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 recent 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 an efficient new method to solve constrained polynomial optimization problems with many variables or with high degree polynomials. The resulting method is significantly faster and for certain classes of polynomials also better than semidefinite programming as we demonstrate 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 (up to an ε-error); e.g. [BPT13, p. 41] and references therein. However, the size of such programs grows rapidly with the number of variables or the degree of the polynomials.

In the near past, 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 in [GM12,

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.

GM13], the corresponding geometric programs can be solved much faster than in the classical semidefinite approach in practice. Initially, the drawback of this GP based approach was that the lower bounds were by construction worse than bounds obtained via semidefinite programming and that they could only be applied in very special cases [GM12, GM13].

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)* [IdW14a]. SONC certificates are independent of sums of squares. In [IdW14b] the second and third author showed as a fundamental result 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 as semidefinite programming relaxations are linked to sums of squares. Particularly, it was shown in [IdW14b] that there exist rich classes of polynomials for which the GP/SONC based approach is in practice not only *faster* than semidefinite programs, but it also yields *better* bounds than the SDP/SOS approach. The reason is that the cone of sums of squares and the cone of nonnegative circuit polynomials are not contained in each other [IdW14a, Proposition 7.2] while all certificates used by Ghasemi and Marshall are always also sums of squares.

The main contribution of this article is an extension of the results in [IdW14b] to constrained polynomial optimization problems. As in [IdW14b] we mainly focus on the class of *ST-polynomials*, that are polynomials with a simplex Newton polytope satisfying some further conditions; see Section 2.1. The key idea is to follow an approach which was already outlined briefly in [IdW14b, Section 5] and to combine it with ideas by Ghasemi and Marshall in [GM13]. More precisely, we trace back the constrained polynomial optimization problems to unconstrained ones by defining a new polynomial, which incorporates both the objective function and the constrained polynomials. The starting point hereby is a general optimization problem from [IdW14b, Section 5], see (2.5), which provides a lower bound for the constrained problem but which is not a geometric program. A careful relaxation of this problem transforms it into a geometric optimization problem; see program (3.2) and Theorem 3.1. We show that for certain instances of polynomials the program (3.2) provides better bounds than via Lasserre relaxations and the ones in [GM13], see Section 4.

The second goal of this article is to apply polynomial optimization methods based on SONC and geometric programming beyond the class of ST-polynomials efficiently. We develop an initial approach based on triangulations of support sets of the involved polynomials. It allows to obtain 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. Particularly, we demonstrate that our GP based method is *significantly* faster than SDP for high degree examples as it had already been observed in [GM12, IdW14b].

The article is organized as follows. Since our results are build on the foundations provided in the articles [IdW14a, IdW14b], we present the full notational framework and key statements from these articles in Section 2. This also allows the reader to avoid reading the precursor articles if preferred. Note that we adjusted and simplified some notations from [IdW14a, IdW14b]. In Section 3 we provide a relaxation of the program (2.5) from [IdW14b] for constrained polynomial optimization problems, which transforms (2.5) into a geometric optimization problem. Additionally, sufficient conditions on the support of the constrained polynomials are presented, which guarantee that the obtained bounds provided by the new geometric programming relaxations (3.2) are as good as in the original non-geometric program (2.5). In Section 4, we present several examples demonstrating our results in practice. Particularly, we show that our approach outperforms the geometric programming relaxations in [GM13] and also the classical Lasserre relaxations for certain classes of polynomial optimization problems. Moreover, we provide a framework in which one can always expect to outperform the relaxations mentioned before. Finally, in Section 5 we introduce an approach based on triangulations of support sets which allows to obtain lower bounds for polynomial optimization problems for arbitrary polynomials both in the constrained and in the unconstrained case. Again, we provide several examples, which demonstrate that these methods are applicable in practice outperforming SDP based methods in several cases.

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. SONCs were recently introduced by the second and the third author in [IdW14a]; see also [dW15] for an overview. Geometric optimization problems are well-known special cases of convex optimization problems, which were first introduced in [DPZ67]. The second and the third author showed the relation between SONCs and GPs for global nonnegativity problems in a recent article [IdW14b] generalizing similar geometric programs developed by Ghasemi and Marshall [GM12].

2.1. **The Cone of Sums of Nonnegative Circuit Polynomials.**

We denote by $\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \ldots, x_n]$ the vector space of real n-variate polynomials. Let δ_{ij} be the ij-Kronecker symbol, let $e_i = (\delta_{i1}, \ldots, \delta_{in})$ be the i-th standard vector, and let $A \subset \mathbb{N}^n$ be a finite set. We denote by $\mathrm{conv}(A)$ the convex hull of A and we denote by $V(A)$ the vertices of the convex hull of A. We consider polynomials $f \in \mathbb{R}[\mathbf{x}]$ supported on A. That is, f is of the form $f = \sum_{\alpha \in A} f_\alpha x^\alpha$ 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 $\mathrm{New}(f) = \mathrm{conv}\{\alpha \in \mathbb{N}^n : f_\alpha \neq 0\}$.

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 $\mathrm{New}(f) = \mathrm{conv}(A)$. 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.

We remark that the statement remains true for real Laurent polynomials $g \in \mathbb{R}[\mathbf{x}^{\pm 1}] = \mathbb{R}[x_1^{\pm 1}, \ldots, x_n^{\pm 1}]$. Namely, we can then consider g as a polynomial f divided by a monomial square $\mathbf{x}^\alpha$ 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 $\mathrm{New}(f) = \mathrm{conv}(A)$. For simplicity, we denote this assumption by the symbol (♣) from now on.

In the following we consider the class of *ST-polynomials*. For further details about the following objects see [dW15, IdW14a, IdW14b].

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

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

with $r \leqslant n$, exponents $\alpha(j)$ and β, coefficients $f_{\alpha(j)}, f_\beta$, and a set $\Delta(A)$ for which the following hold:

(ST1): The points $\alpha(0), \alpha(1), \ldots, \alpha(r)$ are affinely independent and equal $V(A)$.

(ST2): We define $\Delta(A) = (A \backslash V(A))$. Thus, every $\beta \in \Delta(A)$ can be written uniquely as

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

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.

We denote by $\Delta(f)$ the elements of $\Delta(A)$ which appear as exponents of non-zero terms in the tail part of f and are moreover no monomial squares. I.e., we have

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

If an ST-polynomial f has a tail part consisting of at most one term, then we call f a *circuit polynomial*. ◯

Note that hypothesis (ST1) implies that $V(A) = \{\alpha(0), \ldots, \alpha(r)\}$ is the vertex set of an r-dimensional simplex. By the assumption (♣) it consists of even lattice points, and it coincides with $\mathrm{New}(f) = \mathrm{conv}(A)$. The $\lambda_j^{(\beta)}$ denote the barycentric coordinates of β relative to the vertices $\alpha(j)$ with $j = 0, \ldots, r$.

Given a polynomial f, well established algorithms from convex geometry allow to determine whether f is an ST-polynomial and if so to rewrite it in this form.

The class of ST-polynomials was first defined by the second and third author in [IdW14b] and generalizes a class considered by Fidalgo and Kovacec in [FK11] and by Ghasemi and

Marshall in [GM12, GM13]. In this article we both simplified and generalized the definition of an ST-polynomial slightly compared to [IdW14b].

Nonnegativity of ST-polynomials is closely related to an invariant called the *circuit number*, which was first defined as follows by the second and third author in [IdW14a].

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

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

with $\mathrm{nz}(\beta) = \{j \in \{0, \ldots, r\} : \lambda_j^{(\beta)} \neq 0\}$. ○

The terms "circuit polynomial" and "circuit number" are chosen since β and the $\alpha(j)$ with $j \in \mathrm{nz}(\beta)$ form a *circuit*; this is a minimally affine dependent set, see e.g. [GKZ94].

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 ([IdW14a], 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 are equivalent:*

(1) *f is nonnegative.*
(2) *$|c| \leqslant \Theta_f(\beta)$ and $\beta \notin (2\mathbb{N})^n$ or $c \geqslant -\Theta_f(\beta)$ and $\beta \in (2\mathbb{N})^n$.*

Note that (2) can be equivalently stated as: $|c| \leqslant \Theta_f(\beta)$ or f is a sum of monomial squares. We remark that also the definitions of the circuit polynomials and circuit numbers $\Theta_f(\beta)$ differ slightly from the ones used in [IdW14a] and [IdW14b]. Here $\beta \in \Delta(f)$ is not necessarily an interior point of $\mathrm{New}(f)$ as it is in [IdW14a]. In [IdW14b] the support of ST-polynomials was not fixed but changed when coefficients were equal to zero. This was inconvenient, since it required to take care of special cases. Moreover, in both [IdW14a, IdW14b] it was required that circuit polynomials and ST-polynomials have a constant term. This assumption was convenient to work with in these articles, but it is not necessary. Particularly, for some results in Section 5 we need ST-polynomials to be defined in the more general way as in Definition 2.2.

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. For further details about SONCs see also [dW15, IdW14a, IdW14b].

2.2. **Geometric Programming.** Geometric programming was introduced in [DPZ67]. It is a special type of 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. For an introduction to geometric programming, signomial programming, and an overview about applications see [BKVH07, BV04].

Definition 2.5. A function $p : \mathbb{R}^n_{>0} \to \mathbb{R}$ of the form $p(\mathbf{z}) = p(z_1, \ldots, z_n) = cz_1^{\alpha_1} \cdots z_n^{\alpha_n}$ 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* has the following form.

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

where $p_0, \ldots, p_m$ are posynomials and $q_1, \ldots, q_l$ are monomial functions.

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 see [BKVH07, BV04].

2.3. **SONC Certificates via Geometric Programming in the Unconstrained Case.** In this section we recall the main results from [IdW14b] 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. As a first key result the following statement is shown, which we adjusted slightly to the new notation in this article.

Theorem 2.6. *(*[IdW14b, Theorems 3.4 and 3.5]*) Assume that f is an ST-polynomial as in* (2.1) *and let $k \in \mathbb{R}$. Suppose that for every $(\beta, j) \in \Delta(f) \times \{1, \ldots, r\}$ there exists an $a_{\beta,j} \geqslant 0$, such that:*

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

Then $f - k\mathbf{x}^{\alpha(0)}$ is a sum of nonnegative circuit polynomials $g_1, \ldots, g_s$ such that s is the cardinality of the non-zero and tail terms of f that are non-monomial squares and for every g_i the Newton polytope $\mathrm{New}(g_i)$ *is a face of* $\mathrm{New}(f)$.

Let f_{gp} be the supremum of all $k \in \mathbb{R}$ such that for every $\beta \in \Delta(f)$ there exist nonnegative reals $a_{\beta,1}, \ldots, a_{\beta,r}$ such that the conditions (1) to (4) are satisfied. Then f_{gp} coincides with the supremum of all $k \in \mathbb{R}$ such that there exist nonnegative circuit polynomials

$g_1, g_2, \ldots, g_s$ whose Newton polytopes are faces of $\mathrm{New}(f)$ *and which satisfy* $f - k\mathbf{x}^{\alpha(0)} = \sum_{i=1}^{s} g_i$.

We remark that for the special case of scaled standard simplices the theorem 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; see [IdW14a, IdW14b] for further information.

Theorem 2.6 states

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

The notation f_{gp} indicates that this bound is given by a geometric program. In [IdW14b, Corollary 4.2] the following is shown.

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*

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

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

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

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

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

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

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

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

A key observation is that geometric programming is not only faster than semidefinite programming but the bounds obtained by these approach are also at least as good as the ones obtained by SDP for many cases of ST-polynomials as the following result shows; see [IdW14b, Corollary 3.6].

Corollary 2.8. *Let f be an ST-polynomial without monomial squares such that $\Delta(f)$ is contained in the interior of* $\mathrm{New}(f)$. *Suppose there exists a $\mathbf{v} \in (\mathbb{R}^*)^n$ such that $f_\alpha \mathbf{v}^\alpha < 0$ for all $\alpha \in \Delta(f)$. Then*

$$f_{\mathrm{gp}} \;=\; f^* \;\geqslant\; f_{\mathrm{sos}}.$$

2.4. **SONC Certificates for the Constrained Case.** In this subsection we recall known facts from [IdW14b, Section 5] about SONC certificates applied to constrained polynomial optimization problems.

Let $f, g_1, \ldots, g_s$ be elements of the polynomial ring $\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \ldots, x_n]$ and let

$$K \;=\; \{\mathbf{x} \in \mathbb{R}^n \;:\; g_i(\mathbf{x}) \geqslant 0,\; 1 \leqslant i \leqslant s\}$$

be a basic closed semialgebraic set. We consider the constrained polynomial optimization problem

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

If $s = 0$, then we have no g_i and therefore $K = \mathbb{R}^n$, which leads to a global optimization problem. This case was examined in detail in [IdW14b], and it was shown that finding lower bounds for the case of ST-polynomials is a geometric program; see also 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

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

for $\mu = (\mu_1, \ldots, \mu_s) \in \mathbb{R}^s_{\geqslant 0}$, $g_0 = -f$ and $\mu_0 = 1$. For every fixed $\mu^* \in \mathbb{R}^s_{\geqslant 0}$ the function $G(\mathbf{x}) = G(\mu^*)(\mathbf{x})$ is a polynomial in $\mathbb{R}[\mathbf{x}]$. Following an argument in [GM13] we can assume that all $-g_i$ do not contain monomial squares; see also [IdW14b, Section 5]. We remark that while we consider a fixed support the Newton polytope of $G(\mu)$ is not invariant in general if certain μ_i equal 0 or if term cancellation occurs.

In order to apply the results about geometric programming from [IdW14b] we have to assume that for a given $\mu \in \mathbb{R}^s_{\geqslant 0}$ the polynomial $G(\mu)$ is an ST-polynomial in $\mathbb{R}[\mathbf{x}]$. Results in [IdW14b, Section 3 and 4] imply that $G(\mu)_{\mathrm{gp}}$ is a lower bound for $G(\mu)$ on $\mathbb{R}^n$, see Section 2.4. If $G(\mu)$ is not an ST-polynomial for some $\mu \in \mathbb{R}^s_{\geqslant 0}$, then we set $G(\mu)_{\mathrm{gp}} = -\infty$, since the corresponding geometric program is infeasible. In consequence, if μ is fixed, then $G(\mu)_{\mathrm{gp}}$ is a lower bound for f on the semialgebraic set K. Thus, we have

$$s(f, \mathbf{g}) \;=\; \sup\{G(\mu)_{\mathrm{gp}} : \mu \in \mathbb{R}^s_{\geqslant 0}\} \;\leqslant\; f_K^*$$

where $\mathbf{g} = (g_1, \ldots, g_s)$ and $G(\mu)_{\mathrm{gp}}$ denotes the optimal value of the geometric program introduced in [IdW14b], see Section 2.4. Hence, for every fixed μ the bound $G(\mu)_{\mathrm{gp}}$ is computable by a geometric program. Unfortunately, this does not imply that the supremum is computable by a geometric program 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 [IdW14b], which is a geometric program under special conditions. We recall these results in the following.

We introduce some notation first. Let $G(\mu)(\mathbf{x})$ be defined as in (2.4). We consider the set of all $\mu \in [0, \infty)^s$ such that $G(\mu)$ is an ST-polynomial. That are all μ such that $G(\mu)_{\mathrm{gp}} \neq -\infty$.

For $0 \leqslant i \leqslant s$ let $A_i \subset \mathbb{N}^n$ be the support of the polynomial g_i and let $A = \bigcup_{i=0}^s A_i$ be the union of all supports of polynomials g_i. Assume that we have $\mathrm{New}(G(\mu)) = \mathrm{conv}(A)$ and $V(A) = \{\alpha(0), \ldots, \alpha(r)\}$ for all μ with $G(\mu)_{\mathrm{gp}} \neq -\infty$. Moreover, since we assume $G(\mu)$ to be an ST-polynomial, we have that $\mathrm{conv}(A)$ is a simplex with even vertex set $\{\alpha(0), \ldots, \alpha(r)\} \subset (2\mathbb{N})^n$.

We define $\Delta(A)$ in the sense of Section 2.1 as the set of exponents of the tail terms of $G(\mu)$ and $\Delta(G(\mu)) \subseteq \Delta(A)$ as the set of exponents which have a non-zero coefficient and are not a monomial square. Moreover, we define $\Delta(G) = \Delta(f) \cup \Delta(-g_1) \cup \cdots \cup \Delta(-g_s)$. Note that $\Delta(G(\mu)) \subseteq \Delta(G) \subseteq \Delta(A)$ for all μ. We have by Section 2.1

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

with coefficients $G(\mu)_{\alpha(j)}, G(\mu)_\beta \in \mathbb{R}$ depending on μ. We use in the last sum that $\Delta(G(\mu)) \subseteq \Delta(G)$ for all $\mu \in [0, \infty)^s$. We set the coefficients $G(\mu)_\beta = 0$ for all $\beta \in \Delta(G) \backslash \Delta(G(\mu))$.

As before, we denote by $\{\lambda_0^{(\beta)}, \ldots, \lambda_r^{(\beta)}\}$ the barycentric coordinates of the lattice point $\beta \in \Delta(A)$ with respect to the vertices of the simplex $\mathrm{New}(G(\mu)) = \mathrm{conv}(A)$. Following [IdW14b] we define for every $\beta \in \Delta(G)$ a set

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

Furthermore, we define the nonnegative real set R as

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

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

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

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

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

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

In [IdW14b, Theorems 5.1 and 5.2] the second and third author show the following.

Theorem 2.9. *Let* γ *be the optimal value of the optimization problem* (2.5)*. Then we have* $f_{\alpha(0)} - \gamma \leqslant s(f, \mathbf{g})$*. The optimization problem* (2.5) *restricted to* $\mu \in (0, \infty)^s$ *is a signomial program if for every* $\beta \in \Delta(G)$ *it holds that* $G(\mu)_\beta$ *has the same sign for every choice of* μ*.*

Assume additionally that every linear form $G(\mu)_{\alpha(j)} = -\sum_{i=0}^s \mu_i \cdot g_{i,\alpha(j)}$ *corresponding to a vertex* $\alpha(j)$ *of* $\mathrm{New}(G(\mu))$ *has only one summand and is strictly positive. Assume moreover that for all* $\beta \in \Delta(G)$ *the linear form* $G(\mu)_\beta = -\sum_{i=0}^s \mu_i \cdot g_{i,\beta}$ *has only positive terms. If furthermore all* $g_{i,\alpha(0)}$ *for* $1 \leqslant i \leqslant s$ *are greater or equal than zero, then* (2.5) *is a geometric program.*

We remark that a $G(\mu)_\beta = -\sum_{i=0}^s \mu_i \cdot g_{i,\beta}$ which has only negative terms can be left out in the above program and the corresponding b_β can be set equal to zero.

3. Constrained Polynomial Optimization via Signomial and Geometric Programming

In this section, we provide relaxations of the program (2.5) 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 3.3. Moreover, we show that under certain extra assumptions the bound obtained by the new program (3.2) will equal the optimal bound $s(f, \mathbf{g})$ from the previous section; see Theorem 3.4. Furthermore, we demonstrate in the following Sections 4 and 5 that the resulting programs can compete with Lasserre relaxations and even outperform them in various cases.

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

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

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

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

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

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

Sums of monomial squares as described in case (3) are ignored in our program (2.5); see also the remark at the very end of Section 2.4. 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 $G(\mu)_{\alpha(j)}$ and $G(\mu)_\beta$ as linear forms in μ since we have for all $j = 0,\ldots,r$

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

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

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

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

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

We consider the following optimization problem in the variables $\mu_1,\ldots,\mu_s$ and $a_{\beta,1},\ldots,a_{\beta,n}, b_\beta$ for every $\beta \in \Delta(G)$.

$$
(3.2)\quad \begin{cases} \text{minimize} & p(\mu, \{(\mathbf{a}_\beta, b_\beta) \ : \ \beta \in \Delta(G)\}) \text{ over the subset of } R \\ & (1) \quad \sum\limits_{\beta \in \Delta(G)} a_{\beta,j} \ \leqslant \ G(\mu)_{\alpha(j)} \ \text{ for all } \ 1 \leqslant j \leqslant r, \\ \text{defined by:} & (2) \quad \prod\limits_{j \in \mathrm{nz}(\beta)} \left(\frac{a_{\beta,j}}{\lambda_j^{(\beta)}}\right)^{\lambda_j^{(\beta)}} \ \geqslant b_\beta, \ \text{ for all } \ \beta \in \Delta(G) \text{ with } \lambda_0^{(\beta)} = 0, \\ & (3) \quad G(\mu)_\beta^+ \leqslant b_\beta \ \text{ for all } \ \beta \in \Delta(G), \ \text{ and} \\ & (4) \quad G(\mu)_\beta^- \leqslant b_\beta \ \text{ for all } \ \beta \in \Delta(G). \end{cases}
$$

Note that this problem is by condition (1) obviously feasible only for choices of μ such that $G(\mu)_{\alpha(j)} > 0$ for all $\alpha(j)$ since all $a_{\beta,j}$ are strictly positive. We set the output as $-\infty$ in all other cases. Indeed, with very little additional assumptions the program (3.2) is a geometric program. Moreover, it is a relaxation of the program (2.5).

Theorem 3.1. *Assume that for every $1 \leqslant j \leqslant r$ the form $G(\mu)_{\alpha(j)} = -\sum_{i=0}^{s} \mu_i \cdot g_{i,\alpha(j)}$ 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 $\mu \in (0,\infty)^s$ is a geometric program. Assume that γ_{gp} denotes the optimal value of (3.2) and γ denotes the optimal value of (2.5). Then we have*

$$f_{\alpha(0)} - \gamma_{\mathrm{gp}} \ \leqslant \ f_{\alpha(0)} - \gamma \ \leqslant \ s(f, \mathbf{g}).$$

Proof. If we restrict ourselves to $\mu \in (0,\infty)^s$, then all functions involved in (3.2) depend on variables in $\mathbb{R}_{>0}$. By assumption every $G(\mu)_{\alpha(j)}$ has exactly one strictly positive term. Thus, we can express constraint (1) as

$$\frac{\sum\limits_{\beta \in \Delta(G)} a_{\beta,j} + G(\mu)_{\alpha(j)}^-}{G(\mu)_{\alpha(j)}^+} \ \leqslant \ 1,$$

with $G(\mu)_{\alpha(j)}^-$ and $G(\mu)_{\alpha(j)}^+$ defined analogously as in (3.1). Since $G(\mu)_{\alpha(j)}^+$ is a monomial the left hand side is a posynomial in μ and $\mathbf{x}$. The constraints (2) – (4) are obviously 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 (2.5) because $\max\{a,b\} \geqslant |a-b|$ for all $a, b \in \mathbb{R}\backslash\{0\}$. Since furthermore $g_{i,\alpha(0)}^+ \geqslant g_{i,\alpha(0)}$ it follows that $\gamma_{\mathrm{gp}} \leqslant \gamma$ by the definitions of (3.2) and (2.5). The last inequality follows from Theorem 2.9. $\square$

One expects the programs (2.5) and (3.2) to have a similar optimal value if, for example, $g_{i,\alpha(0)} \geqslant 0$ for most $i = 1, \ldots, s$ and if one $G(\mu)_\beta^+, G(\mu)_\beta^-$ is identically zero for most $\beta \in \Delta(G)$. Note that one of $G(\mu)_\beta^+, G(\mu)_\beta^-$ is zero if and only if $\max\{G(\mu)_\beta^+, G(\mu)_\beta^-\} =$

$|G(\mu)_\beta^+ - G(\mu)_\beta^-| = |G(\mu)_\beta|$ if and only if the $g_{i,\beta}$ are all $\geqslant 0$ or all $\leqslant 0$ for $i = 0, \ldots, s$.

We give an example to demonstrate how a given constrained polynomial optimization problem can be translated into the geometric program (3.2). We discuss several further examples in the following Section 4.

Example 3.2. 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, we have $\Delta(G) = \{\beta\} = \{(3,2)\}$. Thus, we introduce 4 variables $(a_{\beta,1}, a_{\beta,2}, b_\beta, \mu)$. First, we compute the barycentric coordinates of β and get

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

Now we match the coefficients of $G(\mu)$ to the vertices $\alpha(0) = (0,0), \alpha(1) = (2,4), \alpha(2) = (6,2)$:

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

$\hexagon$

In the following, we strengthen Theorem 2.9 by reformulating the program (3.2) such that it is *always* a signomial program and not only if for every $\beta \in \Delta(G)$ it holds that $G(\mu)_\beta$ has the same sign for every choice of μ. 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

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

We consider the following program.

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

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

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

We obtain the following statement.

Theorem 3.3. *The optimization problem* (3.3) *restricted to* $\mu \in (0,\infty)^s$ *is a signomial program. Assume that* γ_{snp} *denotes the optimal value of* (3.3) *and* $\gamma_{\mathrm{gp}}, \gamma$ *denote the optimal values of* (3.2) *and* (2.5) *as before. Then we have*

$$f_0 - \gamma_{\mathrm{gp}} \ \leqslant \ f_0 - \gamma_{\mathrm{snp}} \ \leqslant \ f_0 - \gamma \ \leqslant \ s(f, \mathbf{g}).$$

Particularly, we have $\gamma_{\mathrm{snp}} = \gamma$ *if the program* (2.5) *attains its optimal value for* $\mu \in (0,\infty)^s$.

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 moderately, then, indeed, the output $f_{\alpha(0)} - \gamma_{\mathrm{gp}}$ of (3.2) equals the output $f_{\alpha(0)} - \gamma$ of (2.5) and particularly the bound $s(f, \mathbf{g})$.

Theorem 3.4. *Assume that for every* $1 \leqslant j \leqslant r$ *the form* $G(\mu)_{\alpha(j)} = -\sum_{i=0}^s \mu_i \cdot g_{i,\alpha(j)}$ *has exactly one strictly positive term. Furthermore, assume that* $g_{i,\alpha(0)} \geqslant 0$ *for all* $i = 1, \ldots, s$, *and that* $\Delta(A_i) \cap \Delta(A_l) = \varnothing$ *for all* $0 \leqslant i < l \leqslant s$. *Let* γ *be the optimal value of the program* (2.5). *If the optimal value* $s(f, \mathbf{g}) = \sup\{G(\mu)_{\mathrm{gp}} : \mu \in \mathbb{R}^s_{\geqslant 0}\}$ *is attained for some* $\mu \in (0,\infty)^s$, *then* $f_{\alpha(0)} - \gamma_{\mathrm{gp}} = f_{\alpha(0)} - \gamma = s(f, \mathbf{g})$, *where, as before,* γ_{gp} *denotes the optimal value of* (3.2).

Note that the condition $\Delta(A_i) \cap \Delta(A_l) = \varnothing$ for all $0 \leqslant i < l \leqslant s$ is particularly satisfied if $\Delta(-g_i) = \varnothing$ for all but one $i = 0, \ldots, s$.

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

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

Furthermore, we have $g_{i,\alpha(0)} \geqslant 0$ for all $i = 1, \ldots, s$ 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 (2.5) and (3.2) coincide.

By assumption, every $G(\mu)_{\alpha(j)}$ 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 $f_{\alpha(0)} - \gamma_{\mathrm{gp}} \geqslant s(f, \mathbf{g})$ for $f_{\alpha(0)} - \gamma_{\mathrm{gp}} = f_{\alpha(0)} - \gamma = s(f, \mathbf{g})$ to hold. Let $\mu^* \in (0, \infty)^s$ be such that $G(\mu^*)_{\mathrm{gp}} = s(f, \mathbf{g})$. By Corollary 2.7 $G(\mu^*)_{\mathrm{gp}}$ is given by a feasible point $(a_{\beta,1}, \ldots, a_{\beta,r})$ of the program

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

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

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

Hence, $f_{\alpha(0)} - \gamma_{\mathrm{gp}} \geqslant G(\mu^*)_{\mathrm{gp}} = s(f, \mathbf{g})$.

$\square$

4. Examples for Constrained Optimization via Geometric Programming and a Comparison to Lasserre Relaxations

One of the main results in [IdW14b] is the observation that lower bounds arising from geometric programming can be better than the bounds obtained by semidefinite programming. This is an important observation considering that geometric programs can be solved more efficiently than semidefinite programs in practice, especially for sparse polynomials with high degree. Here, we show that this observation also holds for constrained polynomial optimization of the form

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

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

$$f_{\mathrm{sos}}^{(d)} \;=\; \sup\left\{r \;:\; f-r=\sum_{i=0}^{s}\sigma_i g_i,\;\; \sigma_i\in\Sigma\mathbb{R}[\mathbf{x}]^2,\;\; \deg(\sigma_i g_i)\leqslant 2d\right\}$$

where $g_0 = 1$ and $2d \geqslant \max_{1\leqslant i\leqslant s}\{\deg(f), \deg(g_i)\}$.

In the following 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].

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

$$K \;=\; \{(x,y)\in\mathbb{R}^2 \;:\; x\geqslant 0\}.$$

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

$$f_{\mathrm{sos}}^{(3)} \;=\; \sup\left\{r \;:\; f-r=\sigma_0+\sigma_1\cdot g_1,\;\; \sigma_0,\sigma_1\in\Sigma\mathbb{R}[\mathbf{x}]^2, \deg(\sigma_j g_j)\leqslant 6\right\} \;=\; -\infty,$$

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

Now, we consider $s(f,g_1) = \sup\{G(\mu)_{\mathrm{gp}} : \mu\in\mathbb{R}_{\geqslant 0}\} \;\leqslant\; f_K^*$ where $G(\mu) = f-\mu_1 g_1$ with $\mu_1\geqslant 0$. Note that $\mathrm{New}(G(\mu))$ is a simplex for every choice of μ. In particular, for $\mu_1 = 0$ we have that $G(\mu)_{\mathrm{gp}} = f_{\mathrm{gp}} = 0$, since the Motzkin polynomial is a SONC polynomial; see Section 2.1 and also [IdW14a]. It follows that

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

and 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 $f_{\mathrm{sos}}^{(d)} \geqslant s(f,\mathbf{g})$ holds in general. ⬠

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 $-\infty = f_{\mathrm{sos}}^{(4)}$

and the optimal solution is given for $d = 8$ with $f_{\mathrm{sos}}^{(8)} \approx 0.4474$. 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 $\mathrm{New}(G(\mu))$ 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:

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

such that the variables satisfy

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

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

$$\gamma_{\mathrm{gp}} \;\approx\; 0.5526$$

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

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

The equality $f_{\alpha(0)} - \gamma_{\mathrm{gp}} = s(f, g_1)$ is not surprising, since the assumptions of Theorem 3.4 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 programing and the Lasserre approach have a runtime below 1 second. However, 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 leave 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, we can see that after multiplying the exponents by 10, the only negative 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 $f^* = 1$ 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 $f^* = 1$ at a minimizer $10^{-7} \cdot (-0.1057, 0.1711)$, which, of course, is the origin up to a numerical error. ◯

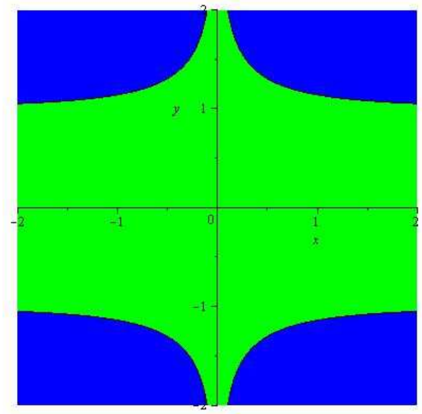

FIGURE 1. The feasible set for the constrained optimization problem in Example 4.2 is the unbounded green area.

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^*, since K is unbounded. Symbolically, we were able to prove a global minimum of $f^* = -15$ with four global minimizers $(2,2,2), (-2,-2,2), (-2,2,-2), (2,-2,-2)$ using the quantifier elimination software SYNRAC, 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 $\Delta(G) = \{\beta, \overline{\beta}, \hat{\beta}\} = \{(1,1,1),(2,1,1),(1,2,1)\}$. Thus, our geometric program has the following 13 variables

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

Hence, program (3.2) is of the form

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

such that

(1) $a_{\beta,1} + a_{\overline{\beta},1} + a_{\hat{\beta},1} \leqslant 1,\ a_{\beta,2} + a_{\overline{\beta},2} + a_{\hat{\beta},2} \leqslant 1,\ a_{\beta,3} + a_{\overline{\beta},3} + a_{\hat{\beta},3} + \mu \leqslant 1,$

(2) $\frac{1}{2} \cdot b_{\overline{\beta}} \cdot \left(a_{\overline{\beta},1}\right)^{-\frac{1}{2}} \cdot \left(a_{\overline{\beta},3}\right)^{-\frac{1}{2}} \leqslant 1,$

$\frac{1}{2} \cdot b_{\hat{\beta}} \cdot \left(a_{\hat{\beta},2}\right)^{-\frac{1}{2}} \cdot \left(a_{\hat{\beta},3}\right)^{-\frac{1}{2}} \leqslant 1,$

(3) $8 \cdot b_\beta^{-1} \leqslant 1,\ \mu \cdot b_\beta^{-1} \leqslant 1,\ \mu \cdot b_{\overline{\beta}}^{-1} \leqslant 1,\ \mu \cdot b_{\hat{\beta}}^{-1} \leqslant 1\,.$

This leads to $\gamma_{\text{gp}} = \frac{1}{256} \cdot 8^4 = 16$ and so $f_{\alpha(0)} - \gamma_{\text{gp}} = -15$. 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

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

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^*, GLOPTIPOLY was not able to certify that the correct bound was found. ◯

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 (♣) is never satisfied since for every $\mu > 0$ we have a vertex $(0,0)$ 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 $s(f, g_1) = f_{\text{gp}}$. Since f is the homogenized Motzkin polynomial we obtain immediately $f_{\text{gp}} = f^* = 0$. 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 $g_1 \geqslant 0$ or $g_1 \leqslant 0$. For further information see [Nie13b, Examples 5.3 and 5.4]. ◯

It is an obvious question for which classes of polynomial optimization problems the geometric programming bound developed in this paper outperforms the Lasserre relaxations in optimality. Even though this seems to be a challenging task, one can answer this question combinatorially. The second and the third author showed in [IdW14a] that a nonnegative circuit polynomial f is a sum of squares if and only if $\text{New}(f)$ has a special lattice point structure. In particular, it was shown that a nonnegative circuit polynomial f cannot be a sum of squares if $\text{New}(f)$ is what is called an *M-simplex* in [IdW14a]. However, f will always be a sum of squares if it is supported on what is called an *H-simplex* in [IdW14a]. Hence, our geometric programming bounds will outperform the Lasserre relaxations in optimality at least if $\text{New}(G(\mu))$ is an M-simplex (and in more cases as it is shown in [IdW14a]). However, whether a simplex is an M-simplex or an H-simplex or something in between is not easy to decide, see [IdW14a]. Therefore, the quality of geometric programming bounds compared to semidefinite programming bounds is very closely related to understanding these combinatorial aspects of the Newton polytopes.

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 $\text{New}(G(\mu)) = \text{conv}\{0, 2d\,e_1, \ldots, 2d\,e_n\}$. 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 $1 \leqslant j \leqslant n$ 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 (2.5) coincides with the program (3) in [GM13]. One drawback of this setting is that the geometric programming bounds obtained from (2.5) are at most as good as the bound $f_{\text{sos}}^{(d)}$, see [GM13]. However, as we have proven in the previous examples, there are also cases where the geometric programming bounds outperform $f_{\text{sos}}^{(d)}$, 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). ⬠

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 $\text{New}(G(\mu))$ is 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 $\text{New}(G(\mu))$ 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 $g_{s+1} = x_1^{2d} + \ldots + x_n^{2d} + c$ 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, it can and will still be outperformed in 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.

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 [IdW14a, IdW14b] 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 [IdW14a, 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 $g_1, \ldots, g_k$ and positive real numbers $\mu_1, \ldots, \mu_k$ such that $f = \sum_{i=1}^k \mu_i g_i$. Then f is nonnegative.*

Of course, it is not obvious how to find a SONC decomposition 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 (♣). We denote

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

such that $f_{\alpha(j)}\mathbf{x}^{\alpha(j)}$ are monomial squares. By (♣), $V(A)$ are the vertices of $\text{New}(f)$ and we have $V(A) \subseteq \{\alpha(0), \ldots, \alpha(d)\}$; equality, however, is not required here. Namely,

$\{\alpha(0), \ldots, \alpha(d)\}$ can also contain exponents of monomial squares in $\Delta(A)\backslash\Delta(f)$ which are not vertices of $\mathrm{conv}(A)$. For simplicity we assume in the following that the affine span of A is n-dimensional. We proceed as follows:

(1) Choose a triangulation $T_1, \ldots, T_k$ of exponents $\alpha(0), \ldots, \alpha(d) \in A$ corresponding to the monomial squares.
(2) Compute the induced covering $A_1, \ldots, A_k$ of A given by $A_i = A \cap T_i$ for $1 \leqslant i \leqslant k$.
(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 $\beta \in A_1, \ldots, A_l$ with $1 < l \leqslant k$. Then we choose $f_{\beta,1}, \ldots, f_{\beta,l} \in \mathbb{R}$ such that $\sum_{i=1}^{l} f_{\beta,i} = f_\beta$ and $\mathrm{sign}(f_{\beta,i}) = \mathrm{sign}(f_\beta)$ for all $1 \leqslant i \leqslant l$. We proceed analogously for $\alpha(0), \ldots, \alpha(d)$.
(4) Define new polynomials $g_1, \ldots, g_k$ such that

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

Note that by (1) and (2) A_i is a set of integers such that $\mathrm{conv}(A_i)$ is a simplex with even vertices and A_i contains no even points corresponding to monomial squares except for the vertices of $\mathrm{conv}(A_i)$. Thus, by (2)–(4) we see that all g_i are ST-polynomials. Namely, 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 $\mathrm{New}(g_i) = \mathrm{conv}(A_i)$; we denote the minimizer by m_i^*. We make the following observation about these minimizers which was similarly already pointed out in [IdW14a, 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 $\alpha \in \mathbb{Z}^{2n}$. 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 (♣) 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 $b_\alpha \mathbf{x}^\alpha \geqslant 0$ it is also nonnegative. □

Proposition 5.3. *Let f, $g_1, \ldots, g_k$, and m_i^* be as before. Assume for $i = 1, \ldots, k$ that m_i^* corresponds to the monomial square $f_{\alpha(j),i}\mathbf{x}^{\alpha(j_i)}$ with $\alpha(j_i) \in \{\alpha(1), \ldots, \alpha(d)\} \cap V(A_i)$. 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 $i = 1, \ldots, l$ with $l \leqslant k$ the exponents $\alpha(j_i)$ are the origin, then $f_{\alpha(0)} - \sum_{i=1}^{l} m_i^*$ is a lower bound for f^*.*

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 is an immediate consequence with the definitions of the m_i^*'s and f^*. □

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 $n-1$ vertex, 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 the following 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

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

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

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

Moreover, we remark that 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 $\tilde{f}_{\mathrm{gp}} \approx 3.572$. ○

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

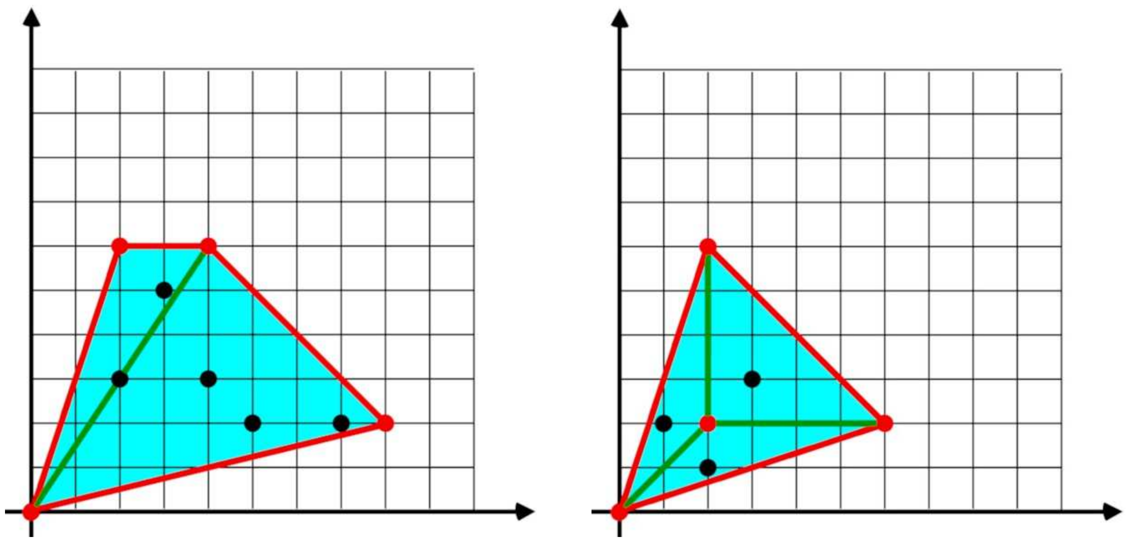

FIGURE 2. The Newton polytopes of the polynomials in the Examples 5.4 and 5.5 and their triangulations.

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

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

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 $g_i - b_ix_1^2x_2^2$ 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:

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

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

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

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 g_1 and g_2 with respect to the constant term; we optimize g_3 with respect to $x_1^2x_2^6$. 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

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

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

$$f_{\mathrm{gp}} \;=\; 1-0.4268 \;=\; 0.5732$$

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

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

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

$$\begin{aligned}
\tilde{g}_1 &= 0.25+2x_1^2x_2^6+1.217x_1^2x_2^2-2x_1x_2^2,\\
\tilde{g}_2 &= 0.75+1x_1^6x_2^2+3.652x_1^2x_2^2-1x_1^2x_2,\\
\tilde{g}_3 &= 1x_1^2x_2^6+1x_1^6x_2^2+1.13x_1^2x_2^2-3x_1^3x_2^3.
\end{aligned}$$

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

$$\tilde{f}_{\mathrm{gp}} \;\approx\; 1-(0.0801+0.2616) \;=\; 0.6583.$$

The corresponding optimal SONC decomposition is given by

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

$\hexagon$

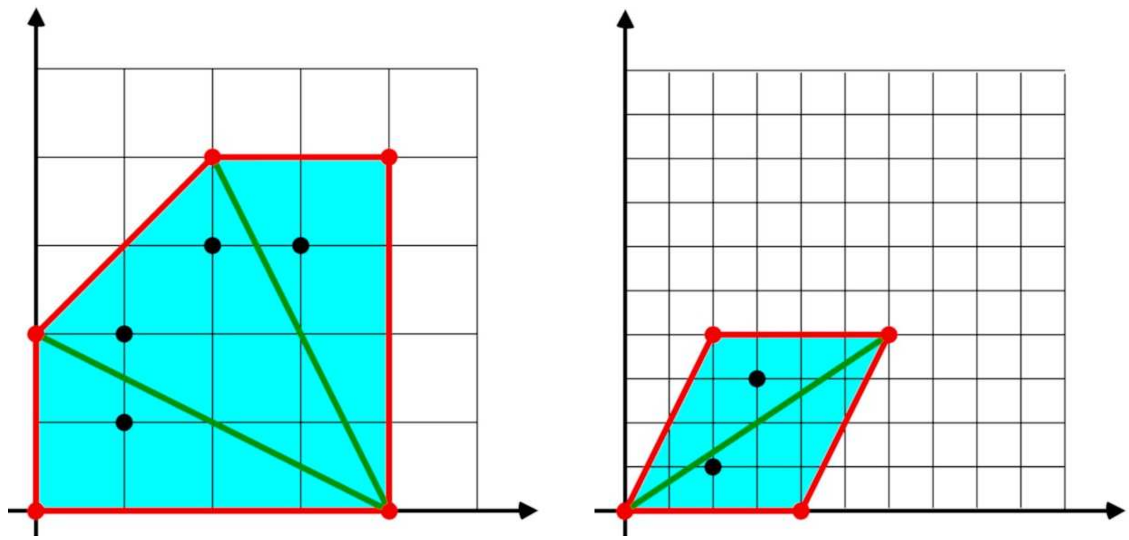

FIGURE 3. The Newton polytopes of the polynomials in the Examples 5.6 and 5.7 and their triangulations.

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

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

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

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

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

$\diamond$

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 still 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 3.4 are fulfilled we know $s(f, g) = 1$.

Checking this optimization problem with Lasserre relaxation, we get $f_{\text{sos}} = f^* = 1$, 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 $f_{\text{sos}} = f^* = 1$ 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. $\diamond$

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, Nonconvex 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, 53–56.

[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.

[GKZ94] I.M. Gelfand, M.M. Kapranov, and A.V. Zelevinsky, *Discriminants, Resultants and Multidimensional Determinants*, Birkhäuser, Boston, 1994.

[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*, 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.

[IdW14a] S. Iliman and T. de Wolff, *Amoebas, nonnegative polynomials and sums of squares supported on circuits*, 2014, To appear in Res. Math. Sci.; see also arXiv:1402.0462.

[IdW14b] ———, *Lower bounds for polynomials with simplex newton polytopes based on geometric programming*, 2014, To appear in SIAM J. Optim.; see also arXiv:1402.6185.

[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.

[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.

[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.

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, Goethe-Universität, FB 12 – Institut für Mathematik, Postfach 11 19 32, 60054 Frankfurt am Main, Germany

E-mail address: `iliman@math.uni-frankfurt.de`

Timo de Wolff, Texas A&M University, Department of Mathematics, College Station, TX 77843-3368, USA

E-mail address: `dewolff@math.tamu.edu`