (\hat{b})_m)$ and $B_k((a)_m, (b)_m)$ have accuracy higher than $\eta/(1+l)$. Therefore, it is clear that both $(\tilde{C})_m$ and $(\hat{C})_m$ can be possible choices to be bisected at iteration $m+1$. This fact, together with the induction hypothesis which certifies that $(C)_m$ possesses the smallest volume between all the hyper-rectangles obtained up to that iteration, guaranteeing that either $(\tilde{C})_m$ or $(\hat{C})_m$, will be branched at the next iteration. Therefore, the algorithm will generate l nested hyper-rectangles. Now, $(\lambda^*)_0 \in [f^* - \eta/l, f^*]$ implies that

$$(\lambda^*)_m \in [f^* - \eta/l, f^*], \quad \text{for all } m = 1, \ldots, l. \tag{20}$$

Eq. (16) and Eq.(20) together with the fact that $(\lambda)_m \geq (\lambda^*)_m$ imply

$$(\lambda)_m \in \left[f^* - \eta/l, f^* + \frac{m\eta}{l+1}\right], \; \forall\, m = 1, \ldots, l. \tag{21}$$

Finally, as a special case $m = l$, one can write:

$$f^* - \frac{\eta}{1+l} \leq B_k((a)_l, (b)_l) \leq f^* + \frac{l\eta}{l+1}.$$

Now, note that the $\eta/(l+1)$-accuracy of $B_k((a)_l, (b)_l)$ implies that $(C)_l$ is feasible. It also can be implied that $(C)_l \cap S$ contains y such that $f(y) \geq B_k((a)_l, (b)_l \geq f(y) - \eta/(l+1)$. Therefore, $|f(y) - f^*| \leq \eta \leq \varepsilon$.

Finally, if x is the point return by Algorithm E_k, then based on the definition of l, it is implied that after the last iteration $m = l-1$, the largest diagonal of the branched hyper-rectangle is less than $\eta \leq \varepsilon$, hence $\|y - x\|_2 \leq \varepsilon$, as desired. ■

Theorem 1 ensures that for any accuracy $\varepsilon > 0$ there exists a $k \in \mathbb{N}$ such that Algorithm E_k returns ε-approximate solutions to the GPO problem with a logarithmic bound on the number of branching loops. The following corollary shows that these ε-approximate solutions can themselves approximately satisfy the constraints of the original GPO as follows.

Corollary 1: Let GPO Problem (4) have nonempty and compact feasible set that is contained in $C(a,b)$ for some $a, b \in \mathbb{R}^n$. For any given $\delta > 0$, there exists $\varepsilon > 0$ such that if ε and $x = E(\eta, l, a, b, f, g_i)$ satisfy the conditions in Theorem 1, then

$$|f(x) - f^*| \leq \delta \text{ and } \quad g_i(x) \geq -\delta, \, \forall i = 1, \ldots, s. \tag{22}$$

Proof: Let L be such that any polynomial $h \in \{f, g_1, \dots, g_s\}$ satisfies $|h(c) - h(d)| \le L|c-d|_2$, $\forall c, d \in C(a,b)$. (Existence of L follows from the Lipschitz continuity of polynomials on compact sets.) Choose ε such that $\varepsilon \le \delta / L$. Let ε and x satisfy the conditions in Thoerem 1. It is straightforward to show that x satisfies Eq. (22). ■

Unfortunately, of course, Theorem 1 does not provide a bound on the size of k (although the proof implies an exponential bound).

VIII. Numerical Results

Consider the following GPO problem.

$$\begin{aligned} \underset{x\in\mathbb{R}^6}{min.} \quad & f(x) = 7x_1x_5^3 + 6x_1x_5^2x_6 + 9x_2x_4^3 + 4x_2x_4x_5 + \\ & \qquad 3x_2x_5x_6 + x_3x_4x_5 \\ \text{subject to} \quad & g_1(x) = 100 - (x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2) \ge 0 \\ & g_2(x) = x_1^3 + x_2^2x_4 + x_3x_5^2 \ge 0 \\ & g_3(x) = x_2^2x_1 + x_5^3 + x_4x_1x_2 \ge 0 \\ & h_1(x) = x_1 + x_2^2 - x_3^2 + x_4x_5 = 0 \\ & h_2(x) = x_5x_1 - x_4^2 = 0. \end{aligned}$$

In this example we have 6 variables, an objective function of degree 4 and several equality and inequality constraints of degree 4 or less. Polynomial optimization problems similar to this example are important in economic dispatch models [5] or optimal power flow [6] and since the ideal generated by equality constraints in this case are not zero dimensional, the Moment approach to extracting solutions fails. We applied Algorithm E_5 to this problem with parameters $\eta = 0.005$ and $l = 200$, using Sedumi to solve the SDPs associated with the SOS and Moment problems. As seen in Figure 3, the branch and bound algorithm converges relatively quickly to a certain level of error and then saturates. Iterations past this point do not significantly improve accuracy of the feasible point. As predicted, this saturation and residual error (blue shaded region) is due the use of a fixed degree bound $k = 5$. As k is decreased, the residual error increases and as k is increased the residual error decreases. For this problem the final iteration returns the point $\hat{x} = [5.1416,\ 3.9307,\ 0.7568,\ -4.6777,\ 4.2676,\ -4.1504]$ for which all inequalities are feasible and the equality constraints h_1 and h_2 have errors of 0.0563 and 0.0610, respectively. The objective value is $f(\hat{x}) = -3693.3$.

To illustrate the practical significance of the proposed algorithm, we use the $\hat{x}$ obtained from E_5 as the seed to the standard MATLAB optimizer `fmincon`. The result is the point $\hat{x_2} = [5.1274,\ 3.9372,\ 0.8043,\ -4.6793,\ 4.2704,\ -4.1748]$, where again all inequalities are feasible and now the equality constraints h_1 and h_2 have errors of 1.015×10^{-9} and 0.019×10^{-9} respectively. Furthermore, the objective value is now $f(\hat{x_2}) = -3719$ which closely tracks the GLB value of -3718.94. To illustrate the importance of the initial guess on the results from `fmincon`, we ran several batteries of tests, successively decreasing the accuracy of the initial guess. Using $\hat{x_2}$ as the centroid, we proposed 50 randomly distributed initial guesses within a radius of $1, 2, 5$ and 10. These results are listed in Table I and indicate the increasing number of initial guesses which resulted in either local optima or no feasible solution - ultimately at 76% for maximum radius 10. These numerical tests indicate that the results of this paper have the potential to dramatically improve our ability to solve global polynomial optimization solvers.

TABLE I

A comparison of starting conditions of interior point method and the resulting number of suboptimal (OBV $< f(\hat{x_2})$) or failed solutions over 50 trials.

Max Perturbation	Suboptimal	No Solution	Trials
± 1	8	1	50
± 2	16	0	50
± 5	24	2	50
± 10	36	2	50

IX. Conclusion

We have proposed a sequence of Algorithms E_k, $k \in \mathbb{N}$ to extract solutions to the GPO problem based on a combination of Branch and Bound and SOS/Moment relaxations. The computational-complexity of Algorithm E_k is polynomial in k, polynomial in the number of constraints and linear in the number of branches l. Additionally, for any scalar $\varepsilon > 0$, there exist $k \in \mathbb{N}$ such that Algorithms E_k, in $O(\log(1/\varepsilon))$ number of iterations, returns a point that is within the ε-distance of a feasible and ε-suboptimal point. For a fixed degree of semidefinite relaxations, our numerical case study demonstrates convergence to a level of residual error which can then be decreased by increasing the degree, or through the use of local optimization algorithms that can find a local optimum near the ε-suboptimal point returned by the E_k algorithm with greater precision. In ongoing work, we seek to bound this residual error as a function of degree using available bounds on the error of SOS/Moment relaxations.

Appendix

The following lemma gives an algebraic property of the polynomials of the form $w(x) = (x - a_i)(b_i - x)$ which are used to define the augmented feasible set S_{ab}.

Lemma 2: Let $a \le c < d \le b \in \mathbb{R}$, $g := (x-a)(b-x)$ and $h := (x-c)(d-x)$. Then, there exist α, β and $\gamma \in \mathbb{R}$, such that

$$g(x) = \alpha h(x) + \beta (x + \gamma)^2, \quad \alpha, \beta \ge 0$$

Proof: Without loss of generality, one can assume that $a = 0$ (consider the change of variable $z := x - a$). Now let $p^2 := c$, $q^2 := d - c$, and $r^2 := b - d$. First, we consider the case where $p^2, r^2 \neq 0$. This leads to two sub-cases:

Case 1 : $r^2 \neq p^2$. Let

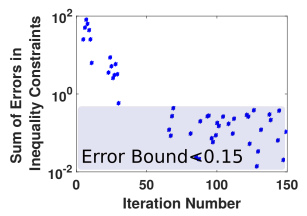
$$\gamma = \frac{p^4 + p^2q^2 - \sqrt{p^2r^2(p^2+q^2)(q^2+r^2)}}{r^2 - p^2}, \beta = \frac{p^4 + p^2q^2}{\gamma^2 - p^4 - p^2q^2},$$

and $\alpha = \beta + 1$. Verifying the equality $g(x) = \alpha h(x) + \beta (x + \gamma)^2$ is straightforward. To show that $\beta, \alpha \ge 0$, we use the following.

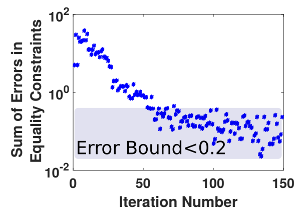
$$\beta \ge 0 \iff \gamma^2 > p^4 + p^2q^2$$

$$\iff \left(p^4 + p^2q^2 - \sqrt{p^2r^2(p^2+q^2)(q^2+r^2)}\right)^2 > (p^4 + p^2q^2)(r^2 - p^2)^2$$

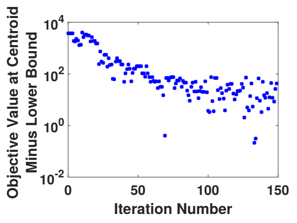
$$\iff \underbrace{(p^4 + p^2q^2)^2 + p^2r^2(p^2+q^2)(q^2+r^2) - (p^4 + p^2q^2)(r^2 - p^2)^2}_{L} >$$



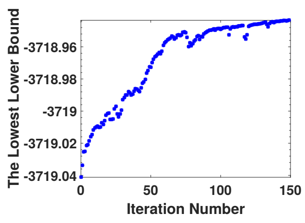
(a) Sum of errors in inequality constraints at center, $\sum_{g_i(\mathbf{x}_m)\leq 0} |g_i(\mathbf{x}_m)|$ vs. number of iterations



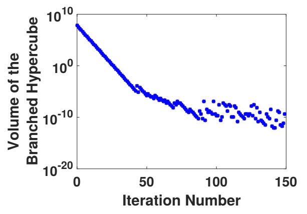
(b) Sum of errors in equality constraints at center, $\sum_j |h_j(\mathbf{x}_m)|$ vs. number of iterations



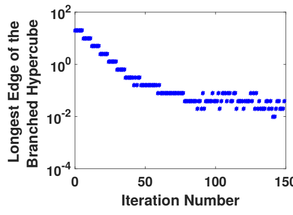
(c) Gap between lower bound and objective value at centroid, $|f(\mathbf{x}_m)-l|$ vs. # of iterations



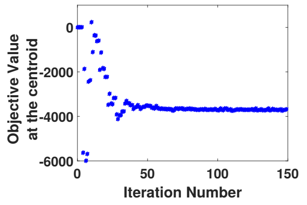
(d) Lower bound vs. number of iterations



(e) Lower bound vs. number of iteration



(f) Longest edge of the branched hypercube vs. number of iterations



(g) Objective value at center point vs. number of iterations

Fig. 3. Numerical results of the example

$$\underbrace{2\left(p^4+p^2q^2\right)\sqrt{p^2r^2(p^2+q^2)(q^2+r^2)}}_{U} \iff \begin{cases} L>0 \\ L^2>U^2 \end{cases}$$

After simplification we have:

$$L^2-U^2 = p^4q^4(p^2+q^2)^2(p^2-r^2)^2 > 0, \quad \text{and}$$
$$L = p^2\left(p^2+q^2\right)\left(p^2q^2+2p^2r^2+q^2r^2\right) > 0$$

which completes the proof for Case 1.

Case 2 : $r^2 = p^2$. In this case, let

$$\gamma = -\frac{2p^2+q^2}{2}, \beta = \frac{4p^2(p^2+q^2)}{q^4}, \alpha = \beta + 1$$

Equality and positivity for this case can then be easily verified. Now, suppose $r^2 = p^2 = 0$. In this case, simply set $\beta = 0$, $\alpha = 1$. If $p^2 = 0, r^2 \neq 0$, set $\beta = \frac{b}{d} - 1$, $\alpha = \frac{b}{d}$, $\gamma = 0$. The case $p^2 \neq 0$, $r^2 = 0$ is similar to $p^2 = 0$, $r^2 \neq 0$,through the change of variable $z = b - x$. ∎

ACKNOWLEDGMENTS

The authors gratefully acknowledge funding from NSF with the award number CMMI-1538374 for the present work.

REFERENCES

[1] L. Khachiyan, "Polynomial algorithms in linear programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 20, no. 1, pp. 53 – 72, 1980.
[2] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.
[3] Z.-Q. Luo and S. Zhang, "A semidefinite relaxation scheme for multivariate quartic polynomial optimization with quadratic constraints," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1716–1736, 2010.
[4] M. Laurent, *Emerging Applications of Algebraic Geometry*, ch. Sums of Squares, Moment Matrices and Optimization Over Polynomials, pp. 157–270. New York, NY: Springer New York, 2009.
[5] J. B. Park, Y. W. Jeong, J. R. Shin, and K. Y. Lee, "An improved particle swarm optimization for nonconvex economic dispatch problems," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 156–166, 2010.
[6] B. Ghaddar, J. Marecek, and M. Mevissen, "Optimal power flow as a polynomial optimization problem," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 539–546, 2016.
[7] J. Lavaei, "Optimal decentralized control problem as a rank-constrained optimization," in *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pp. 39–45, IEEE, 2013.
[8] M. A. B. Sassi and A. Girard, "Computation of polytopic invariants for polynomial dynamical systems using linear programming," *Automatica*, vol. 48, no. 12, pp. 3114 – 3121, 2012.
[9] B. Sturmfels, *Solving systems of polynomial equations*. No. 97, American Mathematical Soc., 2002.
[10] H. D. Sherali and C. H. Tuncbilek, "New reformulation linearization/convexification relaxations for univariate and multivariate polynomial programming problems," *Operations Research Letters*, vol. 21, no. 1, pp. 1 – 9, 1997.
[11] N. Z. Shor, "Quadratic optimization problems," *Soviet Journal of Computer and Systems Sciences*, vol. 25, no. 6, pp. 1–11, 1987.
[12] W. Cook, T. Koch, D. E. Steffy, and K. Wolter, "A hybrid branch-and-bound approach for exact rational mixed-integer programming," *Mathematical Programming Computation*, vol. 5, no. 3, pp. 305–344, 2013.
[13] D. E. Steffy and K. Wolter, "Valid linear programming bounds for exact mixed-integer programming," *INFORMS Journal on Computing*, vol. 25, no. 2, pp. 271–284, 2013.
[14] J. Lasserre, "Polynomials nonnegative on a grid and discrete optimization," *Transactions of the American Mathematical Society*, vol. 354, no. 2, pp. 631–649, 2002.
[15] J. Nie, "Optimality conditions and finite convergence of lasserre's hierarchy," *Mathematical Programming*, vol. 146, no. 1, pp. 97–121, 2013.
[16] P. A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, Citeseer, 2000.
[17] J. B. Lasserre, *Moments, positive polynomials and their applications*, vol. 1. London: World Scientific, 2009.
[18] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, "Introducing sostools: A general purpose sum of squares programming solver," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 1, pp. 741–746, IEEE, 2002.
[19] D. Henrion and J.-B. Lasserre, *Positive Polynomials in Control*, ch. Detecting Global Optimality and Extracting Solutions in GloptiPoly, pp. 293–310. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.
[20] M. Schweighofer, "Optimization of polynomials on compact semialgebraic sets," *SIAM Journal on Optimization*, vol. 15, no. 3, pp. 805–825, 2005.
[21] J. Nie and M. Schweighofer, "On the complexity of putinar's positivstellensatz," *Journal of Complexity*, vol. 23, no. 1, pp. 135 – 150, 2007.
[22] G. Stengle, "A nullstellensatz and a positivstellensatz in semialgebraic geometry," *Mathematische Annalen*, vol. 207, no. 2, pp. 87–97, 1974.