



HAL
open science

AN APPROACH TO CONSTRAINED POLYNOMIAL OPTIMIZATION VIA NONNEGATIVE CIRCUIT POLYNOMIALS AND GEOMETRIC PROGRAMMING

Mareike Dressler, Sadik Iliman, Timo de Wolff

► **To cite this version:**

Mareike Dressler, Sadik Iliman, Timo de Wolff. AN APPROACH TO CONSTRAINED POLYNOMIAL OPTIMIZATION VIA NONNEGATIVE CIRCUIT POLYNOMIALS AND GEOMETRIC PROGRAMMING. MEGA 2017 - International Conference on Effective Methods in Algebraic Geometry, Jun 2017, Nice, France. hal-01829714

HAL Id: hal-01829714

<https://inria.hal.science/hal-01829714>

Submitted on 4 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AN APPROACH TO CONSTRAINED POLYNOMIAL OPTIMIZATION VIA NONNEGATIVE CIRCUIT POLYNOMIALS AND GEOMETRIC PROGRAMMING

MAREIKE DRESSLER, SADIK ILIMAN, AND TIMO DE WOLFF

ABSTRACT. In this article we combine two developments in polynomial optimization. On the one hand, we consider nonnegativity certificates based on sums of nonnegative circuit polynomials, which were recently introduced by the second and the third author. On the other hand, we investigate geometric programming methods for constrained polynomial optimization problems, which were recently developed by Ghasemi and Marshall. We show that the combination of both results yields a new method to solve certain classes of constrained polynomial optimization problems. We test the new method experimentally and compare it to semidefinite programming in various examples.

1. INTRODUCTION

Solving polynomial optimization problems is a key challenge in countless applications like dynamical systems, robotics, control theory, computer vision, signal processing, and economics; e.g. [BPT13, Las10]. It is well-known that polynomial optimization problems are NP-hard in general both in the constrained and in the unconstrained case [DG14]. Starting with the seminal work of Lasserre in [Las01], relaxation methods were developed which are significantly faster and provide lower bounds. These methods were studied intensively by means of aspects like exactness and quality of the relaxations [dKL10, Nie13a, Nie13b, Nie14], the speed of the computations [Las10, PS03], and geometrical aspects of the underlying structures [Ble06, Ble12]. A great majority of these results are based on the original approach by Lasserre, called *Lasserre relaxation*, which relies on *semidefinite programming (SDP)* methods and *sums of squares (SOS)* certificates to provide lower bounds for polynomial optimization problems. SDPs can be solved in polynomial time in problem size (up to an ε -error); e.g. [BPT13, p. 41] and references therein. However, the size of such programs grows exponentially with the number of variables n or the degree d of the polynomials, as its size is given by the number of monomials of n -variate monomials of degree at most d .

Recently, Ghasemi and Marshall suggested a promising alternative approach both for constrained and unconstrained optimization problems based on *geometric programming (GP)* [GM12, GM13]. GPs can also be solved in polynomial time (up to an ε -error) [NN94]; see also [BKVH07, Page 118], but, by experimental results, e.g. [BKVH07, GM12, GM13, GLM14], in practice the corresponding geometric programs can be solved *significantly*

2010 *Mathematics Subject Classification.* 12D15, 14P99, 52B20, 90C25.

Key words and phrases. Certificate, geometric programming, nonnegative polynomial, semidefinite programming, sum of nonnegative circuit polynomials, sum of squares, triangulation.

faster than their counterparts in semidefinite programming. The lower bounds obtained by Ghasemi and Marshall are, however, by construction worse than lower bounds obtained via semidefinite programming, and they can only be applied in very special cases.

Independent of Ghasemi and Marshall, the second and the third author recently developed a new certificate for nonnegativity of real polynomials called *sums of nonnegative circuit polynomials (SONC)* [IdW16a]. SONC certificates are independent of SOS certificates. In [IdW16b] the second and third author showed that the GP based approach for unconstrained optimization by Ghasemi and Marshall can be generalized crucially via SONC certificates. In consequence, the presented geometric programs are linked to sums of nonnegative circuit polynomials similarly as semidefinite programming relaxations are linked to sums of squares. Particularly, there exist various classes of polynomials for which the GP/SONC based approach is not only *faster* but, it also yields *better* bounds than the SDP/SOS approach. The reason is that all certificates used by Ghasemi and Marshall are always SOS, while SONCs are not SOS in general; see [IdW16a, Proposition 7.2].

The first contribution of this article is an extension of the results in [IdW16b] to constrained polynomial optimization problems. We focus on the class of *ST-polynomials*, that are polynomials which have a Newton polytope that is a simplex and which are satisfying some further conditions; see Section 2.1. The starting point is a general optimization problem from [IdW16b, Section 5], see (2.6), which provides a lower bound for the constrained problem but which is not a geometric program. Using results from [GM13], we relax the program (2.6) into a geometric optimization problem; see program (3.2) and Theorem 3.1. Additionally, we show in Theorem 3.4 that (2.6) can always at least be transformed into a signomial program; see Section 2.2 for background information. Furthermore, we prove that the new, relaxed geometric program (3.2) provides bounds as good as the initial program (2.6) for certain special cases, see Theorem 3.5.

In Section 4, we provide examples testing our new program (3.2) in practice and comparing it with semidefinite programming. Moreover, we demonstrate that increasing the degree of a given problem has almost *no* effect on the runtime of our program (3.2). This is in sharp contrast to SDPs, where one can nullify increased runtimes induced by high degrees only by additional pre-processing methods, e.g. by exploiting sparsity.

Furthermore, a bound obtained by Ghasemi and Marshall in [GM13] can never be better than the bound given by the d -th Lasserre relaxation for some specific d determined by the degrees of the involved polynomials. In Section 4 we provide examples showing that our program (3.2) can provide bounds which are better than the particularly d -th Lasserre relaxations.

The second contribution of this article is to apply polynomial optimization methods based on SONCs and GPs efficiently *beyond* the class of ST-polynomials. In Section 5, we develop an initial approach based on triangulations of support sets of the involved polynomials. It yields bounds for nonnegativity based on SONC/GP for *arbitrary* polynomials both in the constrained and in the unconstrained case. We provide several examples and

compare the new bounds to the ones obtained by SDP based methods.

Acknowledgements. We thank Henning Seidler for various comments and his support on some of the computations. We also thank the anonymous referees, who helped us to improve the article significantly.

The third author was partially supported by the DFG grant WO 2206/1-1.

2. PRELIMINARIES

In this section we recall key results about sums of nonnegative circuit polynomials (SONCs) and geometric programming (GP), which are used in this article.

2.1. The Cone of Sums of Nonnegative Circuit Polynomials. We denote by $\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \dots, x_n]$ the vector space of real n -variate polynomials. Let δ_{ij} be the ij -Kronecker symbol, let $\mathbf{e}_i = (\delta_{i1}, \dots, \delta_{in})$ be the i -th standard vector, and let $A \subset \mathbb{N}^n$ be a finite set. We denote by $\text{conv}(A)$ the convex hull of A and by $V(A)$ the vertices of $\text{conv}(A)$. We consider polynomials $f \in \mathbb{R}[\mathbf{x}]$ supported on A . That is, f is of the form $f(\mathbf{x}) = \sum_{\alpha \in A} f_{\alpha} \mathbf{x}^{\alpha}$ with $f_{\alpha} \in \mathbb{R}$, $\mathbf{x}^{\alpha} = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. We call a lattice point *even* if it is in $(2\mathbb{N})^n$. Furthermore, we denote the Newton polytope of f as $\text{New}(f) = \text{conv}\{\alpha \in \mathbb{N}^n : f_{\alpha} \neq 0\}$.

For a given $A \subset \mathbb{N}^n$ we define $\Delta(A) = A \setminus V(A)$. Let f be as before. We denote by $\Delta(f)$ the elements of $\Delta(A)$ which appear as exponents of non-zero terms, that are no monomial squares. I.e., we have

$$\Delta(f) = \{\alpha \in \Delta(A) : f_{\alpha} \neq 0 \text{ and } (f_{\alpha} < 0 \text{ or } \alpha \notin (2\mathbb{N})^n)\}.$$

A polynomial is nonnegative on the entire \mathbb{R}^n only if the following necessary conditions are satisfied; see e.g. [Rez78].

Proposition 2.1. *Let $A \subset \mathbb{N}^n$ be a finite set and $f \in \mathbb{R}[\mathbf{x}]$ be supported on A such that $\text{New}(f) = \text{conv}(A)$. Then f is nonnegative on \mathbb{R}^n only if:*

- (1) *All elements of $V(A)$ are even.*
- (2) *If $\alpha \in V(A)$, then the corresponding coefficient f_{α} is strictly positive.*

In other words, if $\alpha \in V(A)$, then the term $f_{\alpha} \mathbf{x}^{\alpha}$ has to be a monomial square.

The statement remains true for real Laurent polynomials $g \in \mathbb{R}[\mathbf{x}^{\pm 1}] = \mathbb{R}[x_1^{\pm 1}, \dots, x_n^{\pm 1}]$, since we can consider g as a polynomial f divided by a monomial square \mathbf{x}^{α} for an even α ; this is of relevance in Section 5. For the remainder of the article, we assume that these necessary conditions in Proposition 2.1 are satisfied including $\text{New}(f) = \text{conv}(A)$. For simplicity, we denote this assumption by the symbol \clubsuit from now on.

In what follows we consider the class of *ST-polynomials*. For further details about the following objects defined in this section see [dW15, IdW16a, IdW16b]; see also [FK11, GM12, GM13].

Definition 2.2. Let $f \in \mathbb{R}[\mathbf{x}]$ be supported on $A \subset \mathbb{N}^n$ such that (\clubsuit) holds. Then f is called an *ST-polynomial* if it is of the form

$$(2.1) \quad f(\mathbf{x}) = \sum_{j=0}^r f_{\alpha(j)} \mathbf{x}^{\alpha(j)} + \sum_{\beta \in \Delta(A)} f_{\beta} \mathbf{x}^{\beta},$$

with $r \leq n$, exponents $\alpha(j)$ and β , and coefficients $f_{\alpha(j)}, f_{\beta}$, for which the following conditions hold:

- (ST1): The points $\alpha(0), \alpha(1), \dots, \alpha(r)$ are affinely independent and equal $V(A)$.
 (ST2): Every exponent $\beta \in \Delta(A)$ can be written uniquely as

$$\beta = \sum_{j=0}^r \lambda_j^{(\beta)} \alpha(j) \quad \text{with } \lambda_j^{(\beta)} \geq 0 \quad \text{and} \quad \sum_{j=0}^r \lambda_j^{(\beta)} = 1.$$

◻

Note that hypotheses (ST1) and (ST2) imply that $V(A) = \{\alpha(0), \dots, \alpha(r)\}$ is the vertex set of an r -dimensional simplex. By the assumption (\clubsuit) it consists of even lattice points, and it coincides with $\text{New}(f) = \text{conv}(A)$. The $\lambda_j^{(\beta)}$ denote the barycentric coordinates of β relative to the vertices $\alpha(j)$ with $j = 0, \dots, r$. The ‘‘ST’’ in ‘‘ST-polynomial’’ is short for ‘‘simplex tail’’. The tail part is given by the sum $\sum_{\beta \in \Delta(A)} f_{\beta} \mathbf{x}^{\beta}$, while the other terms define the simplex part. If an ST-polynomial f has a tail part consisting of at most one term, then we call f a *circuit polynomial*.

Nonnegativity of ST-polynomials is closely related to an invariant called the *circuit number*.

Definition 2.3. Let f be an ST-polynomial with support set A . For every $\beta \in \Delta(A)$ we define the corresponding *circuit number* as

$$(2.2) \quad \Theta_f(\beta) = \prod_{j \in \text{nz}(\beta)} \left(\frac{f_{\alpha(j)}}{\lambda_j^{(\beta)}} \right)^{\lambda_j^{(\beta)}}$$

with $\text{nz}(\beta) = \{j \in \{0, \dots, r\} : \lambda_j^{(\beta)} \neq 0\}$, $f_{\alpha(j)}$, and $\lambda_j^{(\beta)}$ as before. ◻

The terms ‘‘circuit polynomial’’ and ‘‘circuit number’’ are chosen since β and the $\alpha(j)$ with $j \in \text{nz}(\beta)$ form a *circuit*; this is a minimally affine dependent set, see e.g. [Oxl11].

A fundamental fact is that nonnegativity of a circuit polynomial f can be decided by comparing its tail coefficient f_{β} with its corresponding circuit number $\Theta_f(\beta)$ alone.

Theorem 2.4 ([IdW16a], Theorem 3.8). *Let f be a circuit polynomial with unique tail term $f_{\beta} \mathbf{x}^{\beta}$ and let $\Theta_f(\beta)$ be the corresponding circuit number, as defined in (2.2). Then the following statements are equivalent:*

- (1) f is nonnegative.
- (2) $|f_{\beta}| \leq \Theta_f(\beta)$ and $\beta \notin (2\mathbb{N})^n$ or $f_{\beta} \geq -\Theta_f(\beta)$ and $\beta \in (2\mathbb{N})^n$.

Note that (2) can be equivalently stated as: $|f_\beta| \leq \Theta_f(\beta)$ or f is a sum of monomial squares.

Writing a polynomial as a sum of nonnegative circuit polynomials is a certificate of nonnegativity. We denote by **SONC** the class of polynomials that are *sums of nonnegative circuit polynomials* or the property of a polynomial to be in this class.

2.2. Geometric Programming. Geometric programming was introduced in [DPZ67]. It is a convex optimization problem and has applications for example in nonlinear network flow problems, optimal control, optimal location problems, chemical equilibrium problems and particularly in circuit design problems.

Definition 2.5. A function $p : \mathbb{R}_{>0}^n \rightarrow \mathbb{R}$ of the form $p(\mathbf{z}) = p(z_1, \dots, z_n) = cz_1^{\alpha_1} \cdots z_n^{\alpha_n}$ with $c > 0$ and $\alpha_i \in \mathbb{R}$ is called a *monomial (function)*. A sum $\sum_{i=0}^k c_i z_1^{\alpha_1(i)} \cdots z_n^{\alpha_n(i)}$ of monomials with $c_i > 0$ is called a *posynomial (function)*.

A *geometric program (GP)* has the following form:

$$(2.3) \quad \begin{cases} \text{minimize} & p_0(\mathbf{z}), \\ \text{subject to:} & (1) \ p_i(\mathbf{z}) \leq 1 \text{ for all } 1 \leq i \leq m, \\ & (2) \ q_j(\mathbf{z}) = 1 \text{ for all } 1 \leq j \leq l, \end{cases}$$

where p_0, \dots, p_m are posynomials and q_1, \dots, q_l are monomial functions. \square

Geometric programs can be solved with interior point methods. In [NN94], the authors prove worst-case polynomial time complexity of this method; see also [BKVH07, Page 118]. A *signomial program* is given like a geometric program except that the coefficients c_i of the involved posynomials can be arbitrary real numbers.

For an introduction to geometric programming, signomial programming, and an overview about applications see [BKVH07, BV04].

2.3. SONC Certificates via Geometric Programming in the Unconstrained Case. In this section we recall the main results from [IdW16b] about SONC certificates obtained via geometric programming for unconstrained polynomial optimization problems. These results always require that the polynomial in the optimization problem is an ST-polynomial in the sense of Section 2.1.

Theorem 2.6. ([IdW16b, Theorems 3.4 and 3.5]) *Assume that f is an ST-polynomial as in (2.1) with $\alpha(0) = 0$ and let $k \in \mathbb{R}$. Suppose that for every $(\beta, j) \in \Delta(f) \times \{1, \dots, r\}$ there exists an $a_{\beta,j} \geq 0$, such that:*

- (1) $a_{\beta,j} > 0$ if and only if $\lambda_j^{(\beta)} > 0$,
- (2) $|f_\beta| \leq \prod_{j \in \text{nz}(\beta)} \left(\frac{a_{\beta,j}}{\lambda_j^{(\beta)}} \right)^{\lambda_j^{(\beta)}}$ for every $\beta \in \Delta(f)$ with $\lambda_0^{(\beta)} = 0$,
- (3) $f_{\alpha(j)} \geq \sum_{\beta \in \Delta(f)} a_{\beta,j}$ for all $1 \leq j \leq r$,
- (4) $f_{\alpha(0)} - k \geq \sum_{\substack{\beta \in \Delta(f) \\ \lambda_0^{(\beta)} \neq 0}} \lambda_0^{(\beta)} |f_\beta|^{1/\lambda_0^{(\beta)}} \prod_{\substack{j \in \text{nz}(\beta) \\ j \geq 1}} \left(\frac{\lambda_j^{(\beta)}}{a_{\beta,j}} \right)^{\lambda_j^{(\beta)}/\lambda_0^{(\beta)}}.$

Then $f - k$ is a sum of nonnegative circuit polynomials g_1, \dots, g_s such that $s := |\Delta(f)|$, and for every g_i the Newton polytope $\text{New}(g_i)$ is a face of $\text{New}(f)$.

Let f_{sonc} be the supremum of all $k \in \mathbb{R}$ such that for every $\beta \in \Delta(f)$ there exist nonnegative reals $a_{\beta,1}, \dots, a_{\beta,r}$ such that the conditions (1) to (4) are satisfied. Then f_{sonc} coincides with the supremum of all $k \in \mathbb{R}$ such that there exist nonnegative circuit polynomials g_1, g_2, \dots, g_s whose Newton polytopes are faces of $\text{New}(f)$ and which satisfy $f - k = \sum_{i=1}^s g_i$.

We remark that Theorem 2.6 remains valid if all terms of an ST-polynomial are multiplied by a monomial square $\mathbf{x}^{\alpha(0)} \in (2\mathbb{N})^n$. In this case, condition needs to be replaced by

$$(4') \quad (f_{\alpha(0)} - k)\mathbf{x}^{\alpha(0)} \geq \sum_{\substack{\beta \in \Delta(f) \\ \lambda_0^{(\beta)} \neq 0}} \lambda_0^{(\beta)} |f_{\beta}|^{1/\lambda_0^{(\beta)}} \prod_{\substack{j \in \text{nz}(\beta) \\ j \geq 1}} \left(\frac{\lambda_j^{(\beta)}}{a_{\beta,j}} \right)^{\lambda_j^{(\beta)}/\lambda_0^{(\beta)}}.$$

We then obtain a nonnegativity bound for the (positive) the coefficient of the term $\mathbf{x}^{\alpha(0)}$ instead of the constant term. We apply this slightly more general version of Theorem 2.6 in Section 5.

For the special case of scaled standard simplices Theorem 2.6 was shown earlier by Ghasemi and Marshall [GM12, Theorem 3.1]. In this special case every sum of nonnegative circuit polynomials is also a sum of binomial squares which is not true in general. For example, the Motzkin polynomial is an ST-polynomial with one interior term, which is not even a SOS.

Theorem 2.6 states

$$f_{\text{sonc}} = \sup\{k \in \mathbb{R} : f - k\mathbf{x}^{\alpha(0)} \text{ is a SONC}\}.$$

The bound f_{sonc} is given by a geometric program [IdW16b, Corollary 4.2]:

Corollary 2.7. *Let $f \in \mathbb{R}[\mathbf{x}]$ be an ST-polynomial. Let R be the subset of an $r|\Delta(f)|$ -dimensional real space given by*

$$R = \{(a_{\beta,j}) : a_{\beta,j} \in \mathbb{R}_{>0} \text{ for every } \beta \in \Delta(f) \text{ and } j \in \text{nz}(\beta)\}.$$

Then $f_{\text{sonc}} = f_{\alpha(0)} - m^*$, where m^* is given as the output of the following geometric program:

$$\left\{ \begin{array}{l} \text{minimize} \quad \sum_{\substack{\beta \in \Delta(f) \\ \lambda_0^{(\beta)} \neq 0}} \lambda_0^{(\beta)} |f_{\beta}|^{1/\lambda_0^{(\beta)}} \prod_{\substack{j \in \text{nz}(\beta) \\ j \geq 1}} \left(\frac{\lambda_j^{(\beta)}}{a_{\beta,j}} \right)^{\lambda_j^{(\beta)}/\lambda_0^{(\beta)}} \quad \text{over the subset } R' \text{ of } R \\ \\ \text{defined by:} \\ (1) \quad \sum_{\beta \in \Delta(f)} (a_{\beta,j}/f_{\alpha(j)}) \leq 1 \text{ for every } 1 \leq j \leq r, \\ (2) \quad |f_{\beta}| \prod_{j \in \text{nz}(\beta)} \left(\frac{\lambda_j^{(\beta)}}{a_{\beta,j}} \right)^{\lambda_j^{(\beta)}} \leq 1 \text{ for every } \beta \in \Delta(f) \text{ with } \lambda_0^{(\beta)} = 0. \end{array} \right.$$

Hence, the optimal bound to find a SONC decomposition of an ST-polynomial is provided by geometric programming. Since a polynomial with a SONC decomposition is nonnegative, geometric programming can be used to find certificates of nonnegativity.

Following the literature, e.g. [BPT13, Lau09], we define a *global polynomial optimization problem* for some $f \in \mathbb{R}[\mathbf{x}]$ as the problem to determine the real number

$$f^* = \inf\{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\} = \sup\{\lambda \in \mathbb{R} : f - \lambda \geq 0\}.$$

One can find a lower bound for f^* by relaxing the nonnegativity condition in the above problem to finding the real number

$$f_{\text{sos}} = \sup \left\{ \lambda \in \mathbb{R} : f - \lambda = \sum_{i=1}^k q_i^2 \text{ for some } q_i \in \mathbb{R}[\mathbf{x}] \right\}.$$

The bound f_{sos} for the optimal SOS decomposition of f can be determined by semidefinite programming. By construction, we have $f_{\text{sos}} \leq f^*$; see [Las10].

A key observation is that the bounds obtained by this approach can be better than the ones obtained by SDP as the following result shows; see [IdW16b, Corollary 3.6].

Corollary 2.8. *Let f be an ST-polynomial with $\Delta(A) = \Delta(f)$ such that $\Delta(f)$ is contained in the interior of $\text{New}(f)$. Let $\boldsymbol{\alpha}(0)$ be the origin and suppose that there exists a vector $\mathbf{v} \in (\mathbb{R}^*)^n$ such that $f_{\boldsymbol{\alpha}} \cdot \mathbf{v}^{\boldsymbol{\alpha}} < 0$ for all $\boldsymbol{\alpha} \in \Delta(f)$. Then*

$$f_{\text{sonc}} = f^* \geq f_{\text{sos}}.$$

2.4. SONC Certificates for the Constrained Case. In this subsection we restate facts from [IdW16b, Section 5] about SONC certificates applied to constrained polynomial optimization problems.

Let f, g_1, \dots, g_s be elements of the polynomial ring $\mathbb{R}[\mathbf{x}]$ and let

$$K = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \geq 0 \text{ for all } i = 1, \dots, s\}$$

be a basic closed semialgebraic set defined by g_1, \dots, g_s . We consider the constrained polynomial optimization problem

$$f_K^* = \inf_{\mathbf{x} \in K} f(\mathbf{x}) = \sup\{\gamma \in \mathbb{R} : f(\mathbf{x}) - \gamma \geq 0 \text{ for all } \mathbf{x} \in K\}.$$

If $s = 0$, then we have no g_i and therefore $K = \mathbb{R}^n$, which leads to the global optimization problem explained in Section 2.3.

To obtain a general lower bound for f on K which is computable by geometric programming we replace the considered polynomials by a new function. Let

$$(2.4) \quad G(\boldsymbol{\mu})(\mathbf{x}) = f(\mathbf{x}) - \sum_{i=1}^s \mu_i g_i(\mathbf{x}) = - \sum_{i=0}^s \mu_i g_i(\mathbf{x})$$

for $\boldsymbol{\mu} = (\mu_1, \dots, \mu_s) \in \mathbb{R}_{\geq 0}^s$, $g_0 = -f$ and $\mu_0 = 1$. For every fixed $\boldsymbol{\mu}^* \in \mathbb{R}_{\geq 0}^s$ the function $G(\mathbf{x}) = G(\boldsymbol{\mu}^*)(\mathbf{x})$ is a polynomial in $\mathbb{R}[\mathbf{x}]$. Following an argument in [GM13] we can assume that all monomial squares of $-g_i$ are vertices of $\text{New}(G(\boldsymbol{\mu}))$: One can reduce to this case by neglecting all monomial squares not corresponding to such a vertex. That is,

for all $i = 0, \dots, s$ one can replace g_i by \tilde{g}_i , which resemble g_i without monomial squares of $-g_i$ in the interior of $\text{New}(G(\boldsymbol{\mu}))$. Then $-\tilde{g}_i \leq -g_i$ on \mathbb{R}^n for $i = 0, \dots, s$, thus, $K \subseteq \tilde{K}$, where $\tilde{K} = \{\mathbf{x} \in \mathbb{R}^n : \tilde{g}_i(\mathbf{x}) \geq 0, 1 \leq i \leq s\}$, as well as $f_{\tilde{K}}^* \leq f_K^*$.

Let $A_i \subset \mathbb{N}^n$ be the support of the polynomial g_i for $i = 0, \dots, s$ and let $A = \bigcup_{i=0}^s A_i$ be the union of all supports of polynomials g_i . We remark that while we consider a fixed support the Newton polytope of $G(\boldsymbol{\mu})$ is not invariant in general since certain μ_i might equal 0 or term cancellation might occur. If for some $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^s$ the polynomial $G(\boldsymbol{\mu})$ is an ST-polynomial, then we assume that $\text{New}(G(\boldsymbol{\mu})) = \text{conv}(A)$ and $V(A) = \{\boldsymbol{\alpha}(0), \dots, \boldsymbol{\alpha}(r)\} \subset (2\mathbb{N})^n$ and we denote $G(\boldsymbol{\mu})_{\text{sonc}}$ as the optimal value of the geometric program in Corollary 2.7. Theorem 2.6 implies that $G(\boldsymbol{\mu}) - G(\boldsymbol{\mu})_{\text{sonc}} \mathbf{x}^{\boldsymbol{\alpha}(0)} \geq 0$ and $G(\boldsymbol{\mu})_{\text{sonc}} \in \mathbb{R}$ is the maximal possible choice for nonnegativity. Hence, we obtain a bound for the coefficient of the term $\mathbf{x}^{\boldsymbol{\alpha}(0)}$ depending on the other coefficients of $G(\boldsymbol{\mu})$ certifying nonnegativity of $G(\boldsymbol{\mu})$. If $G(\boldsymbol{\mu})$ is not an ST-polynomial for some $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^s$, then we set $G(\boldsymbol{\mu})_{\text{sonc}} = -\infty$, since the corresponding geometric program is infeasible. Thus, by (2.4), if $\boldsymbol{\mu}$ is fixed, then $G(\boldsymbol{\mu})_{\text{sonc}}$ is a lower bound for f on the semialgebraic set K regarding the coefficient of $\mathbf{x}^{\boldsymbol{\alpha}(0)}$. Let $\mathbf{g} = (g_1, \dots, g_s)$. We define

$$s(f, \mathbf{g}) = \sup\{G(\boldsymbol{\mu})_{\text{sonc}} : \boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^s\}.$$

Thus, we have particularly for $\boldsymbol{\alpha}(0) = \mathbf{0}$:

$$(2.5) \quad s(f, \mathbf{g}) \leq f_K^*.$$

For every fixed $\boldsymbol{\mu}$ the bound $G(\boldsymbol{\mu})_{\text{sonc}}$ is computable by a geometric program. Unfortunately, this does not imply that the supremum is computable by a GP as well. However, following ideas by Ghasemi and Marshall [GM12] the second and third author presented a general optimization program for a lower bound of $s(f, \mathbf{g})$ in [IdW16b], which is a geometric program under special conditions. We recall these results in what follows.

We define $\Delta(A)$ in the sense of Section 2.1 as the set of exponents of the tail terms of $G(\boldsymbol{\mu})$ and $\Delta(G(\boldsymbol{\mu})) \subseteq \Delta(A)$ as the set of exponents which have a non-zero coefficient and are not a monomial square. Moreover, we define $\Delta(G) = \bigcup_{\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^s} \Delta(G(\boldsymbol{\mu}))$. Note that $\Delta(G(\boldsymbol{\mu})) \subseteq \Delta(G) \subseteq \Delta(A)$ for all $\boldsymbol{\mu}$. We have by Section 2.1, Definition 2.2

$$G(\boldsymbol{\mu})(\mathbf{x}) = - \sum_{i=0}^s \mu_i g_i(\mathbf{x}) = \sum_{j=0}^r G(\boldsymbol{\mu})_{\boldsymbol{\alpha}(j)} \mathbf{x}^{\boldsymbol{\alpha}(j)} + \sum_{\boldsymbol{\beta} \in \Delta(G)} G(\boldsymbol{\mu})_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$$

with coefficients $G(\boldsymbol{\mu})_{\boldsymbol{\alpha}(j)}, G(\boldsymbol{\mu})_{\boldsymbol{\beta}} \in \mathbb{R}$ depending on $\boldsymbol{\mu}$. We set the coefficients $G(\boldsymbol{\mu})_{\boldsymbol{\beta}} = 0$ for all $\boldsymbol{\beta} \in \Delta(G) \setminus \Delta(G(\boldsymbol{\mu}))$.

As before, we denote by $\{\lambda_0^{(\boldsymbol{\beta})}, \dots, \lambda_r^{(\boldsymbol{\beta})}\}$ the barycentric coordinates of the lattice point $\boldsymbol{\beta} \in \Delta(A)$ with respect to the vertices of the simplex $\text{New}(G(\boldsymbol{\mu})) = \text{conv}(A)$. We define for every $\boldsymbol{\beta} \in \Delta(G)$ a set

$$R_{\boldsymbol{\beta}} = \{\mathbf{a}_{\boldsymbol{\beta}} : \mathbf{a}_{\boldsymbol{\beta}} = (a_{\boldsymbol{\beta},1}, \dots, a_{\boldsymbol{\beta},r}) \in \mathbb{R}_{>0}^r\}.$$

Furthermore, we define the nonnegative real set R as

$$R = [0, \infty)^s \times \bigtimes_{\boldsymbol{\beta} \in \Delta(G)} (R_{\boldsymbol{\beta}} \times \mathbb{R}_{\geq 0}).$$

Hence, R is the Cartesian product of $[0, \infty)^s$ and $|\Delta(G)|$ many copies $\mathbb{R}_{>0}^r \times \mathbb{R}_{\geq 0}$; each given by one R_β with $\beta \in \Delta(G)$ and one $\mathbb{R}_{\geq 0}$. We define the function p from R to $\mathbb{R}_{\geq 0}$ as

$$p(\boldsymbol{\mu}, \{(\mathbf{a}_\beta, b_\beta) : \beta \in \Delta(G)\}) = \sum_{i=1}^s \mu_i g_{i, \alpha(0)} + \sum_{\substack{\beta \in \Delta(G) \\ \lambda_0^{(\beta)} \neq 0}} \lambda_0^{(\beta)} \cdot b_\beta^{\frac{1}{\lambda_0^{(\beta)}}} \cdot \prod_{\substack{j \in \text{nz}(\beta) \\ j \geq 1}} \left(\frac{\lambda_j^{(\beta)}}{a_{\beta, j}} \right)^{\frac{\lambda_j^{(\beta)}}{\lambda_0^{(\beta)}}}$$

where, as before, $\alpha(0)$ is a vertex of $\text{New}(G(\boldsymbol{\mu}))$ and $g_{i, \alpha(0)}$ is the coefficient of the monomial $\mathbf{x}^{\alpha(0)}$ in the polynomial g_i .

For the coefficient $G(\boldsymbol{\mu})_\beta$ of the term with exponent β of $G(\boldsymbol{\mu})$ we use the notation $G(\boldsymbol{\mu})_\beta = -\sum_{i=0}^s \mu_i \cdot g_{i, \beta}$. In other words, $G(\boldsymbol{\mu})_\beta$ is a linear form in the μ_i 's given by the coefficients of the polynomials g_i ; analogously for $G(\boldsymbol{\mu})_{\alpha(j)}$. We consider the following optimization problem:

$$(2.6) \left\{ \begin{array}{l} \text{minimize} \quad p(\boldsymbol{\mu}, \{(\mathbf{a}_\beta, b_\beta) : \beta \in \Delta(G)\}) \text{ over the subset of } R \\ \\ (1) \quad \sum_{\beta \in \Delta(G)} a_{\beta, j} \leq G(\boldsymbol{\mu})_{\alpha(j)} \text{ for all } 1 \leq j \leq r, \\ \\ \text{defined by:} \quad (2) \quad \prod_{j \in \text{nz}(\beta)} \left(\frac{a_{\beta, j}}{\lambda_j^{(\beta)}} \right)^{\lambda_j^{(\beta)}} \geq b_\beta \text{ for every } \beta \in \Delta(G) \text{ with } \lambda_0^{(\beta)} = 0, \text{ and} \\ \\ (3) \quad |G(\boldsymbol{\mu})_\beta| \leq b_\beta \text{ for every } \beta \in \Delta(G) \text{ with } \lambda_0^{(\beta)} \neq 0. \end{array} \right.$$

In [IdW16b, Theorems 5.1 and 5.2] the second and third author show the following theorem.

Theorem 2.9. *Let γ be the optimal value of the optimization problem (2.6). Then we have $f_{\alpha(0)} - \gamma \leq s(f, \mathbf{g})$. The optimization problem (2.6) restricted to $\boldsymbol{\mu} \in (0, \infty)^s$ is a signomial program if for every $\beta \in \Delta(G)$ it holds that $G(\boldsymbol{\mu})_\beta$ has the same sign for every choice of $\boldsymbol{\mu}$.*

Assume additionally that every linear form $G(\boldsymbol{\mu})_{\alpha(j)} = -\sum_{i=0}^s \mu_i \cdot g_{i, \alpha(j)}$ corresponding to a vertex $\alpha(j)$ of $\text{New}(G(\boldsymbol{\mu}))$ has only one summand and is strictly positive. Assume moreover that for all $\beta \in \Delta(G)$ the linear form $G(\boldsymbol{\mu})_\beta = -\sum_{i=0}^s \mu_i \cdot g_{i, \beta}$ has only positive terms. If furthermore all $g_{i, \alpha(0)}$ for $1 \leq i \leq s$ are greater than or equal to zero, then (2.6) is a geometric program.

3. CONSTRAINED POLYNOMIAL OPTIMIZATION VIA SIGNOMIAL AND GEOMETRIC PROGRAMMING

In this section, we provide relaxations of the program (2.6) following ideas of Ghasemi and Marshall in [GM13]. The goal is to weaken the assumptions which are needed to obtain a geometric program or at least a signomial program. We provide such relaxations in the programs (3.2) and (3.3) and provide the desired properties in the Theorems 3.1 and 3.4. Moreover, we show that under certain extra assumptions the bound obtained by

the new program (3.2) equals the optimal bound $s(f, \mathbf{g})$ from the previous section; see Theorem 3.5.

Let all notation regarding $G(\boldsymbol{\mu})$ be given as in Section 2.4. Assume that we have for each $0 \leq i \leq s$

$$g_i = \sum_{\boldsymbol{\beta} \in A_i} g_{i,\boldsymbol{\beta}} \cdot \mathbf{x}^{\boldsymbol{\beta}}$$

with $g_{i,\boldsymbol{\beta}} \in \mathbb{R}$. We have $\Delta(A_i) \subseteq \Delta(A)$ and hence write

$$g_i = \sum_{j=0}^r g_{i,\boldsymbol{\alpha}(j)} \mathbf{x}^{\boldsymbol{\alpha}(j)} + \sum_{\boldsymbol{\beta} \in \Delta(A)} g_{i,\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$$

and set $g_{i,\boldsymbol{\alpha}(j)} = 0$ for all $\boldsymbol{\alpha}(j) \in V(A) \setminus A_i$ and $g_{i,\boldsymbol{\beta}} = 0$ for all $\boldsymbol{\beta} \in \Delta(A) \setminus A_i$. We remark that three cases can occur for $\boldsymbol{\beta} \in \Delta(A) \cap A_i$:

- (1) $-g_{i,\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$ is not a monomial square. Then we have $\boldsymbol{\beta} \in \Delta(G)$.
- (2) $-g_{i,\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$ is a monomial square, but there exists another g_l such that $-g_{l,\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$ is not a monomial square. Then we have $\boldsymbol{\beta} \in \Delta(G)$.
- (3) $-g_{i,\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$ is a monomial square, and there exists no other g_l such that $-g_{l,\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$ is not a monomial square. Then we have $\boldsymbol{\beta} \notin \Delta(G)$.

Sums of monomial squares as described in case (3) are ignored in our program (2.6). Hence, we can also ignore this case here. We investigate the other two cases in detail now. As already mentioned in Section 2.4 we can interpret the coefficients $G(\boldsymbol{\mu})_{\boldsymbol{\alpha}(j)}$ and $G(\boldsymbol{\mu})_{\boldsymbol{\beta}}$ as linear forms in $\boldsymbol{\mu}$ since we have for all $j = 0, \dots, r$

$$G(\boldsymbol{\mu})_{\boldsymbol{\alpha}(j)} = - \sum_{i=0}^s \mu_i \cdot g_{i,\boldsymbol{\alpha}(j)} \quad \text{and} \quad G(\boldsymbol{\mu})_{\boldsymbol{\beta}} = - \sum_{i=0}^s \mu_i \cdot g_{i,\boldsymbol{\beta}}.$$

We decompose every $G(\boldsymbol{\mu})_{\boldsymbol{\beta}}$ into a positive and a negative part such that $G(\boldsymbol{\mu})_{\boldsymbol{\beta}} = G(\boldsymbol{\mu})_{\boldsymbol{\beta}}^+ - G(\boldsymbol{\mu})_{\boldsymbol{\beta}}^-$, where

$$(3.1) \quad G(\boldsymbol{\mu})_{\boldsymbol{\beta}}^- = \sum_{g_{i,\boldsymbol{\beta}} > 0} \mu_i \cdot g_{i,\boldsymbol{\beta}} \quad \text{and} \quad G(\boldsymbol{\mu})_{\boldsymbol{\beta}}^+ = - \sum_{g_{i,\boldsymbol{\beta}} < 0} \mu_i \cdot g_{i,\boldsymbol{\beta}}.$$

This decomposition is *independent* of the choice of $\boldsymbol{\mu}$ in the sense that no $g_{i,\boldsymbol{\beta}}$ can be a summand of *both* $G(\boldsymbol{\mu})_{\boldsymbol{\beta}}^+$ and $G(\boldsymbol{\mu})_{\boldsymbol{\beta}}^-$ for different choices of $\boldsymbol{\mu}$ since $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^s$. The key idea is to redefine the constraint $b_{\boldsymbol{\beta}} \geq |G(\boldsymbol{\mu})_{\boldsymbol{\beta}}|$ by a new constraint $b_{\boldsymbol{\beta}} \geq \max\{G(\boldsymbol{\mu})_{\boldsymbol{\beta}}^+, G(\boldsymbol{\mu})_{\boldsymbol{\beta}}^-\}$. Let R be defined as in Section 2.4 and let $g_{i,\boldsymbol{\alpha}(0)}^+ = \max\{g_{i,\boldsymbol{\alpha}(0)}, 0\}$, i.e., we only consider the terms with exponents $\boldsymbol{\alpha}(0)$ which are positive in the g_i and thus negative in $G(\boldsymbol{\mu})$.

We redefine p as

$$p(\boldsymbol{\mu}, \{(\mathbf{a}_\beta, b_\beta) : \beta \in \Delta(G)\}) = \sum_{i=1}^s \mu_i g_{i, \alpha(0)}^+ + \sum_{\substack{\beta \in \Delta(G) \\ \lambda_0^{(\beta)} \neq 0}} \lambda_0^{(\beta)} \cdot b_\beta^{\frac{1}{\lambda_0^{(\beta)}}} \cdot \prod_{\substack{j \in \text{nz}(\beta) \\ j \geq 1}} \left(\frac{\lambda_j^{(\beta)}}{a_{\beta, j}} \right)^{\frac{\lambda_j^{(\beta)}}{\lambda_0^{(\beta)}}}.$$

We consider the following optimization problem in the variables μ_1, \dots, μ_s and $a_{\beta, 1}, \dots, a_{\beta, r}, b_\beta$ for every $\beta \in \Delta(G)$.

$$(3.2) \quad \left\{ \begin{array}{l} \text{minimize} \quad p(\boldsymbol{\mu}, \{(\mathbf{a}_\beta, b_\beta) : \beta \in \Delta(G)\}) \text{ over the subset of } R \\ \\ (1) \quad \sum_{\beta \in \Delta(G)} a_{\beta, j} \leq G(\boldsymbol{\mu})_{\alpha(j)} \text{ for all } 1 \leq j \leq r, \\ \\ \text{defined by:} \quad (2) \quad \prod_{j \in \text{nz}(\beta)} \left(\frac{a_{\beta, j}}{\lambda_j^{(\beta)}} \right)^{\lambda_j^{(\beta)}} \geq b_\beta, \text{ for all } \beta \in \Delta(G) \text{ with } \lambda_0^{(\beta)} = 0, \\ \\ (3) \quad G(\boldsymbol{\mu})_\beta^+ \leq b_\beta \text{ for all } \beta \in \Delta(G), \text{ and} \\ \\ (4) \quad G(\boldsymbol{\mu})_\beta^- \leq b_\beta \text{ for all } \beta \in \Delta(G). \end{array} \right.$$

This problem is, by condition (1), feasible only for choices of $\boldsymbol{\mu}$ such that $G(\boldsymbol{\mu})_{\alpha(j)} > 0$ for all $\alpha(j)$ since all $a_{\beta, j}$ are strictly positive. We set the output as $-\infty$ in all other cases. Indeed, with some additional assumptions the program (3.2) is a geometric program. Moreover, it is a relaxation of the program (2.6).

Theorem 3.1. *Assume that for every $1 \leq j \leq r$ the form $G(\boldsymbol{\mu})_{\alpha(j)} = -\sum_{i=0}^s \mu_i \cdot g_{i, \alpha(j)}$ has exactly one strictly positive term, i.e. there exists exactly one strictly negative $g_{i, \alpha(j)}$. Then the optimization problem (3.2) restricted to $\boldsymbol{\mu} \in (0, \infty)^s$ is a geometric program. Assume that γ_{sonc} denotes the optimal value of (3.2) and γ denotes the optimal value of (2.6). Then we have*

$$f_{\alpha(0)} - \gamma_{\text{sonc}} \leq f_{\alpha(0)} - \gamma \leq s(f, \mathbf{g}).$$

The typical choice for $\alpha(0)$ is the origin which yields a lower bound for f to be non-negative on K with the inequality (2.5):

Corollary 3.2. *Let all assumptions be as in Theorem 3.1. If $\alpha(0)$ is the origin, then we have*

$$f_0 - \gamma_{\text{sonc}} \leq f_0 - \gamma \leq s(f, \mathbf{g}) \leq f_K^*.$$

Proof. (Theorem 3.1) If we restrict ourselves to $\boldsymbol{\mu} \in (0, \infty)^s$, then all functions involved in (3.2) depend on variables in $\mathbb{R}_{>0}$. By assumption every $G(\boldsymbol{\mu})_{\alpha(j)}$ has exactly one strictly

positive term. Thus, we can express constraint (1) as

$$\frac{\sum_{\beta \in \Delta(G)} a_{\beta,j} + G(\boldsymbol{\mu})_{\alpha(j)}^-}{G(\boldsymbol{\mu})_{\alpha(j)}^+} \leq 1,$$

with $G(\boldsymbol{\mu})_{\alpha(j)}^-$ and $G(\boldsymbol{\mu})_{\alpha(j)}^+$ defined analogously as in (3.1). Since $G(\boldsymbol{\mu})_{\alpha(j)}^+$ is a monomial the left hand side is a posynomial in $\boldsymbol{\mu}$ and \mathbf{x} . The constraints (2) – (4) are posynomial constraints in the sense of Definition 2.5 of a geometric program. The function p is also a posynomial since all terms are nonnegative by construction and all exponents are rational. Moreover, every b_{β} in (3.2) has to be greater or equal than the corresponding b_{β} in (2.6) because $\max\{a, b\} \geq |a - b|$ for all $a, b \in \mathbb{R} \setminus \{0\}$. Since furthermore $g_{i, \alpha(0)}^+ \geq g_{i, \alpha(0)}$ it follows that $\gamma_{\text{sonc}} \leq \gamma$ by the definitions of (3.2) and (2.6). The last inequality follows from Theorem 2.9. \square

One expects the programs (2.6) and (3.2) to have a similar optimal value if, for example, $g_{i, \alpha(0)} \geq 0$ for most $i = 1, \dots, s$ and if one $G(\boldsymbol{\mu})_{\beta}^+, G(\boldsymbol{\mu})_{\beta}^-$ is identically zero for most $\beta \in \Delta(G)$. Note that one of $G(\boldsymbol{\mu})_{\beta}^+, G(\boldsymbol{\mu})_{\beta}^-$ is zero if and only if $\max\{G(\boldsymbol{\mu})_{\beta}^+, G(\boldsymbol{\mu})_{\beta}^-\} = |G(\boldsymbol{\mu})_{\beta}^+ - G(\boldsymbol{\mu})_{\beta}^-| = |G(\boldsymbol{\mu})_{\beta}|$ if and only if the $g_{i, \beta}$ are all ≥ 0 or all ≤ 0 for $i = 0, \dots, s$.

We give an example to demonstrate how a given constrained polynomial optimization problem can be translated into the geometric program (3.2). In Section 4, we provide several further examples including actual computations of infima using the GP-solver CVX.

Example 3.3. Let $f = 1 + 2x^2y^4 + \frac{1}{2}x^3y^2$ and $g_1 = \frac{1}{3} - x^6y^2$. From these two polynomials we obtain a function

$$G(\mu) = \left(1 - \frac{1}{3}\mu\right) + 2x^2y^4 + \mu x^6y^2 + \frac{1}{2}x^3y^2.$$

For $G(\mu)$ to be an ST-polynomial, we have to choose $\mu \in (0, 3)$. Here, the vertices of $\text{New}(G(\mu))$ are $\boldsymbol{\alpha}(0) = (0, 0)$, $\boldsymbol{\alpha}(1) = (2, 4)$, $\boldsymbol{\alpha}(2) = (6, 2)$, and we have $\Delta(G) = \{\boldsymbol{\beta}\} = \{(3, 2)\}$. Thus, we introduce 4 variables $(a_{\beta,1}, a_{\beta,2}, b_{\beta}, \mu)$. First, we compute the barycentric coordinates of $\boldsymbol{\beta}$ and get

$$\lambda_0^{(\boldsymbol{\beta})} = \frac{3}{10}, \quad \lambda_1^{(\boldsymbol{\beta})} = \frac{3}{10}, \quad \lambda_2^{(\boldsymbol{\beta})} = \frac{2}{5}.$$

We match the coefficients of $G(\mu)$ with the vertices $\boldsymbol{\alpha}(j)$:

- $g_{1, \alpha(0)}^+ = \max\{\frac{1}{3}, 0\} = \frac{1}{3}$,
- $G(\mu)_{\alpha(1)} = 2$, $G(\mu)_{\alpha(2)} = \mu$,
- $G(\mu)_{\beta}^+ = \frac{1}{2}$, $G(\mu)_{\beta}^-$ does not exist.

Hence, Program (3.2) is of the form:

$$\inf \left\{ \frac{1}{3}\mu + \frac{3}{10} \cdot b_{\beta}^{\frac{10}{3}} \cdot \left(\frac{3}{10}\right)^1 \cdot \left(\frac{2}{5}\right)^{\frac{4}{3}} \cdot (a_{\beta,1})^{-1} \cdot (a_{\beta,2})^{-\frac{4}{3}} \right\}$$

such that:

- (1) $a_{\beta,1} \leq 2, a_{\beta,2} \leq \mu$.
- (2) The second constraint does not appear, because we do not have $\lambda_0^{(\beta)} = 0$.
- (3) $\frac{1}{2} \leq b_{\beta}$.
- (4) The fourth constraint does not appear, because we do not have a $G(\mu)_{\beta}^-$.

◻

In what follows, we extend Theorem 2.9 by reformulating the program (3.2) such that it is *always* applicable. On the one hand, the new program is only a signomial program instead of a geometric program in general. On the other hand, the reformulated program covers the missing cases of Theorem 3.1 and also yields better bounds than the corresponding geometric program (3.2) in general. Let

$$q(\boldsymbol{\mu}, \{(\mathbf{a}_{\beta}, c_{\beta}) : \beta \in \Delta(G)\}) = \sum_{i=1}^s \mu_i g_{i,\alpha(0)} + \sum_{\substack{\beta \in \Delta(G) \\ \lambda_0^{(\beta)} \neq 0}} \lambda_0^{(\beta)} \cdot c_{\beta}^{\frac{1}{\lambda_0^{(\beta)}}} \cdot \prod_{\substack{j \in \text{nz}(\beta) \\ j \geq 1}} \left(\frac{\lambda_j^{(\beta)}}{a_{\beta,j}} \right)^{\frac{\lambda_j^{(\beta)}}{\lambda_0^{(\beta)}}}.$$

We consider the following program.

$$(3.3) \quad \left\{ \begin{array}{l} \text{minimize} \quad q(\boldsymbol{\mu}, \{(\mathbf{a}_{\beta}, c_{\beta}) : \beta \in \Delta(G)\}) \text{ over the subset of } R \\ \\ \text{defined by:} \quad (1) \quad \sum_{\beta \in \Delta(G)} a_{\beta,j} \leq G(\boldsymbol{\mu})_{\alpha(j)} \text{ for all } 1 \leq j \leq r, \\ (2) \quad \prod_{j \in \text{nz}(\beta)} \left(\frac{a_{\beta,j}}{\lambda_j^{(\beta)}} \right)^{\lambda_j^{(\beta)}} \geq c_{\beta} \text{ for all } \beta \in \Delta(G) \text{ with } \lambda_0^{(\beta)} = 0, \\ (3) \quad G(\boldsymbol{\mu})_{\beta}^+ - G(\boldsymbol{\mu})_{\beta}^- \leq c_{\beta} \text{ for all } \beta \in \Delta(G), \text{ and} \\ (4) \quad G(\boldsymbol{\mu})_{\beta}^- - G(\boldsymbol{\mu})_{\beta}^+ \leq c_{\beta} \text{ for all } \beta \in \Delta(G). \end{array} \right.$$

The key difference between this program and (3.2) is that

$$c_{\beta} \geq \max\{G(\boldsymbol{\mu})_{\beta}^+ - G(\boldsymbol{\mu})_{\beta}^-, G(\boldsymbol{\mu})_{\beta}^- - G(\boldsymbol{\mu})_{\beta}^+\} = |G(\boldsymbol{\mu})_{\beta}|.$$

We obtain the following statement.

Theorem 3.4. *The optimization problem (3.3) restricted to $\boldsymbol{\mu} \in (0, \infty)^s$ is a signomial program. Assume that γ_{snp} denotes the optimal value of (3.3) and $\gamma_{\text{sonc}}, \gamma$ denote the optimal values of (3.2) and (2.6) as before. Then we have*

$$f_{\alpha(0)} - \gamma_{\text{sonc}} \leq f_{\alpha(0)} - \gamma_{\text{snp}} \leq f_{\alpha(0)} - \gamma \leq s(f, \mathbf{g}).$$

Particularly, we have $\gamma_{\text{snp}} = \gamma$ if the program (2.6) attains its optimal value for $\boldsymbol{\mu} \in (0, \infty)^s$.

Proof. The proof is analogue to the proof of Theorem 3.1. The only difference is that certain terms can have a negative sign now and hence posynomials then become signomials. The statement follows with the definition of a signomial program; see Section 2.2. ◻

Finally, we show that if we strengthen the assumptions in Theorem 3.1, then, the output $f_{\alpha(0)} - \gamma_{\text{sonc}}$ of (3.2) equals the output $f_{\alpha(0)} - \gamma$ of (2.6) and particularly the bound $s(f, \mathbf{g})$.

Theorem 3.5. *Assume that for every $1 \leq j \leq r$ the form $G(\boldsymbol{\mu})_{\alpha(j)} = -\sum_{i=0}^s \mu_i \cdot g_{i,\alpha(j)}$ has exactly one strictly positive term. Furthermore, assume that $g_{i,\alpha(0)} \geq 0$ for all $i = 1, \dots, s$, and that $\Delta(A) \cap A_i \cap A_l = \emptyset$ for all $0 \leq i < l \leq s$. Let γ be the optimal value of the program (2.6). If the optimal value $s(f, \mathbf{g}) = \sup\{G(\boldsymbol{\mu})_{\text{sonc}} : \boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^s\}$ is attained for some $\boldsymbol{\mu} \in (0, \infty)^s$, then $f_{\alpha(0)} - \gamma_{\text{sonc}} = f_{\alpha(0)} - \gamma = s(f, \mathbf{g})$, where, as before, γ_{sonc} denotes the optimal value of (3.2).*

Note that the condition $\Delta(A) \cap A_i \cap A_l = \emptyset$ is satisfied if the supports of g_i and g_l differ in all elements that are not vertices of $\text{New}(G(\boldsymbol{\mu}))$.

Proof. The assumption $\Delta(A) \cap A_i \cap A_l = \emptyset$ for all $0 \leq i < l \leq s$ implies for every $\boldsymbol{\beta} \in \Delta(G)$ that $G(\boldsymbol{\mu})_{\boldsymbol{\beta}} = -\sum_{i=0}^s \mu_i \cdot g_{i,\boldsymbol{\beta}} = -\mu_k \cdot g_{k,\boldsymbol{\beta}}$, for some $k \in [0, s]$. Therefore, we have for every $\boldsymbol{\beta} \in \Delta(G)$ that

$$\max\{G(\boldsymbol{\mu})_{\boldsymbol{\beta}}^+, G(\boldsymbol{\mu})_{\boldsymbol{\beta}}^-\} = |\mu_k \cdot g_{k,\boldsymbol{\beta}}| = |G(\boldsymbol{\mu})_{\boldsymbol{\beta}}|.$$

Furthermore, we have $g_{i,\alpha(0)} \geq 0$ for all $i = 1, \dots, s$ by assumption and thus we obtain $\sum_{i=1}^s \mu_i g_{i,\alpha(0)} = \sum_{i=1}^s \mu_i g_{i,\alpha(0)}^+$. Hence, the two programs (2.6) and (3.2) coincide.

By assumption, every $G(\boldsymbol{\mu})_{\alpha(j)}$ consists of exactly one positive term. Therefore, (3.2) is a GP by Theorem 3.1. Considering Theorem 3.1 it suffices to show the inequality $f_{\alpha(0)} - \gamma_{\text{sonc}} \geq s(f, \mathbf{g})$ for $f_{\alpha(0)} - \gamma_{\text{sonc}} = f_{\alpha(0)} - \gamma = s(f, \mathbf{g})$ to hold. Let $\boldsymbol{\mu}^* \in (0, \infty)^s$ be such that $G(\boldsymbol{\mu}^*)_{\text{sonc}} = s(f, \mathbf{g})$. By Corollary 2.7 $G(\boldsymbol{\mu}^*)_{\text{sonc}}$ is given by a feasible point $(a_{\boldsymbol{\beta},1}, \dots, a_{\boldsymbol{\beta},r})$ of the program

$$\left\{ \begin{array}{l} \text{minimize} \quad \sum_{\substack{\boldsymbol{\beta} \in \Delta(G) \\ \lambda_0^{(\boldsymbol{\beta})} \neq 0}} \lambda_0^{(\boldsymbol{\beta})} \cdot |\mu_k^* \cdot g_{k,\boldsymbol{\beta}}|^{\frac{1}{\lambda_0^{(\boldsymbol{\beta})}}} \cdot \prod_{\substack{j \in \text{enz}(\boldsymbol{\beta}) \\ j \geq 1}} \left(\frac{\lambda_j^{(\boldsymbol{\beta})}}{a_{\boldsymbol{\beta},j}} \right)^{\frac{\lambda_j^{(\boldsymbol{\beta})}}{\lambda_0^{(\boldsymbol{\beta})}}} \quad \text{over the subset } R' \text{ of } R \\ \\ \text{defined by:} \quad (1) \quad \sum_{\boldsymbol{\beta} \in \Delta(G)} a_{\boldsymbol{\beta},j} \leq G(\boldsymbol{\mu}^*)_{\alpha(j)} \quad \text{for all } 1 \leq j \leq r, \text{ and} \\ (2) \quad \prod_{j \in \text{enz}(\boldsymbol{\beta})} \left(\frac{a_{\boldsymbol{\beta},j}}{\lambda_j^{(\boldsymbol{\beta})}} \right)^{\lambda_j^{(\boldsymbol{\beta})}} \geq |\mu_k^* \cdot g_{k,\boldsymbol{\beta}}| \quad \text{for all } \boldsymbol{\beta} \in \Delta(G) \text{ with } \lambda_0^{(\boldsymbol{\beta})} = 0. \end{array} \right.$$

Then every $(a_{\boldsymbol{\beta},1}, \dots, a_{\boldsymbol{\beta},r}, b_{\boldsymbol{\beta}}, \boldsymbol{\mu}^*)$ with $b_{\boldsymbol{\beta}} \geq |\mu_k^* \cdot g_{k,\boldsymbol{\beta}}|$ for all $\boldsymbol{\beta} \in \Delta(G)$ is a feasible point of (3.2). Furthermore,

$$\begin{aligned}
 f_{\alpha(0)} - \sum_{i=1}^s \mu_i^* g_{i,\alpha(0)}^+ &= \sum_{\substack{\beta \in \Delta(G) \\ \lambda_0^{(\beta)} \neq 0}} \lambda_0^{(\beta)} \cdot b_{\beta}^{\frac{1}{\lambda_0^{(\beta)}}} \cdot \prod_{\substack{j \in \text{nz}(\beta) \\ j \geq 1}} \left(\frac{\lambda_j^{(\beta)}}{a_{\beta,j}} \right)^{\frac{\lambda_j^{(\beta)}}{\lambda_0^{(\beta)}}} \\
 &= G(\boldsymbol{\mu}^*)(0) - \sum_{\substack{\beta \in \Delta(G) \\ \lambda_0^{(\beta)} \neq 0}} \lambda_0^{(\beta)} \cdot b_{\beta}^{\frac{1}{\lambda_0^{(\beta)}}} \cdot \prod_{\substack{j \in \text{nz}(\beta) \\ j \geq 1}} \left(\frac{\lambda_j^{(\beta)}}{a_{\beta,j}} \right)^{\frac{\lambda_j^{(\beta)}}{\lambda_0^{(\beta)}}} \\
 &\geq G(\boldsymbol{\mu}^*)(0) - \sum_{\substack{\beta \in \Delta(G) \\ \lambda_0^{(\beta)} \neq 0}} \lambda_0^{(\beta)} \cdot |\mu_k^* \cdot g_{k,\beta}|^{\frac{1}{\lambda_0^{(\beta)}}} \cdot \prod_{\substack{j \in \text{nz}(\beta) \\ j \geq 1}} \left(\frac{\lambda_j^{(\beta)}}{a_{\beta,j}} \right)^{\frac{\lambda_j^{(\beta)}}{\lambda_0^{(\beta)}}}.
 \end{aligned}$$

Hence, $f_{\alpha(0)} - \gamma_{\text{sonc}} \geq G(\boldsymbol{\mu}^*)_{\text{sonc}} = s(f, \mathbf{g})$. □

4. EXAMPLES FOR CONSTRAINED OPTIMIZATION VIA GEOMETRIC PROGRAMMING AND A COMPARISON TO LASSERRE RELAXATIONS

We consider constrained polynomial optimization problems of the form

$$f_K^* = \inf_{\mathbf{x} \in K} f(\mathbf{x}),$$

where K is a basic closed semialgebraic set defined by $g_1, \dots, g_s \geq 0$. One of the main results in [IdW16b] is the observation that lower bounds for global optimization problems arising from SONCs via GP can not only be computed faster, but also be better than the bounds obtained by SOS via SDP. Here, we show that competitive bounds arising from SONC via GP can also be obtained for constrained problems. Particularly, if $2d$ is the maximal total degree of f and g_1, \dots, g_s , then the bound given by the d -th Lasserre relaxation is not necessarily as good as our optimal solution, which is in contrast to the bounds obtained by Ghasemi and Marshall; see Example 4.5 for further details. Moreover, we provide runtimes for rescaled version of our examples demonstrating that the runtime of the GP approach is not sensitive to increasing the degree of a given problem. This has been observed by several authors in the past, e.g. [BKVH07, GM12, GM13], and is in contrast to the runtime of SDPs.

Let $\Sigma\mathbb{R}[\mathbf{x}]^2$ denote the set of *n -variate sums of squares*. We consider the *d -th Lasserre relaxation* [Las10]

$$(4.1) \quad f_{\text{sos}}^{(d)} = \sup \left\{ r : f - r = \sum_{i=0}^s \sigma_i g_i, \sigma_i \in \Sigma\mathbb{R}[\mathbf{x}]^2, g_0 = 1, \deg(\sigma_i g_i) \leq 2d \right\},$$

where $d \geq \max\{\lceil \deg(f)/2 \rceil, \max_{1 \leq i \leq s} \{\lceil \deg(g_i)/2 \rceil\}\}$.

In what follows we provide several examples comparing Lasserre relaxation using the MATLAB SDP solver GLOPTIPOLY [HLL09] to our approach given in program (3.2) using the MATLAB GP solver CVX [BG08, BGY06]. In every example in this section we optimize with respect to the constant term when applying program (3.2).

Example 4.1. Let $f = 1 + x^4y^2 + x^2y^4 - 3x^2y^2$ be the Motzkin polynomial and $g_1 = x^3y^2$. Then

$$K = \{(x, y) \in \mathbb{R}^2 : x \geq 0 \text{ or } y = 0\}.$$

Since f is globally nonnegative and has two zeros $(1, 1), (1, -1)$ on K , e.g. [Rez00], we have $f_K^* = 0$. We consider the third Lasserre relaxation and obtain

$$f_{\text{sos}}^{(3)} = \sup \{r : f - r = \sigma_0 + \sigma_1 \cdot g_1, \sigma_0, \sigma_1 \in \Sigma\mathbb{R}[\mathbf{x}]^2, \deg(\sigma_0) \leq 6, \deg(\sigma_1 g_1) \leq 6\} = -\infty,$$

since the problem is infeasible. Note that K is unbounded. Hence, it is not necessarily the case that $f_{\text{sos}}^{(d)} > -\infty$ for sufficiently high relaxation order d . Here, using GLOPTIPOLY, one can find that $f_{\text{sos}}^{(7)} = 0 = f_K^*$.

Now, we consider $s(f, g_1) = \sup\{G(\mu)_{\text{sonc}} : \mu \in \mathbb{R}_{\geq 0}\} \leq f_K^*$ where $G(\mu) = f - \mu g_1$ with $\mu \geq 0$. Note that $\text{New}(G(\mu))$ is a simplex for every choice of μ . In particular, for $\mu = 0$ we have that $G(\mu)_{\text{sonc}} = f_{\text{sonc}} = 0$, since the Motzkin polynomial is a SONC polynomial; see Section 2.1 and also [IdW16a]. It follows that

$$-\infty = f_{\text{sos}}^{(3)} < s(f, g_1) = 0 = f_K^*.$$

Hence, $s(f, g_1)$ yields the exact solution compared to the Lasserre relaxation. This is in sharp contrast to the geometric programming approach proposed in [GM13] where $f_{\text{sos}}^{(d)} \geq s(f, \mathbf{g})$ holds in general. \square

Note that there are techniques which handle polynomial optimization problems over unbounded feasible sets better than the SOS/SDP method, e.g. using the gradient ideal [NDS06] or using gradient tentacles [Sch06]. Here, we restrict ourselves, however, to a comparison to SDP based methods.

Example 4.2. Let $f = 1 + x^4y^2 + xy$ and $g_1 = \frac{1}{2} + x^2y^4 - x^2y^6$. The feasible set K is a non-compact set depicted in Figure 1. Using GLOPTIPOLY, one can check that $-\infty = f_{\text{sos}}^{(4)}$ and the optimal solution is given for $d = 8$ with $f_{\text{sos}}^{(8)} \approx 0.4474$. In this case one can extract the minimizers $(-0.557, 1.2715)$ and $(0.557, -1.2715)$.

We compare the results to our approach via geometric programming instead of Lasserre relaxations. From f and g_1 we get $G(\mu) = (1 - \frac{1}{2}\mu) + x^4y^2 + \mu x^2y^6 + xy - \mu x^2y^4$. Note that $\text{New}(G(\mu))$ is a two dimensional simplex if $\mu \notin \{0, 2\}$. Then, we have $\Delta(G) = \{\beta, \tilde{\beta}\} = \{(1, 1), (2, 4)\}$. Hence, we introduce the variables $(a_{\beta,1}, a_{\beta,2}, a_{\tilde{\beta},1}, a_{\tilde{\beta},2}, b_{\beta}, b_{\tilde{\beta}}, \mu)$. Therefore, the geometric program (3.2) reads as follows:

$$\inf \left\{ \frac{1}{2}\mu + \frac{7}{10} \cdot b_{\tilde{\beta}}^{\frac{10}{7}} \cdot \left(\frac{1}{5}\right)^{\frac{2}{7}} \cdot \left(\frac{1}{10}\right)^{\frac{1}{7}} \cdot (a_{\beta,1})^{-\frac{2}{7}} \cdot (a_{\beta,2})^{-\frac{1}{7}} + \frac{1}{5} \cdot b_{\beta}^5 \cdot \left(\frac{1}{5}\right)^1 \cdot \left(\frac{3}{5}\right)^3 \cdot (a_{\tilde{\beta},1})^{-1} \cdot (a_{\tilde{\beta},2})^{-3} \right\}$$

such that the variables satisfy

$$a_{\beta,1} + a_{\tilde{\beta},1} \leq 1, \quad a_{\beta,2} + a_{\tilde{\beta},2} \leq \mu \quad \text{and} \quad 1 \leq b_{\beta}, \quad \mu \leq b_{\tilde{\beta}}.$$

We use the MATLAB solver CVX to solve the program given above. The optimal solution is given by

$$(a_{\beta,1}, a_{\beta,2}, a_{\tilde{\beta},1}, a_{\tilde{\beta},2}, b_{\beta}, b_{\tilde{\beta}}, \mu) = (0.9105, 0.0540, 0.0895, 0.0319, 1.0000, 0.0859, 0.0859).$$

This leads to

$$\gamma_{\text{sonc}} \approx 0.5526$$

and hence $f_{\alpha(0)} - \gamma_{\text{sonc}} \approx 0.4474$. Thus, we have

$$f_{\text{sos}}^{(8)} = f_{\alpha(0)} - \gamma_{\text{sonc}} = s(f, g_1).$$

The equality $f_{\alpha(0)} - \gamma_{\text{sonc}} = s(f, g_1)$ is not surprising, since the assumptions of Theorem 3.5 are satisfied. Thus, we get the optimal solution immediately via geometric programming whereas one needs 5 relaxation steps via Lasserre relaxation. In this example both geometric programming and the Lasserre approach have a runtime below 1 second.

Now, we demonstrate that our approach, in contrast to SDPs, is not sensitive to increasing the degree, since it is based on GPs. Namely, if we multiply all exponents in f and g_1 by 10, then the approaches differ significantly. By multiplying the exponents by 10 we have made a severe change to the problem since the term $x^{10}y^{10}$ is now a monomial square such that the exponent is a lattice point in the interior of the Newton polytope of the adjusted $G(\mu)$. Therefore, we have to ignore this term out when running the constrained optimization program (3.2). The adjusted program yields with CVX an output NaN in below one second. However, the reason is that it computes $\mu = 0$, which is the correct answer. Namely, after multiplying the exponents by 10, the only non monomial square terms are given by g_1 . Thus, the optimal choice is $\mu = 0$, and we can see that the minimal value is attained at $(0, 0)$ and $f_K^* = 1$ is given by the constant term of f .

In comparison, we have a runtime of approximately 1110 seconds, i.e. approximately 18.5 minutes with GLOPTIPOLY. After this time GLOPTIPOLY provides an output “Run into numerical problems.”. It claims, however, to have solved the problem and provides the correct minimum $f_K^* = 1$ at a minimizer $10^{-7} \cdot (-0.1057, 0.1711)$, which, of course, is the origin up to a numerical error.

We remark that this particular artificial increase of the degree can, of course, be handled with SDP methods using pre-processing methods, e.g. a change of variables. The purpose of the rescaling of this and later examples is, however, to show that SONC/GP methods are *inherently* not sensitive to an increase of the degree. \square

Example 4.3. Let $f = 1 + x^2z^2 + y^2z^2 + x^2y^2 - 8xyz$ and $g_1 = x^2yz + xy^2z + x^2y^2 - 2 + xyz$. Using GLOPTIPOLY, we get the following sequence of lower bounds:

$$f_{\text{sos}}^{(2)} = f_{\text{sos}}^{(3)} = f_{\text{sos}}^{(4)} = -\infty < f_{\text{sos}}^{(5)} \approx -14.999.$$

However, one cannot certify the optimality via GLOPTIPOLY in this case. Additionally, the sequence $f_{\text{sos}}^{(d)}$ is not guaranteed to converge to f_K^* , since K is unbounded. Symbolically, we were able to prove a global minimum of $f_K^* = -15$ with four global minimizers

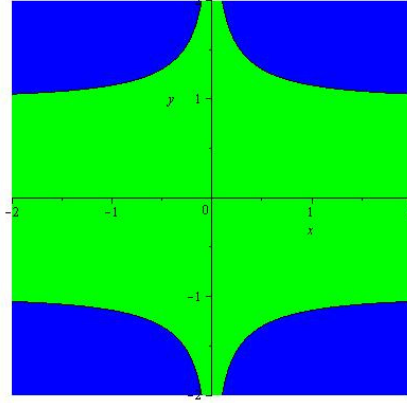


FIGURE 1. The feasible set for the constrained optimization problem in Example 4.2 is the unbounded green (light) area.

$(2, 2, 2), (-2, -2, 2), (-2, 2, -2), (2, -2, -2)$ using the quantifier elimination software SYN-RAC, see [AY03]. Now, we consider the approach via geometric programming instead of Lasserre relaxations. We have

$$G(\mu) = (1 + 2\mu) + x^2z^2 + y^2z^2 + (1 - \mu)x^2y^2 + (-8 - \mu)xyz - \mu x^2yz - \mu xy^2z.$$

Therefore, $G(\mu)$ is an ST-polynomial for $\mu \in [0, 1)$, and we have $\Delta(G) = \{\beta, \bar{\beta}, \hat{\beta}\} = \{(1, 1, 1), (2, 1, 1), (1, 2, 1)\}$. Thus, our geometric program has the following 13 variables

$$(a_{\beta,1}, a_{\beta,2}, a_{\beta,3}, a_{\bar{\beta},1}, a_{\bar{\beta},2}, a_{\bar{\beta},3}, a_{\hat{\beta},1}, a_{\hat{\beta},2}, a_{\hat{\beta},3}, b_{\beta}, b_{\bar{\beta}}, b_{\hat{\beta}}, \mu).$$

Hence, program (3.2) is of the form

$$\inf \left\{ 0 \cdot \mu + \frac{1}{4} \cdot b_{\beta}^4 \cdot \left(\frac{1}{4}\right) \cdot \left(\frac{1}{4}\right) \cdot \left(\frac{1}{4}\right) \cdot (a_{\beta,1})^{-1} \cdot (a_{\beta,2})^{-1} \cdot (a_{\beta,3})^{-1} \right\}$$

such that

- (1) $a_{\beta,1} + a_{\bar{\beta},1} + a_{\hat{\beta},1} \leq 1, a_{\beta,2} + a_{\bar{\beta},2} + a_{\hat{\beta},2} \leq 1, a_{\beta,3} + a_{\bar{\beta},3} + a_{\hat{\beta},3} + \mu \leq 1,$
- (2) $\frac{1}{2} \cdot b_{\bar{\beta}} \cdot (a_{\bar{\beta},1})^{-\frac{1}{2}} \cdot (a_{\bar{\beta},3})^{-\frac{1}{2}} \leq 1,$
- $\frac{1}{2} \cdot b_{\hat{\beta}} \cdot (a_{\hat{\beta},2})^{-\frac{1}{2}} \cdot (a_{\hat{\beta},3})^{-\frac{1}{2}} \leq 1,$
- (3) $8 \cdot b_{\beta}^{-1} \leq 1, \mu \cdot b_{\beta}^{-1} \leq 1, \mu \cdot b_{\bar{\beta}}^{-1} \leq 1, \mu \cdot b_{\hat{\beta}}^{-1} \leq 1.$

This leads to $\gamma_{\text{sonc}} = \frac{1}{256} \cdot 8^4 = 16$ and so $f_{\alpha(0)} - \gamma_{\text{sonc}} = -15$. The runtime for this example is below 1 second. Multiplying the exponents of f and g_1 by 10 yields the same results; the runtime for the geometric program remains **below 1 second**. In comparison, GLOPTIPOLY yields

$$f_{\text{sos}}^{(d)} = -\infty \text{ for } d \leq 19,$$

and provides a bound

$$f_{\text{sos}}^{(20)} \approx -14.999$$

in the 20-th relaxation after 36563 seconds, i.e. approximately **10.16 hours**. Moreover, although this bound is numerically equal to f_K^* , GLOPTIPOLY was not able to certify that the correct bound was found. \square

Example 4.4. Let $f = z^6 + x^4y^2 + x^2y^4 - 3x^2y^2z^2$ and $g_1 = x^2 + y^2 + z^2 - 1$. We obtain $G(\mu) = f - \mu g_1$. This problem is infeasible in the sense of program (3.2). Namely, condition (\clubsuit) is never satisfied since for any $\mu > 0$ we have a vertex $(2, 0, 0)$ or $(0, 2, 0)$ of $\text{New}(G(\mu))$ with a negative coefficient. Therefore, one can immediately conclude that $s(f, g_1)$ has to be obtained for $\mu = 0$. Thus, we have $s(f, g_1) = f_{\text{sonc}}$. Since f is the homogenized Motzkin polynomial we obtain immediately $f_{\text{sonc}} = f_K^* = 0$. An analogous argumentation holds for the variation $\tilde{G}(\mu) = f + \mu g_1$.

It is well-known that SDP solvers have serious issues with optimizing f for $g_1 \geq 0$ or $g_1 \leq 0$. For further information see [Nie13b, Examples 5.3 and 5.4]. \square

In the last example in this section we show that for special simplices our geometric programming approach coincides with the one in [GM13].

Example 4.5. Suppose that $\text{New}(G(\mu)) = \text{conv}\{0, 2d\mathbf{e}_1, \dots, 2d\mathbf{e}_n\}$. Hence, the Newton polytope is a $2d$ -scaled standard simplex in \mathbb{R}^n , which is the case if the pure powers x_j^{2d} for $1 \leq j \leq n$ are present in the polynomial f or in the constrained polynomials g_i . The corresponding polynomial $G(\mu)$ is an ST-polynomial; see Section 2.1. Indeed, all examples in [GM13, Example 4.8] are of that form and thus all of them are ST-polynomials.

In this case the program (2.6) coincides with the program (3) in [GM13]. One drawback of this setting is that the geometric programming bounds obtained from (2.6) are at most as good as the bound $f_{\text{sos}}^{(d)}$. Namely, if the Newton polytope of a circuit polynomial is a scaled standard simplex, then it is nonnegative if and only if it is a sum of squares; see [IdW16a] for further details. Thus, if we are in the setting of Ghasemi and Marshall and $G(\mu)$ is nonnegative, then it is a sum of squares of degree at most $2d$ which guarantees the existence of a decomposition in the sense of $f_{\text{sos}}^{(d)}$; see (4.1).

However, as we have shown in the previous examples, in the case of our program (2.6) there exist also cases where the geometric programming bounds are better than $f_{\text{sos}}^{(d)}$, since our approach is more general than in [GM13]. The reason is that the cones of sums of nonnegative circuit polynomials and sums of squares do not contain each other (but both of them are contained in the cone of nonnegative polynomials); see [IdW16a, Prop. 7.2]. \square

We point out that we make **no** assumption about the feasible set K . In particular, it is not assumed to be compact as it is in the classical setting via Lasserre relaxations in order to guarantee convergence of the relaxations. However, the crucial point in our setting so far is that $G(\mu)$ has to be an ST-polynomial. In the following Section 5 we lay the foundation for the usage of our geometric programming approach also for non-ST-polynomials.

But even if $G(\mu)$ is not an ST-polynomial, then we can enforce it to be an ST-polynomial in the case of a compact K . This can be achieved by adding a redundant constraint $g_{s+1} = x_1^{2d} + \dots + x_n^{2d} + c$ for $c \in \mathbb{R}$ to the feasible set K . In consequence $\text{New}(G(\mu))$ is a

$2d$ -scaled standard simplex and by the previous example our approach coincides with the one in [GM13]. Hence, the Lasserre relaxation cannot be outperformed in quality anymore. However, our approach can still have the better runtime. It would be interesting to add other redundant inequalities to K such that the corresponding bounds are better than the ones obtained via Lasserre relaxations. Unfortunately, no systematic way is known so far.

Furthermore, we consider in this paper numerical methods to certify nonnegativity of polynomials. Often it is desirable to obtain exact solutions, i.e. a symbolic certification. It would be interesting to find symbolic certificates from our provided numerical solution. For SOS polynomials this has been studied e.g. by Peyrl and Parrilo [PP08] and by Kalfoten, Li, Yang, and Zhi [KLYZ12].

5. OPTIMIZATION FOR NON-ST-POLYNOMIALS

The goal of this section is to provide a first approach to tackle optimization problems (both constrained and unconstrained) which cannot be expressed as a single ST-polynomial using the methods developed in [IdW16a, IdW16b] and in Section 3 in this article. A more careful investigation of these general types of nonnegativity problems will be content of a follow-up article.

We start with the case of global nonnegativity for arbitrary polynomials via SONC certificates. We recall the following statement from [IdW16a, Definition 7.1 and Proposition 7.2], which immediately follows from Section 2.1.

Fact 5.1. *Let $f \in \mathbb{R}[\mathbf{x}]$ and assume that there exist SONC polynomials g_1, \dots, g_k and positive real numbers μ_1, \dots, μ_k such that $f = \sum_{i=1}^k \mu_i g_i$. Then f is nonnegative.*

Of course, if a SONC decomposition exists, then it is not obvious how to find it in general. For ST-polynomials we know that we can find a SONC decomposition via the geometric optimization problem described in Theorem 2.6. Thus, we investigate a general polynomial $f \in \mathbb{R}[\mathbf{x}]$ supported on a set $A \subset \mathbb{N}^n$ satisfying \clubsuit . We denote

$$f = \sum_{j=0}^d f_{\alpha^{(j)}} \mathbf{x}^{\alpha^{(j)}} + \sum_{\beta \in \Delta(f)} f_{\beta} \mathbf{x}^{\beta}$$

such that $f_{\alpha^{(j)}} \mathbf{x}^{\alpha^{(j)}}$ are monomial squares. By \clubsuit , $V(A)$ are the vertices of $\text{New}(f)$ and we have $V(A) \subseteq \{\alpha(0), \dots, \alpha(d)\}$; equality, however, is not required here: $\{\alpha(0), \dots, \alpha(d)\}$ can also contain exponents of monomial squares in $\Delta(A) \setminus \Delta(f)$ which are not vertices of $\text{conv}(A)$. For simplicity we assume in what follows that the affine span of A is n -dimensional. We proceed as follows:

- (1) Choose a triangulation T_1, \dots, T_k of exponents $\alpha(0), \dots, \alpha(d) \in A$ corresponding to the monomial squares.
- (2) Compute the induced covering A_1, \dots, A_k of A given by $A_i = A \cap T_i$ for $1 \leq i \leq k$.
- (3) Assume that $\beta \in \Delta(f) \subset A$ is contained in more than one of the A_i 's. Let without loss of generality $\beta \in A_1, \dots, A_l$ with $1 < l \leq k$. Then we choose $f_{\beta,1}, \dots, f_{\beta,l} \in \mathbb{R}$

such that $\sum_{i=1}^l f_{\beta,i} = f_{\beta}$ and $\text{sign}(f_{\beta,i}) = \text{sign}(f_{\beta})$ for all $1 \leq i \leq l$. We proceed analogously for $\alpha(0), \dots, \alpha(d)$.

(4) Define new polynomials g_1, \dots, g_k such that

$$g_i = \sum_{\beta \in A_i} f_{\beta,i} \mathbf{x}^{\beta}.$$

Note that by (1) and (2) the covering A_i is a set of integer tuples such that $\text{conv}(A_i)$ is a simplex with even vertices and A_i contains no even points corresponding to monomial squares except for the vertices of $\text{conv}(A_i)$. Thus, by (2)–(4) we see that all g_i are ST-polynomials, since the signs of the $f_{\beta,i}$ are identical with the signs of the coefficients of f . Therefore, monomial squares $f_{\alpha(j)} \mathbf{x}^{\alpha(j)}$ of f get decomposed into a sum of monomial squares $\sum_{i=1}^k f_{\alpha(j),i} \mathbf{x}^{\alpha(j)}$ such that each individual monomial square $f_{\alpha(j),i} \mathbf{x}^{\alpha(j)}$ is a term of exactly one g_i . We proceed analogously for the terms $f_{\beta} \mathbf{x}^{\beta}$. Additionally, it follows by construction that $f = \sum_{i=1}^k g_i$. We apply the GP proposed in Corollary 2.7 on each of the g_i with respect to a monomial square $f_{\alpha(j),i} \mathbf{x}^{\alpha(j)}$, which is a vertex of $\text{New}(g_i) = \text{conv}(A_i)$ (not necessarily the same $\alpha(j)$ for every g_i); we denote the minimizer by m_i^* . We make the following observation about these minimizers which was similarly already pointed out in [IdW16a, Section 3]:

Lemma 5.2. *Let $f \in \mathbb{R}[\mathbf{x}]$ be a nonnegative circuit polynomial. Let $b_{\alpha} \mathbf{x}^{\alpha}$ be a monomial with $b_{\alpha} > 0$ and $\alpha \in (2\mathbb{Z})^n$. Then $b_{\alpha} \mathbf{x}^{\alpha} \cdot f$ is also a nonnegative circuit polynomial.*

Note particularly that if $\mathbf{v} \in (\mathbb{R}^*)^n$ satisfies $f(\mathbf{v}) = 0$, then $(b_{\alpha} \mathbf{x}^{\alpha} \cdot f)(\mathbf{v}) = 0$.

Proof. It is easy to see that all conditions for \clubsuit as well as the conditions (ST1) and (ST2) remain valid for $b_{\alpha} \mathbf{x}^{\alpha} \cdot f$. Thus, $b_{\alpha} \mathbf{x}^{\alpha} \cdot f$ still is a circuit polynomial and since $b_{\alpha} \mathbf{x}^{\alpha} \geq 0$ it is also nonnegative. \square

Proposition 5.3. *Let f, g_1, \dots, g_k , and m_i^* be as explained above. Assume for $i = 1, \dots, k$ that m_i^* corresponds to the monomial square $f_{\alpha(j_i)} \mathbf{x}^{\alpha(j_i)}$ with $\alpha(j_i) \in \{\alpha(1), \dots, \alpha(d)\} \cap V(A_i)$. Then $f - \sum_{i=1}^k m_i^* \mathbf{x}^{\alpha(j_i)}$ is a SONC and hence nonnegative. Thus, the m_i^* provide bounds for the coefficients $f_{\alpha(j),i}$ for f to be nonnegative. Particularly, if for $i = 1, \dots, l$ with $l \leq k$ the exponents $\alpha(j_i)$ are the origin, then $f_{\alpha(0)} - \sum_{i=1}^l m_i^*$ is a lower bound for $f^* = \sup\{\gamma \in \mathbb{R} \mid f - \gamma \geq 0\}$.*

Proof. By construction, we know that $g_i - m_i^* \mathbf{x}^{\alpha(j_i)}$ is a SONC. Thus, $f - \sum_{i=1}^k m_i^* \mathbf{x}^{\alpha(j_i)} = \sum_{i=1}^k g_i - m_i^* \mathbf{x}^{\alpha(j_i)}$ is a SONC, too. The last part follows by the definitions of the m_i^* 's and f^* . \square

Note that the decomposition of f into the g_i 's is not unique. First, the triangulation in (1) is not unique in general. And, second, the decomposition of the terms in (3) is arbitrary. Note also that there exist several monomial squares which appear in more than one g_i , since membership in A_i is given by the chosen triangulation and every simplex T_1 intersects at least one other simplex T_2 in an $n - 1$ dimensional face, which means that $A_1 \cap A_2$ contains at least n even elements. As mentioned in the introduction, the problem to identify an optimal triangulation and an optimal decomposition of coefficients will be

discussed in a follow-up article.

We provide some examples to show how this generalized approach can be used in practice.

Example 5.4. Let $f = 6 + x_1^2x_2^6 + 2x_1^4x_2^6 + 1x_1^8x_2^2 - 1.2x_1^2x_2^3 - 0.85x_1^3x_2^5 - 0.9x_1^4x_2^3 - 0.73x_1^5x_2^2 - 1.14x_1^7x_2^2$. We choose a triangulation

$$\{(\mathbf{0}, \mathbf{0}), (\mathbf{2}, \mathbf{6}), (\mathbf{4}, \mathbf{6}), (2, 3), (3, 5)\}, \{(\mathbf{0}, \mathbf{0}), (\mathbf{4}, \mathbf{6}), (\mathbf{8}, \mathbf{2}), (2, 3), (4, 3), (5, 2), (7, 2)\}.$$

Here and in what follows the vertices of each simplex are printed in red (bold). For the corresponding Newton polytope see Figure 2. We split the coefficients equally among the two triangulations and obtain two ST-polynomials

$$\begin{aligned} g_1 &= 3 + x_1^2x_2^6 + x_1^4x_2^6 - 0.6x_1^2x_2^3 - 0.85x_1^3x_2^5, \text{ and} \\ g_2 &= 3 + x_1^4x_2^6 + 1x_1^8x_2^2 - 0.6x_1^2x_2^3 - 0.9x_1^4x_2^3 - 0.73x_1^5x_2^2 - 1.14x_1^7x_2^2. \end{aligned}$$

Using CVX, we apply the GP from Corollary 2.7 and obtain optimal values $m_1^* = 0.2121$, $m_2^* = 2.5193$, and a SONC decomposition

$$\begin{aligned} 0.173 + \varepsilon x_1^2x_2^6 + 0.522x_1^4x_2^6 - 0.6x_1^2x_2^3 &+ 0.04 + x_1^2x_2^6 + 0.478x_1^4x_2^6 - 0.85x_1^3x_2^5 &+ \\ 0.427 + 0.211x_1^4x_2^6 + \varepsilon x_1^8x_2^2 - 0.6x_1^2x_2^3 &+ 0.663 + 0.436x_1^4x_2^6 + 0.085x_1^8x_2^2 - 0.9x_1^4x_2^3 &+ \\ 0.753 + 0.186x_1^4x_2^6 + 0.177x_1^8x_2^2 - 0.73x_1^5x_2^2 &+ 0.676 + 0.167x_1^4x_2^6 + 0.738x_1^8x_2^2 - 1.14x_1^7x_2^2, \end{aligned}$$

with $\varepsilon < 10^{-10}$, i.e. ε is numerically zero. Namely, $(2, 3)$ is located on the segment given by $(\mathbf{0}, \mathbf{0})$ and $(\mathbf{4}, \mathbf{6})$ and thus $(\mathbf{2}, \mathbf{6})$ and $(\mathbf{8}, \mathbf{2})$ have coefficients zero in the convex combinations of the point $(2, 3)$.

Thus, the optimal value f_{sonc} , which provides us a lower bound for f^* , is $f_{\text{sonc}} \approx 6 - 2.731 = 3.269$. In comparison, via Lasserre relaxation one obtains an only slightly better optimal value $f^* = 3.8673$.

Our GP based bound can be improved significantly via making small changes in the distribution of the coefficients. For example, if one decides not to split the coefficient of the term $x_1^2x_2^3$ among g_1 and g_2 equally, but to put the entire weight of the coefficient into g_1 , i.e.,

$$\begin{aligned} \tilde{g}_1 &= 3 + x_1^2x_2^6 + x_1^4x_2^6 - 1.2x_1^2x_2^3 - 0.85x_1^3x_2^5, \text{ and} \\ \tilde{g}_2 &= 3 + x_1^4x_2^6 + 1x_1^8x_2^2 - 0.9x_1^4x_2^3 - 0.73x_1^5x_2^2 - 1.14x_1^7x_2^2, \end{aligned}$$

then this yields to an improved bound $\tilde{f}_{\text{sonc}} \approx 3.572$. \square

The next example shows that we can use the approach of this section to take monomial squares into account, which are not vertices of the Newton polytope of the polynomial which we intend to minimize.

Example 5.5. Let $f = 1 + 3x_1^2x_2^6 + 2x_1^6x_2^2 + 6x_1^2x_2^2 - x_1x_2^2 - 2x_1^2x_2 - 3x_1^3x_2^3$. We choose a triangulation

$$\{(\mathbf{0}, \mathbf{0}), (\mathbf{2}, \mathbf{2}), (\mathbf{2}, \mathbf{6}), (1, 2)\}, \{(\mathbf{0}, \mathbf{0}), (\mathbf{2}, \mathbf{2}), (\mathbf{6}, \mathbf{2}), (2, 1)\}, \{(\mathbf{2}, \mathbf{2}), (\mathbf{2}, \mathbf{6}), (\mathbf{6}, \mathbf{2}), (3, 3)\}.$$

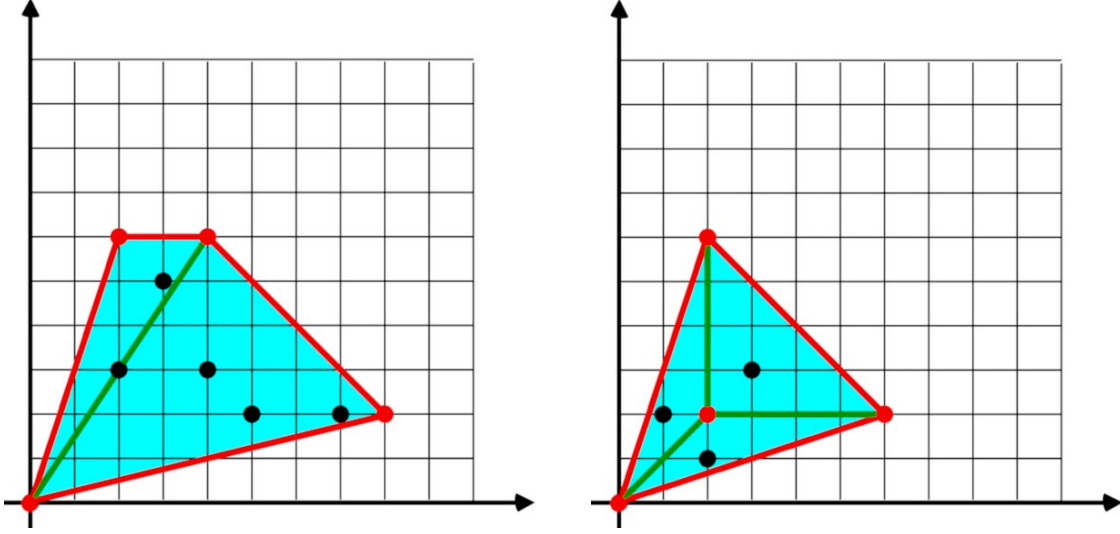


FIGURE 2. The Newton polytopes of the polynomials in the Examples 5.4 and 5.5 and their triangulations.

For the corresponding Newton polytope see Figure 2. First, we split the coefficients equally among the three triangulations such that we obtain

$$\begin{aligned} g_1 &= 0.5 + 1.5x_1^2x_2^6 + 2x_1^2x_2^2 - x_1x_2^2, \\ g_2 &= 0.5 + 1x_1^6x_2^2 + 2x_1^2x_2^2 - 2x_1^2x_2, \\ g_3 &= 1.5x_1^2x_2^6 + 1x_1^6x_2^2 + 2x_1^2x_2^2 - 3x_1^3x_2^3. \end{aligned}$$

All three g_i have a joint monomial $x_1^2x_2^2$. For all $i = 1, 2, 3$ we compute the maximal $b_i > 0$ such that $g_i - b_i x_1^2x_2^2$ is a nonnegative circuit polynomial. This yields a bound for the coefficient of $x_1^2x_2^2$ certifying that f is a SONC and hence nonnegative. We could apply the GP from Corollary 2.7, but since all g_i are circuit polynomials we can compute the corresponding circuit numbers symbolically. We obtain with Theorem 2.4:

$$\begin{aligned} \Theta_{g_1}(1,2) &= \left(\frac{1/2}{1/2}\right)^{\frac{1}{2}} \cdot \left(\frac{3/2}{1/4}\right)^{\frac{1}{4}} \cdot \left(\frac{2-b_1}{1/4}\right)^{\frac{1}{4}} = \sqrt[4]{4 \cdot 4 \cdot 3/2 \cdot (2-b_1)} = 2\sqrt[4]{3/2 \cdot (2-b_1)}, \\ \Theta_{g_2}(2,1) &= \left(\frac{1/4}{1/2}\right)^{\frac{1}{2}} \cdot \left(\frac{1/2}{1/4}\right)^{\frac{1}{4}} \cdot \left(\frac{1-1/2 \cdot b_2}{1/4}\right)^{\frac{1}{4}} = \sqrt[4]{1/4 \cdot 2 \cdot 4(1-1/2 \cdot b_2)} = \sqrt[4]{2-b_2}, \text{ and} \\ \Theta_{g_3}(3,3) &= \left(\frac{1/2}{1/4}\right)^{\frac{1}{4}} \cdot \left(\frac{1/3}{1/4}\right)^{\frac{1}{4}} \cdot \left(\frac{1/3(2-b_3)}{1/2}\right)^{\frac{1}{2}} = \sqrt[4]{2 \cdot 4/3 \cdot \sqrt{2/3 \cdot (2-b_3)}} = 2\sqrt[4]{2/27} \sqrt{2-b_3}. \end{aligned}$$

This provides solutions:

$$\begin{aligned} 2\sqrt[4]{3/2 \cdot (2-b_1)} \geq 1 &\Leftrightarrow 3/2 \cdot (2-b_1) \geq 1/16 \Leftrightarrow b_1 \leq 47/24, \\ \sqrt[4]{2-b_2} \geq 1 &\Leftrightarrow b_2 \leq 1, \\ 2\sqrt[4]{2/27} \sqrt{2-b_3} \geq 1 &\Leftrightarrow \sqrt{2/27} \cdot (2-b_3) \geq 1/4 \Leftrightarrow b_3 \leq 2 - \sqrt{27}/(2\sqrt{2}). \end{aligned}$$

Hence, we obtain the following bound for the coefficient of $x_1^2 x_2^2$:

$$6 - (47/24 + 1 + 2 - \sqrt{27}/(4\sqrt{2})) \approx 6 - 4.03977468 \approx 1.96.$$

A double check with the CVX solver for GPs yields the same value in approximately 0.753 seconds.

We want to compute a bound for f^* . We choose the same triangulation and the same split of coefficients as before, but now we optimize the constant term in g_1 and g_2 , and we optimize the coefficient of $x_1^2 x_2^6$ in g_3 . After a runtime of approximately 0.6657 seconds we obtain optimal values 0.0722, 0.3536, and 0.3164. Thus, we found a lower bound for the constant term given by

$$m_1^* + m_2^* \approx 0.0722 + 0.3536 = 0.4268.$$

The corresponding optimal SONC decomposition is given by

$$\begin{aligned} & 0.0722 + 1.5x_1^2 x_2^6 + 2x_1^2 x_2^2 - x_1^4 x_2^2 \quad + \quad 0.3536 + 1x_1^6 x_2^2 + 2x_1^2 x_2^2 - 1x_1^2 x_2^4 \quad + \\ & 0.3164x_1^2 x_2^6 + 1x_1^6 x_2^2 + 2x_1^2 x_2^2 - 3x_1^3 x_2^3 \end{aligned}$$

Thus, we obtain a bound for f^* given by

$$f_{\text{sonc}} = 1 - 0.4268 = 0.5732$$

We make a comparison and optimize f with Lasserre relaxation. This yields an optimal value

$$f_{\text{sos}} = f^* \approx 0.8383.$$

Therefore, we want to improve our bound. We keep the triangulation, but we use another distribution of the coefficients among the polynomials g_1, g_2 and g_3 and define instead

$$\begin{aligned} \tilde{g}_1 &= 0.25 + 2x_1^2 x_2^6 + 1.217x_1^2 x_2^2 - 2x_1^4 x_2^2, \\ \tilde{g}_2 &= 0.75 + 1x_1^6 x_2^2 + 3.652x_1^2 x_2^2 - 1x_1^2 x_2^4, \\ \tilde{g}_3 &= 1x_1^2 x_2^6 + 1x_1^6 x_2^2 + 1.13x_1^2 x_2^2 - 3x_1^3 x_2^3. \end{aligned}$$

Again, we optimize \tilde{g}_1 and \tilde{g}_2 with respect to the constant term and \tilde{g}_3 with respect to $x_1^2 x_2^6$. We obtain optimal values 0.0801, 0.2616, and 0.9912. Thus, we are able to improve our bound for f^* to

$$\tilde{f}_{\text{sonc}} \approx 1 - (0.0801 + 0.2616) = 0.6583.$$

The corresponding optimal SONC decomposition is given by

$$\begin{aligned} & 0.0801 + 2x_1^2 x_2^6 + 1.205x_1^2 x_2^2 - 2x_1^4 x_2^2 \quad + \quad 0.2616 + 1x_1^6 x_2^2 + 3.615x_1^2 x_2^2 - 1x_1^2 x_2^4 \quad + \\ & 0.991x_1^2 x_2^6 + 1x_1^6 x_2^2 + 2x_1^2 x_2^2 - 3x_1^3 x_2^3. \end{aligned}$$

◻

We discuss a third example which shows that, in the case of global optimization, for the GP/SONC approach it is not necessary to optimize the constant term to obtain a bound for nonnegativity on the coefficients, but that in some cases it can be informative to focus on other vertices of the Newton polytope or on other monomial squares instead.

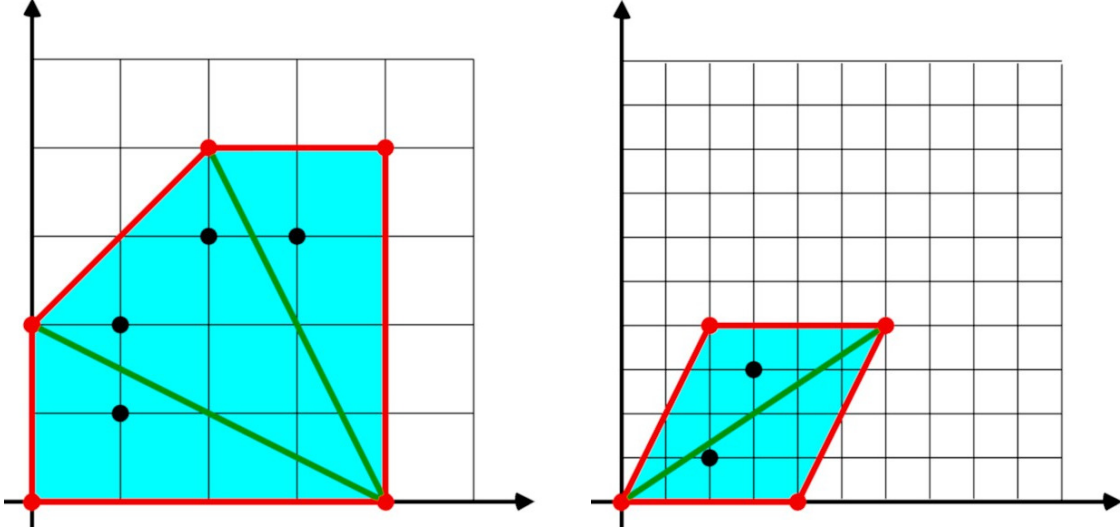


FIGURE 3. The Newton polytopes of the polynomials in the Examples 5.6 and 5.7 and their triangulations.

Example 5.6. Let $f = 1 + x_1^4 + x_2^2 + x_1^2x_2^4 + x_1^4x_2^4 - x_1x_2 - x_1x_2^2 - x_1^2x_2^3 - x_1^3x_2^3$. We choose a triangulation

$$\{(0, 0), (0, 2), (4, 0), (1, 1)\}, \{(0, 2), (2, 4), (4, 0), (1, 2), (2, 3)\}, \{(2, 4), (4, 0), (4, 4), (3, 3)\}.$$

Again, we choose a decomposition of coefficients such that their values split equally. We obtain the following ST-polynomials

$$\begin{aligned} g_1 &= 1 + 1/3 \cdot x_1^4 + 1/2 \cdot x_2^2 - x_1x_2, \\ g_2 &= 1/3 \cdot x_1^4 + 1/2 \cdot x_1^2x_2^4 + 1/2 \cdot x_2^2 - x_1x_2^2 - x_1^2x_2^3, \\ g_3 &= 1/3 \cdot x_1^4 + 1/2 \cdot x_1^2x_2^4 + x_1^4x_2^4 - x_1^3x_2^3. \end{aligned}$$

g_1 and g_3 are circuit polynomials while g_2 contains two negative terms. For the corresponding Newton polytope see Figure 3. Note that only the exponent $(4, 0)$ is contained in the support of all three ST-polynomials. Since $(4, 0)$ is a monomial square which is a vertex of the convex hull of the three support sets, we optimize the corresponding coefficient in g_1, g_2 and g_3 . Applying the GP from Corollary 2.7 yields optimal values

$$m_1^* = 0.0625, \quad m_2^* = 4.2867, \quad \text{and} \quad m_3^* = 0.0625.$$

Since $m_2^* = 4.2867 > 1/3$ we found no certificate of nonnegativity for f . However, we find a SONC decomposition for f if the coefficient $b_{(4,0)}$ of x_1^4 is at least $m_1^* + m_2^* + m_3^* = 4.412$. For this minimal choice of $b_{(4,0)}$ a SONC decomposition is given by

$$\begin{aligned} 0.063x_1^4 + 1 + 0.5x_2^2 - 1x_1^1x_2^1 &+ 2.143x_1^4 + 0.4x_2^2 + 0.1x_1^2x_2^4 - 1x_1^1x_2^2 &+ \\ 2.143x_1^4 + 0.1x_2^2 + 0.4x_1^2x_2^4 - 1x_1^2x_2^3 &+ 0.063x_1^4 + 0.5x_1^2x_2^4 + 1x_1^4x_2^4 - 1x_1^3x_2^3 \end{aligned}$$

◻

Finally, we apply the new method to a constrained optimization problem using the methods developed in Section 3.

Example 5.7. Let $f = 1 + x^4 + x^2y^4$ and $g = \frac{1}{2} + x^2y - x^6y^4 - x^3y^3$. Hence, we obtain $G(\mu) = (1 - \frac{1}{2}\mu) + x^4 + x^2y^4 + \mu x^6y^4 - \mu x^2y + \mu x^3y^3$. Choosing the triangulation

$$\{(\mathbf{0}, \mathbf{0}), (\mathbf{4}, \mathbf{0}), (\mathbf{6}, \mathbf{4}), (2, 1)\}, \{(\mathbf{0}, \mathbf{0}), (\mathbf{6}, \mathbf{4}), (\mathbf{2}, \mathbf{4}), (3, 3)\},$$

we split the coefficients again, such that their values are equal. For the corresponding Newton polytope see Figure 3. We obtain the ST-polynomials

$$\begin{aligned} G_1(\mu) &= \left(\frac{1}{2} - \frac{1}{4}\mu\right) + x^4 + \frac{1}{2}\mu x^6y^4 - \mu x^2y, \\ G_2(\mu) &= \left(\frac{1}{2} - \frac{1}{4}\mu\right) + x^2y^4 + \frac{1}{2}\mu x^6y^4 + \mu x^3y^3. \end{aligned}$$

Therefore, we see that the possible μ values to obtain ST-polynomials are $\mu \in [0, 2)$. We optimize both polynomials with respect to the constant term and obtain $m_1^* = m_2^* = 0$. The CVX solver yields NaN as an optimal value, since 0 is not positive. However, it solves the problem and computes values 0 or $\varepsilon < 10^{-200}$ for all variables, such that $m_1^* = m_2^* = 0$ follows. Hence, $f_{\alpha(0)} - m^* = 1 - 0 = 1$ and because all of the assumptions in Theorem 3.5 are satisfied we know $s(f, g) = 1$.

Checking this optimization problem with Lasserre relaxation, we get $f_{\text{sos}} = f_K^* = 1$, which approves the optimal value. Both, for the SDP and the GP we have runtimes below 1 second.

Now, we tackle the same problem, but we multiply every exponent by 10, and we compare the runtimes again. For the GP we obtain the same result and the runtime remains **below 1 second**. For the SDP we obtain with GLOPTIPOLY $f_{\text{sos}} = f_K^* = 1$ in approximately **5034.5 seconds**, i.e. approximately **1.4 hours**.

In a third approach we tackle the same problem, but we multiply the originally given exponents by 20. In this case GLOPTIPOLY is not able to handle the given matrices anymore. In comparison, we still have a runtime **below 1 second** for our GP providing the same bound as before.

Now, we also re-compute the example with the software SOSTOOLS [PAV⁺13], which, in contrast to GLOPTIPOLY, can exploit sparsity patterns. For the original problem size, the runtime is 0.42 seconds, but already in the case that all exponents are multiplied by 10 we have a runtime of 302.7100 seconds, i.e. about **5 minutes**.

Finally, we make a last comparison using the software SPARSEPOP [WKK⁺09], which especially serves to compute sparse problems based on SOS. Here we start with a runtime of 0.4495 seconds for the original problem. For the case of all exponents of f multiplied by 10 we get a runtime of 81.5438 seconds, i.e. **1.359 minutes**. \square

REFERENCES

- [AY03] H. Anai and H. Yanami, *SyNRAC: a Maple-package for solving real algebraic constraints*, Computational science—ICCS 2003. Part I, Lecture Notes in Comput. Sci., vol. 2657, Springer, Berlin, 2003, pp. 828–837.

- [BG08] S. Boyd and M. Grant, *Graph implementations for nonsmooth convex programs*, Recent Advances in Learning and Control (V. Blondel, S. Boyd, and H. Kimura, eds.), Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008, http://stanford.edu/~boyd/graph_dcp.html, pp. 95–110.
- [BGY06] S. Boyd, M. Grant, and Y. Ye, *Disciplined convex programming*, Global optimization, Non-convex Optim. Appl., vol. 84, Springer, New York, 2006, pp. 155–210.
- [BKVH07] S. Boyd, S.J. Kim, L. Vandenberghe, and A. Hassibi, *A tutorial on geometric programming*, Optim. Eng. **8** (2007), no. 1, 67–127.
- [Ble06] G. Blekherman, *There are significantly more nonnegative polynomials than sums of squares*, Israel J. Math. **153** (2006), 355–380.
- [Ble12] ———, *Nonnegative polynomials and sums of squares*, J. Amer. Math. Soc. **25** (2012), no. 3, 617–635.
- [BPT13] G. Blekherman, P.A. Parrilo, and R.R. Thomas, *Semidefinite optimization and convex algebraic geometry*, MOS-SIAM Series on Optimization, vol. 13, SIAM and the Mathematical Optimization Society, Philadelphia, 2013.
- [BV04] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, Cambridge, 2004.
- [DG14] P.J.C. Dickinson and L. Gijben, *On the computational complexity of membership problems for the completely positive cone and its dual*, Comput. Optim. Appl. **57** (2014), no. 2, 403–415.
- [dKL10] E. de Klerk and M. Laurent, *Error bounds for some semidefinite programming approaches to polynomial minimization on the hypercube*, SIAM J. Optim. **20** (2010), no. 6, 3104–3120.
- [DPZ67] R.J. Duffin, E.L. Peterson, and C. Zener, *Geometric programming: Theory and application*, John Wiley & Sons, Inc., New York-London-Sydney, 1967.
- [dW15] T. de Wolff, *Amoebas, nonnegative polynomials and sums of squares supported on circuits*, Oberwolfach Rep. (2015), no. 23, 1308–1311.
- [FK11] C. Fidalgo and A. Kovacec, *Positive semidefinite diagonal minus tail forms are sums of squares*, Math. Z. **269** (2011), no. 3-4, 629–645.
- [GLM14] M. Ghasemi, J.B. Lasserre, and M. Marshall, *Lower bounds on the global minimum of a polynomial*, Comput. Optim. Appl. **57** (2014), no. 2, 387–402.
- [GM12] M. Ghasemi and M. Marshall, *Lower bounds for polynomials using geometric programming*, SIAM J. Optim. **22** (2012), no. 2, 460–473.
- [GM13] ———, *Lower bounds for a polynomial on a basic closed semialgebraic set using geometric programming*, 2013, Preprint, arxiv:1311.3726.
- [HLL09] D. Henrion, J.B. Lasserre, and J. Löfberg, *GloptiPoly 3: moments, optimization and semidefinite programming*, Optim. Methods Softw. **24** (2009), no. 4-5, 761–779.
- [IdW16a] S. Iliman and T. de Wolff, *Amoebas, nonnegative polynomials and sums of squares supported on circuits*, Res. Math. Sci. **3** (2016), 3:9.
- [IdW16b] ———, *Lower bounds for polynomials with simplex newton polytopes based on geometric programming*, SIAM J. Optim. **26** (2016), no. 2, 1128–1146.
- [KLYZ12] E.L. Kaltofen, B. Li, Z. Yang, and L. Zhi, *Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients*, J. Symbolic Comput. **47** (2012), no. 1, 1–15.
- [Las01] J.B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim. **11** (2000/01), no. 3, 796–817.
- [Las10] ———, *Moments, positive polynomials and their applications*, Imperial College Press Optimization Series, vol. 1, Imperial College Press, London, 2010.
- [Lau09] M. Laurent, *Sums of squares, moment matrices and optimization over polynomials*, Emerging applications of algebraic geometry, IMA Vol. Math. Appl., vol. 149, Springer, New York, 2009, pp. 157–270.
- [NDS06] J. Nie, J. Demmel, and B. Sturmfels, *Minimizing polynomials via sum of squares over the gradient ideal*, Math. Program. **106** (2006), no. 3, Ser. A, 587–606.

- [Nie13a] J. Nie, *Certifying convergence of Lasserre’s hierarchy via flat truncation*, Math. Program. **142** (2013), no. 1-2, Ser. A, 485–510.
- [Nie13b] ———, *An exact Jacobian SDP relaxation for polynomial optimization*, Math. Program. **137** (2013), no. 1-2, Ser. A, 225–255.
- [Nie14] ———, *Optimality conditions and finite convergence of Lasserre’s hierarchy*, Math. Program. **146** (2014), no. 1-2, Ser. A, 97–121.
- [NN94] Y. Nesterov and A. Nemirovskii, *Interior point polynomial algorithms in convex programming*, Studies in Applied Mathematics, Society for Industrial and Applied Mathematics, 1994.
- [Oxl11] J. Oxley, *Matroid theory*, second ed., Oxford Graduate Texts in Mathematics, vol. 21, Oxford University Press, Oxford, 2011.
- [PAV⁺13] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, and P. A. Parrilo, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, <http://arxiv.org/abs/1310.4716>, 2013, Available from <http://www.eng.ox.ac.uk/control/sostools>, <http://www.cds.caltech.edu/sostools> and <http://www.mit.edu/~parrilo/sostools>.
- [PP08] H. Peyrl and P.A. Parrilo, *Computing sum of squares decompositions with rational coefficients*, Theoret. Comput. Sci. **409** (2008), no. 2, 269–281.
- [PS03] P.A. Parrilo and B. Sturmfels, *Minimizing polynomial functions*, Algorithmic and quantitative real algebraic geometry (Piscataway, NJ, 2001), DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. 60, Amer. Math. Soc., Providence, RI, 2003, pp. 83–99.
- [Rez78] B. Reznick, *Extremal PSD forms with few terms*, Duke Math. J. **45** (1978), no. 2, 363–374.
- [Rez00] ———, *Some concrete aspects of Hilbert’s 17th Problem*, Real algebraic geometry and ordered structures (Baton Rouge, LA, 1996), Contemp. Math., vol. 253, Amer. Math. Soc., Providence, RI, 2000, pp. 251–272.
- [Sch06] M. Schweighofer, *Global optimization of polynomials using gradient tentacles and sums of squares*, SIAM J. Optim. **17** (2006), no. 3, 920–942.
- [WKK⁺09] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto, *Algorithm 883: sparsePOP— a sparse semidefinite programming relaxation of polynomial optimization problems*, ACM Trans. Math. Software **35** (2009), no. 2, Art. 15, 13.

MAREIKE DRESSLER, GOETHE-UNIVERSITÄT, FB 12 – INSTITUT FÜR MATHEMATIK, POSTFACH 11 19 32, 60054 FRANKFURT AM MAIN, GERMANY

E-mail address: dressler@math.uni-frankfurt.de

SADIK ILIMAN FRANKFURT AM MAIN, GERMANY

E-mail address: sadik.iliman@gmx.net

TIMO DE WOLFF, TECHNISCHE UNIVERSITÄT BERLIN, INSTITUT FÜR MATHEMATIK, STRASSE DES 17. JUNI 136, 10623 BERLIN, GERMANY

E-mail address: dewolff@math.tu-berlin.de