Approximate GCDs of polynomials and sparse SOS relaxations

Bin Li\textsuperscript{a,1}, Jiawang Nie\textsuperscript{b}, Lihong Zhi\textsuperscript{a}

\textsuperscript{a}Key Laboratory of Mathematics Mechanization, AMSS, Beijing 100080 China
{bli,lzhi}@mmrc.iss.ac.cn; \url{http://www.mmrc.iss.ac.cn/~lzhi/}

\textsuperscript{b}Department of Mathematics, UCSD, 9500 Gilman Drive, La Jolla, CA 92093 USA
njw@math.ucsd.edu; \url{http://www.math.ucsd.edu/~njw/}

Abstract. The problem of computing approximate GCDs of several polynomials with real or complex coefficients can be formulated as computing the minimal perturbation such that the perturbed polynomials have an exact GCD of given degree. We present algorithms based on SOS (Sum of Squares) relaxations for solving the involved polynomial or rational function optimization problems with or without constraints.

Keywords: greatest common divisor, sum of squares, semidefinite programming, global minimization.

1. Introduction

The problem of computing approximate GCDs of several polynomials $f_1, \ldots, f_s \in F[x_1, \ldots, x_n]$, where $F$ is $\mathbb{R}$ or $\mathbb{C}$ can be written as

$$
\min_{p, u_1, \ldots, u_s} \left\| f_1 - p \cdot u_1 \right\|_2^2 + \left\| f_2 - p \cdot u_2 \right\|_2^2 + \cdots + \left\| f_s - p \cdot u_s \right\|_2^2
$$

where $p, u_1, \ldots, u_s \in F[x_1, \ldots, x_n]$ are polynomials such that the total degrees $\text{tdeg}(p) \leq k$, $\text{tdeg}(p \cdot u_i) \leq d_i = \text{tdeg}(f_i)$ for $1 \leq i \leq s$. The minimization problem has many different formulations and various numeric optimization techniques have been proposed, see (Chin and Corless, 1998; Kaltofen et al., 2006a) and references therein. The optimization problem has a globally minimal solution under certain restrictions given in (Kaltofen et al., 2006b). In particular, an algorithm based on global minimization of rational function was proposed in (Karmarkar and Lakshman Y. N., 1996, 1998) to compute approximate GCDs of univariate polynomials. The most expensive part of their algorithm is to find all the real solutions of two bivariate polynomials with high degrees. It has been shown in (Nie et al., 2006) that sum of squares (SOS) relaxation (Lasserre, 2000; Parrilo, 2000) can be used to find the global minimum of the rational
function arises from the approximate GCD computation. The sum of squares programs can be solved by reformulating them as semidefinite programs (SDP), which in turn are solved efficiently by using interior point methods (Nesterov and Nemirovskii, 1993; L. Vandenberghe and Boyd, 1996; Wolkowicz et al., 2000). In the following sections, we show how to apply SOS relaxations to solve different optimization problems formulated in (Chin and Corless, 1998; Karmarkar and Lakshman Y. N., 1996, 1998; Kaltofen et al., 2006a; Nie et al., 2006). The sparsity of the optimization problem has also been exploited.

2. Minimization problems

In this section, we formulate the approximate GCD problem as polynomial or rational function minimization problem with or without constraints. The SOS relaxations are used to solve these optimization problems. We refer to (Parrilo and Sturmfels, 2003; Parrilo, 2000; Lasserre, 2000; Nie et al., 2006; Jibetean and de Klerk, 2006) for description of SOS relaxations and their dual problem.

2.1. Polynomial minimization problem

The minimization problem (1) is a nonlinear least squares problem. As shown in (Chin and Corless, 1998), if a good initial guess is taken, then Newton-like optimization method or Levenberg-Marquardt method can converge very fast to the global optimum. However, if we start with poor initial guess, then these methods may converge to local minimum after taking a large number of iterations.

An entirely different approach was introduced by Shor (Shor, 1987; Shor and Stetsyuk, 1997) and further developed by Parrilo (Parrilo, 2000; Parrilo and Sturmfels, 2003) and Lasserre (Lasserre, 2000). The idea is to express the problem (1) as polynomial minimization problem \( \min_{z \in \mathbb{R}^t} f(z) \) and relax it to the following SOS program:

\[
\begin{align*}
\sup_{r \in \mathbb{R}, W} r^*_{sos} &:= \sup_{r \in \mathbb{R}, W} r \\
\text{s.t.} & \quad f(z) - r = m_d(z)^T W m_d(z) \\
W & \succeq 0
\end{align*}
\]

where the objective polynomial

\[
f(z) = \sum_{i=1}^{s} \left\| f_i - p \cdot u_i \right\|^2 = \sum_{i=1}^{s} \sum_{|\alpha| \leq d_i} |f_{i,\alpha} - \sum_{\beta+\gamma = \alpha} p_{\beta} u_{i,\gamma}|^2
\]

and \( m_d(z) \) is the column vector of all monomials up to degree \( d = \lceil \frac{\deg(f)}{2} \rceil = 2 \).

Denote the numbers of indeterminants in the coefficients of \( p, u_1, \ldots, u_s \) by \( n(p) \), \( n(u_1), \ldots, n(u_s) \) respectively (see the remark below for details) and the variables

\[
z = \{p_1, \ldots, p_{n(p)} \} \cup \left( \bigcup_{i=1}^{s} \{u_{i,1}, \ldots, u_{i,n(u_i)} \} \right).
\]
The length of real symmetric matrix $W$ is $\left(\begin{array}{c} t+2 \\ 2 \end{array}\right)$ and there are $\left(\begin{array}{c} t+4 \\ 4 \end{array}\right)$ equality constraints in (2) for $t = n(p) + \sum_{i=1}^{s} n(u_i)$.

**Remark 2.1** If $F = \mathbb{R}$, the coefficients of $p, u_i$ are real numbers and therefore $n(p) = \binom{n+k}{n}$, $n(u_i) = \binom{n+d_i-k}{n}$. If $F = \mathbb{C}$, the coefficients of $p, u_i$ are complex numbers and we have $n(p) = 2\binom{n+k}{n}$, $n(u_i) = 2\binom{n+d_i-k}{n}$ by separating real and imaginary parts of each coefficient. If we assume that $p$ is monic, then $n(p)$ is $\binom{n+k}{n} - 1$ in real case or $2\binom{n+k}{n} - 1$ in complex case.

Write $f(z) = \sum_{\alpha} f_{\alpha} z^\alpha$, then the dual SDP problem of the SOS program (2) can be described as (Lasserre, 2000):

$$r^*_{mom} := \inf_{y} \sum_{\alpha} f_{\alpha} y_{\alpha}$$

$$\text{s.t.} \quad y_{0\ldots0} = 1$$

$$M_d(y) \succeq 0$$

where $M_d(y) := (y_{\alpha+\beta})_{0 \leq |\alpha|, |\beta| \leq d}$ is called the $d$-th moment matrix of the real vector $y$. When the computed moment matrix $M_d(y^*)$ satisfies some flat extension conditions, the global minimum is achieved and global minimizers can be extracted numerically by solving an eigenvalue problem (Henrion and Lasserre, 2005).

### 2.2. Rational function minimization

Let $f_i, u_i, p$ be coefficient vectors corresponding to polynomials $f_i, u_i, p$ respectively, and $A_i = A_i(p)$ be convolution matrices such that $A_i u_i$ produce the coefficient vector of $p \cdot u_i$. Then the straight-forward formulation of the minimization problem (1) can be written as:

$$\min_{p, u_1, \ldots, u_s} \|f_1 - A_1 u_1\|^2_2 + \cdots + \|f_s - A_s u_s\|^2_2.$$  

(4)

If we fix the coefficients of $p$, the minimum is achieved at

$$u_i := (A^*_i A_i)^{-1} A^*_i f_i, \quad 1 \leq i \leq s,$$  

(5)

and the minimization problem becomes

$$\min_p \sum_{i=1}^{s} (f_i^* f_i - f_i^* A_i (A_i^* A_i)^{-1} A_i^* f_i).$$  

(6)

Here and hereafter $A_i^*$ and $f_i^*$ denote the conjugate transpose of $A_i$ and $f_i$ respectively. This is an unconstrained minimization problem of rational function with positive denominator

$$\text{lcm} (\text{det} (A_i^* A_i), \ldots, \text{det} (A_s^* A_s)).$$

for computing approximate GCDs of univariate polynomials and in (Hitz et al., 1999) for computing nearest bivariate polynomials with a linear (or fixed degree) factor.

Express the minimization problem (6) as \( \min_{z \in \mathbb{R}^d} \frac{f(z)}{g(z)} \), where \( f(z), g(z) \in \mathbb{R}[z_1, \ldots, z_r] \) and \( g(z) \) is a real positive definite polynomial. Similar to the polynomial minimization problem, it can be transferred to a constrained SOS program (Nie et al., 2006):

\[
\begin{align*}
\min_{z \in \mathbb{R}^d} & \quad r \\
\text{s.t.} & \quad f(z) - rg(z) = m_d(z)^T W m_d(z) \\
& \quad W \succeq 0
\end{align*}
\]

Here \( z = \{p_1, \ldots, p_n(p)\} \), and \( m_d(z) \) is the column vector of all monomials up to degree \( d = \lceil \max(t \deg(f), t \deg(g)) \rceil \) where \( t \deg(f) \leq t \deg(g) \leq 2 \sum_{i=1}^s (n+d_i-k_i) \). The length of the real symmetric matrix \( W \) is \( \binom{t+d}{t} \) and there are \( \binom{t+2d}{t} \) equality constraints in (7) for \( t = n(p) \). We can see that there is a trade off between the number of variables and the degrees of polynomials.

**Example 2.1** Consider two polynomials

\[
\begin{align*}
f_1(x) &= x(x+1)^2, \\
f_2(x) &= (x-1)(x+1)^2 + 1/10
\end{align*}
\]

and \( k = 2, F = \mathbb{R} \). Solving the SOS program (2) and its dual problem with

\[
\begin{align*}
f(z) &= \|f_1 - p \cdot u_1\|^2_2 + \|f_2 - p \cdot u_2\|^2_2 = p_1^2 u_{1,1}^2 + (1 - p_1 u_{1,2} - p_2 u_{1,1})^2 \\
&\quad + (2 - p_2 u_{1,2} - u_{1,1})^2 + (1 - u_{1,2})^2 + (-9/10 - p_1 u_{2,1})^2 \\
&\quad + (-1 - p_1 u_{2,2} - p_2 u_{2,1})^2 + \left(1 - u_{2,1} - p_2 u_{2,2}\right)^2 + (1 - u_{2,2})^2
\end{align*}
\]

we get the minimal value \( r_{sos}^* \approx 9.3876e - 4 \). The length of matrix \( W \) in the corresponding SDP problem (2) is 28. From the optimal dual solutions, we find that the global minimum is achieved and the minimizer can be extracted:

\[
z^* \approx (0.9335, 1.9778, 0.02569, 1.0013, -0.9739, 0.9975).
\]

It corresponds to the monic approximate GCD

\[
p(x) \approx 0.9335 + 1.9778x + x^2
\]

with cofactors \( u_1(x) \approx 0.02569 + 1.0013x, u_2(x) \approx -0.9739 + 0.9975x \).

Solving the SOS program (7) and its dual problem with

\[
\begin{align*}
f(z) &= 12/5 p_2 p_1 + 7 p_4^1 + 281/100 p_4^1 - 281/50 p_2^2 p_1 + 11/5 p_1^2 p_2^2 \\
&\quad - 6 p_3^1 + 9/5 p_4^1 + 981/100 p_1^2 + 581/100 p_4^1 + 281/100 \\
&\quad - 6 p_2^2 p_1^1 + 9/5 p_2^3 p_1^1 + 9/5 p_4^1 - 2 p_2 - 2 p_2^3 p_1^1, \\
g(z) &= p_1^1 + p_2^2 p_2^2 + 2 p_1^1 + p_2^4 + p_2^2 + 1 - 2 p_2^2 p_1^1
\end{align*}
\]
we get the minimal value \( r_{sos}^* \approx 9.3876e - 4 \). The length of matrix \( W \) in the corresponding SDP problem (7) is 6. From the optimal dual solutions, we can extract the minimizer \( z^* \approx (0.9335, 1.9778) \). Evaluating the rational function at \( z^* \) shows that

\[
\frac{f(z^*)}{g(z^*)} \approx 9.3876e - 4 \approx r_{sos}^*
\]

which implies that \( z^* \) is the global minimizer. It corresponds to the same monic approximate GCD \( p(x) \).

**Example 2.2** Consider two polynomials

\[
f_1(x_1, x_2) = x_1^3 + 2x_1x_2 + x_2^3 - 1, \quad f_2(x_1, x_2) = x_1^3 + x_1x_2 - x_2 - 1.01
\]

and \( k = 1, F = \mathbb{R} \). Solving the SOS program (2) and its dual problem with

\[
f(z) = \|f_1 - p \cdot u_1\|_2^2 + \|f_2 - p \cdot u_2\|_2^2 = (-1 - p_1 u_{1,1})^2 + (-p_1 u_{1,3} - p_3 u_{1,1})^2 \\
+ (-p_1 u_{1,2} - p_2 u_{1,1})^2 + (2 - p_2 u_{1,3} - p_3 u_{1,2})^2 + (1 - p_3 u_{1,3})^2 \\
+ (1 - p_2 u_{1,2})^2 + (-1.01 - p_1 u_{2,1})^2 + (-p_1 u_{2,3} - p_3 u_{2,1})^2 \\
+ (-1 - p_1 u_{2,2} - p_2 u_{2,1})^2 + (1 - p_2 u_{2,3} - p_3 u_{2,2})^2 + (1 - p_3 u_{2,3})^2 + p_2 u_{2,2}^2
\]

we get the minimal value \( r_{sos}^* \approx 3.89306e - 5 \). The length of matrix \( W \) in the corresponding SDP problem (2) is 55.

Solving the SOS program (7) and its dual problem with

\[
f(z) = -20.02p_2 p_3^3 p_1^2 + 26.0804p_1^2 p_3 p_2^2 - 22.04p_3 p_2 p_1^2 - 22.02p_3 p_2^3 p_1 \\
+ 5.98p_1^3 p_2^2 + 9p_1^2 p_3^2 + 6.0001p_1^2 p_2^2 + 10.0201p_1^2 p_2^2 + 13.0402p_2^2 p_3^2 \\
+ 14.0402p_2^2 p_3^3 + 8p_1^6 + 4.0201p_3^6 + 4.0201p_3^6 - 10p_3 p_2^3 - 6p_2 p_3^5 \\
- 4p_3 p_2^3 - 6.04p_3 p_3^5 p_2^2 - 2p_2 p_3^2 p_1 - 6.02p_1 p_3 p_2^2 - 4.02p_1 p_2 p_3^2 p_2 \\
- 2.02p_1^3 p_3^2 - 2.02p_1 p_3^5 + 2p_2 p_3^5 + 2p_2 p_3^5 + 2p_2 p_3^5 + 2p_1 p_2^5, \\
g(z) = p_1^6 + 2p_1^3 p_2^2 + 2p_1^3 p_3^2 + 2p_1^3 p_2^2 + 5p_1^3 p_3^2 p_2^2 + 2p_2^2 p_3^2 + p_6 + 2p_1^2 p_2^2 \\
+ 2p_2^2 p_3^3 + p_3^3
\]

we get the minimal value

\[
r_{sos}^* \approx 3.89306e - 5.
\]

Here \( f(z), g(z) \) are homogeneous with the coefficients of \( p \). The length of matrix \( W \) in the corresponding SDP problem (7) is 10. From the optimal dual solutions, we get an approximate GCD

\[
p(x) \approx 1.00199 + 0.99937x_2 + x_1.
\]
Example 2.3 Consider two polynomials
\[
\begin{align*}
    f_1(x) &= (x - 0.3)(x + 4.6)(x - 1.45)(x + 10) \\
    f_2(x) &= (x - 0.301)(x + 4.592)(x - 1.458)(x - 0.6)(x - 15)(x + 2)
\end{align*}
\]
and \( k = 3, F = \mathbb{R} \). Solving the SOS program (2) and its dual problem we get \( r_{sos}^* \approx 0.0156 \). The length of matrix \( W \) in the corresponding SDP problem (2) is 55. Solving the SOS program (7) and its dual problem we get the minimal value \( r_{sos}^* \approx 0.0156 \). The length of matrix \( W \) in the corresponding SDP problem (7) is 84.

Example 2.4 (Kaltofen et al., 2006a) Consider the polynomials
\[
\begin{align*}
    f_1 &= 1000x^{10} + x^3 - 1 \\
    f_2 &= x^2 - \frac{1}{100}
\end{align*}
\]
and \( k = 1, F = \mathbb{R} \). Solving the SOS program (7) and its dual problem we get \( r_{sos}^* \approx 0.042157904 \). The length of matrix \( W \) in the corresponding SDP problem (7) is 13. It was shown in (Kaltofen et al., 2006a), after about ten iterations in the average, the STLN algorithm converges to the following local minima:
\[
0.0421579, 0.0463113, 0.0474087, 0.0493292, \ldots
\]
for different initializations.

Example 2.5 (Kaltofen et al., 2006b) Consider two polynomials
\[
\begin{align*}
    f_1(x) &= x^2 + 2x + 1, \quad f_2(x) = x^2 - 2x + 2
\end{align*}
\]
and \( k = 1, F = \mathbb{R} \). For \( p(x) = \epsilon x + 1, u_1(x) = 2x + 1, u_2(x) = -2x + 2 \), the value of \( \|f_1 - p \cdot u_1\|_2^2 + \|f_2 - p \cdot u_2\|_2^2 \) can be arbitrarily near to 2. Solving the SOS program (2) and its dual problem we get \( r_{sos}^* \approx 2.000569 \) and an approximate GCD
\[
p(x) \approx x - 14686.677911.
\]
Solving the SOS program (7) and its dual problem we get the minimal value \( r_{sos}^* \approx 2.000000 \), but extract no minimizers.

2.3. Minimization problem with constraints
As in (Kaltofen et al., 2007, 2006a,b), the problem of computing approximate GCDs of several polynomials can also be formulated as
\[
\begin{align*}
\min & \quad \|\Delta c\|_2^2 \\
\text{s.t.} & \quad S_k(c + \Delta c)x = 0, \quad \exists \ x \neq 0
\end{align*}
\]
where $c$ is the coefficient vector of $f_1, \ldots, f_s$, and the perturbations to the polynomials are parameterized via the vector $\Delta c$, and $S_k(c + \Delta c)$ is the multi-polynomial generalized Sylvester matrix (Kaltofen et al., 2006a). The minimization problem (8) is a quadratic optimization problem with quadratic constraints.

Similar to the method used in (Kaltofen et al., 2006a,b), we can choose one column of $S_k$ and reformulate the problem as

$$\min_{\Delta c, x} \left\| \Delta c \right\|^2_2 + \rho \left\| x \right\|^2_2, \quad (9)$$

$$s.t. \quad A(c + \Delta c)x = b(c + \Delta c).$$

Two alternative formulations are

$$\min_{\Delta c, x} \left\| \Delta c \right\|^2_2, \quad (10)$$

$$s.t. \quad S_k(c + \Delta c)x = 0,$$

$$\left\| x \right\|^2_2 = 1,$$

and

$$\min_{\Delta c, x} \left\| \Delta c \right\|^2_2 + \rho \left\| x \right\|^2_2, \quad (11)$$

$$s.t. \quad S_k(c + \Delta c)x = 0,$$

$$v^T x = 1,$$

where $\rho$ is a small number and $v$ is a random vector. The dimensions of the vectors $\Delta c, x$ are $\sum_{i=1}^s \binom{n+di}{n}$ and $\sum_{i=1}^s \binom{n+di-k}{n}$ respectively.

Let us describe the polynomial minimization problem with constraints as:

$$\min_{x \in \mathbb{R}^t} \sum_{\alpha} f_\alpha x^\alpha, \quad (12)$$

$$s.t. \quad h_1(z) \geq 0, \ldots, h_l(z) \geq 0.$$ We can reformulate it as a convex LMI (Linear Matrix Inequality) optimization problem (or semidefinite program):

$$\inf_y \sum_{\alpha} f_\alpha y_\alpha, \quad (13)$$

$$s.t. \quad y_{0,...,0} = 1,$$

$$M_d(y) \succeq 0,$$

$$M_{d-w_i}(h_i y) \succeq 0, \quad 1 \leq i \leq l$$

where $w_i := \lceil \frac{\deg(h_i)}{2} \rceil$ for $1 \leq i \leq l$, $d \geq \max\{\lceil \frac{\deg(f(z))}{2} \rceil, w_1, \ldots, w_l\}$, $M_d(y)$ and $M_{d-w_i}(h_i y)$ are moment matrices of real vector $y$ defined in (Lasserre, 2000).
Example 2.6 Consider two polynomials

\[ f_1(x) = x^3 - 1, \quad f_2(x) = x^2 - 1.01 \]

and \( k = 1, F = \mathbb{R}, \rho = 10^{-6} \). We choose the first column of \( S_1 \) to be \( b \) and the remaining columns to be matrix \( A \). For minimization problem (9), the minimal perturbation computed by the first-order \((d = 1)\) semidefinite programs is \( 9.9673e - 6 \). The length of the matrix involved in the corresponding SDP is 152. The minimal perturbation computed by the second-order \((d = 2)\) semidefinite program is \( 2.0871e - 5 \). The length of the matrix involved in the corresponding SDP is 6476. The minimizer can be extracted by the second-order semidefinite program.

For minimization problem (10), the lower bounds given by the first and second order semidefinite programs are 0 and 2.0852e - 5 respectively. Here we notice that one feasible solution corresponding to the first-order relaxation in the homogenous model (10) is \( \Delta c = 0, \ x = [0, 1]^T \) with objective value zero.

As pointed out by Erich Kaltofen, if we want to compute the lower bound for the minimization problem (8) by solving problem (9), we have to try all the possible selection of \( b \), which is very time consuming. So we suggest the formulation (11). For minimization problem (11), the lower bounds given by the first and second order semidefinite programs depend on the choice of random vector \( v \). The obtained lower bounds are around \( 10^{-6} \) and \( 10^{-5} \) respectively.

The experiments show that the first-order semidefinite programs give us some useful information on the minimal perturbations. Although we may compute the global minimizer from high-order semidefinite programs, the sizes of the matrices increase quickly.

3. Exploit sparsity in SOS relaxation

In this section, we investigate how to reduce the size of the SOS program by exploiting the special structures of the minimization problems involved in the approximate GCD computation. Examples 2.1 and 2.2 show that the SOS relaxations are dense for the rational function formulation. So in the following, we only exploit the sparsity in the polynomial formulation SOS program (2). The same technique can be applied to the problem (8).

3.1. Exploit Newton polytope

Given a polynomial \( p(x) = \sum_{\alpha} p_{\alpha} x^{\alpha} \), the cage of \( p \), \( C(p) \), is the convex hull of \( \text{supp}(p) = \{ \alpha | p_{\alpha} \neq 0 \} \).

Theorem 3.1 (Reznick, 1978) For any polynomial \( p \), \( C(p^2) = 2C(p) \); for any positive semidefinite (PSD) polynomials \( f \) and \( g \), \( C(f) \subseteq C(f + g) \); if \( f = \sum g_j^2 \) then \( C(g_j) \subseteq \frac{1}{2} C(f) \).
Corollary 3.2 For any PSD polynomials $f$ and $g$, $C(f + g) = C(\text{supp}(f) \cup \text{supp}(g))$; if $f = \sum g_j^2$ then $C(f) = 2C(\bigcup \text{supp}(g_j))$.

Proof. Since $f$ and $g$ are PSD polynomials, according to Theorem 3.1, we have $C(f) \subseteq C(f + g)$ and $C(g) \subseteq C(f + g)$. From

\[ C(f + g) = C(\text{supp}(f + g)) \subseteq C(\text{supp}(f) \cup \text{supp}(g)) = C(C(f) \cup C(g)) \subseteq C(f + g), \]

it is clear that $C(f + g) = C(\text{supp}(f) \cup \text{supp}(g))$. If $f = \sum g_j^2$, then $C(f) = C(\bigcup C(g_j^2)) = 2C(\bigcup C(g_j)) = 2C(\bigcup \text{supp}(g_j))$. □

The SOS program (2) of the polynomial form of approximate GCD problem is to compute polynomials $h_j(z)$ such that

\[ f(z) - r = m_d(z)^T W m_d(z) = \sum h_j(z)^2. \]

By Theorem 3.1 and Corollary 3.2, we have

\[ C(h_j) \subseteq \frac{1}{2} C(f(z) - r) = C(O \cup \left( \bigcup_{i=1}^s \text{supp}(f_{i,\alpha} - \sum_{\beta + \gamma = \alpha} p_{\beta u_{i,\gamma}}) \right)) \]

where $O$ is the origin. For convenience, let $n_1(p) = n(p) + 1$ if we specify $p_{n_1(p)} = 1$, otherwise $n_1(p) = n(p)$. Then we know that 1 and $p_j u_{i,k}$ for $1 \leq j \leq n_1(p), 1 \leq i \leq s, 1 \leq k \leq n(u_i)$ are all monomials existing in $f_{i,\alpha} - \sum_{\beta + \gamma = \alpha} p_{\beta u_{i,\gamma}}$.

Let $b$ be a lattice point in the convex hull $C(h_j)$. According to the property of convex hull, there exist $\lambda_{j,i,k} \geq 0$ such that

\[ z^b = \prod_{i,j,k} (p_j u_{i,k})^{\lambda_{j,i,k}} = \prod_j p_j^{e_j} \prod_{i,k} u_{i,k}^{e_{i,k}}. \]

The exponents $e_j = \sum_{i,k} \lambda_{j,i,k}$ and $e_{i,k} = \sum_j \lambda_{j,i,k}$ are nonnegative integers. Because $\sum_j e_j = \sum_{i,k} e_{i,k} = \sum \lambda_{j,i,k} \leq 1$, so the monomial $z^b$ can only be 1 or $p_j u_{i,k}$ for some $j, i, k$. It means that we only need these monomials in the SOS relaxation of approximate GCD problem.

Let $m_G(z)$ be the column vector of monomials 1 and $p_j u_{i,k}$ for $1 \leq j \leq n_1(p), 1 \leq i \leq s, 1 \leq k \leq n(u_i)$. The sparse SOS program of the polynomial minimization problem (1) is:

\[ r_{sos1}^* := \sup_{r \in \mathbb{R}} r \]

\[ s.t. \quad f(z) - r = m_G(z)^T W m_G(z) \]

\[ W \succeq 0. \]

Let $n(u) = \sum_{i=1}^s n(u_i)$, the length of the real symmetric matrix $W$ is $1 + n_1(p)n(u)$ and there are $1 + n_1(p)n(u) + \binom{n_1(p)+1}{2}\binom{n(u)+1}{2}$ equality constraints.
3.2. Exploit correlative sparsity

Since the polynomial $f(z)$ in SOS program (2) is written as

$$f(z) = \sum_{i=1}^{s} \| f_i - p \cdot u_i \|_2^2,$$

we can define the subsets

$$z_{\Delta i} = \{ p_1, \ldots, p_{n(p)} \} \cup \{ u_{i,1}, \ldots, u_{i,n(u_i)} \}.$$

The collections of variables $z_{\Delta 1}, \ldots, z_{\Delta s}$ satisfy the following running intersection property: for every $k = 1, \ldots, s - 1$,

$$z_{\Delta k+1} \cap \bigcup_{j=1}^{k} z_{\Delta j} \subseteq z_{\Delta i} \text{ for some } 1 \leq i \leq k.$$

According to (Waki et al., 2006; Lasserre, 2006; Nie and Demmel, 2007), we are going to find the maximum $r$ such that

$$r^{*}_{sos2} := \sup_{r \in \mathbb{R}} r$$

$$s.t. \quad f(z) - r = \sum_{i=1}^{s} m(z_{\Delta i})^T W_i m(z_{\Delta i}) \leq 0, \ 1 \leq i \leq s$$

where $m(z_{\Delta i})$ is the column vector of all monomials up to degree 2. The length of $W_i$ is $\left( \frac{n(p) + n(u_i)}{2} \right)^2$.

The following sparse SOS program is obtained by considering both the Newton polytope and correlative sparsity:

$$r^{*}_{sos3} := \sup_{r \in \mathbb{R}} r$$

$$s.t. \quad f(z) - r = \sum_{i=1}^{s} m_{G_i}(z)^T W_i m_{G_i}(z) \leq 0, \ 1 \leq i \leq s$$

where $m_{G_i}(z)$ is the column vector of monomials 1 and $p_ju_{i,k}$ for $1 \leq j \leq n_1(p), 1 \leq k \leq n(u_i)$. The length of $W_i$ is $1 + n_1(p)n(u_i)$.

3.3. Comparison of sparsity strategies

The relation between the optimums of polynomial minimization problem (1), the SOS program (2) and the three sparse SOS programs (14),(15),(16) is

$$r^* \geq r^{*}_{sos} = r^{*}_{sos1} \geq r^{*}_{sos2} \geq r^{*}_{sos3}.$$
The sizes of the SDP matrices in the three kinds of sparse SOS programs are:

\[ m_1 = (1 + n_1(p)u(u))^2, m_2 = \sum_{i=1}^s (n(p)+n(u_i)+2)^2, m_3 = \sum_{i=1}^s (1 + n_1(p)u(u_i))^2. \]

We have that

\[ s \cdot m_2 \geq s \cdot m_3 = s \sum_{i=1}^s (1 + n_1(p)u(u_i))^2 \geq (s + n_1(p)u(u))^2 \geq m_1 \geq m_3. \]

We show in the Table 1 experiments of applying four kinds of SOS relaxations (2), (14), (15), (16) to compute an approximate GCD of three pairs of polynomials \( f_1 \) and \( f_2 \). We notice that the first and third kinds of sparse SOS programs can reduce the size of the optimization problem remarkably. However, the third kind of sparse SOS program can only give a lower bound in general.

4. Implementation and experiments

The methods described above have been implemented by the first author in Matlab based on algorithms in SOSTOOLS (Prajna et al., 2002), YALMIP (Löfberg, 2004) and SeDuMi (Sturm, 1999). We apply Gauss-Newton iterations to improve the accuracy of the results computed by SDP solvers.

In the following Table 2, we compare the minimal residues achieved by different methods for examples in (Chin and Corless, 1998). The third to the fifth columns are the minimal residues computed by SOS programs (2), (7), (14) respectively. The sixth column consists of the minimal residues refined by applying Newton iteration. The last column consists of the minimal residues computed by STLN method in (Kaltofen et al., 2007, 2006a).
Table 2. Experimental results of examples in (Chin and Corless, 1998).

<table>
<thead>
<tr>
<th>$d_i$</th>
<th>$k$</th>
<th>polynomial</th>
<th>rational</th>
<th>poly. sparse</th>
<th>Newton</th>
<th>STLN</th>
</tr>
</thead>
<tbody>
<tr>
<td>5,4</td>
<td>2</td>
<td>1.620473e-8</td>
<td>1.579375e-8</td>
<td>1.560388e-8</td>
<td>1.560294e-8</td>
<td>1.560294e-8</td>
</tr>
<tr>
<td>4,6</td>
<td>3</td>
<td>1.561803e-2</td>
<td>1.561770e-2</td>
<td>1.561754e-2</td>
<td>1.561754e-2</td>
<td>1.561754e-2</td>
</tr>
<tr>
<td>3,3</td>
<td>2</td>
<td>1.702596e-2</td>
<td>1.702596e-2</td>
<td>1.702596e-2</td>
<td>1.702596e-2</td>
<td>1.702596e-2</td>
</tr>
<tr>
<td>3,2,3</td>
<td>2</td>
<td>1.729192e-5</td>
<td>1.729175e-5</td>
<td>1.729175e-5</td>
<td>1.729175e-5</td>
<td>1.729175e-5</td>
</tr>
</tbody>
</table>

In the Table 3, we show the experimental results of random examples generated in the same way described in (Kaltofen et al., 2007, 2006a). The first example is solved by the SOS program (7) and the other examples are solved by the SOS program (14).

5. Conclusions

In this paper, we discussed how to solve approximate GCD problem which can be formulated as an unconstrained quartic polynomial optimization problem about the coefficients of factor polynomials. This is a nonconvex nonlinear least squares problem and it is usually very difficult to find global solutions. This paper proposed various semidefinite relaxation methods for solving this special polynomial optimization. The usual SOS relaxation is often very good to find global solutions, but it is expensive to solve big problems. By exploiting the special sparsity structures of the quartic polynomial arising from the GCD approximations, we proposed various sparse SOS relaxations based on different formulations and sparsity techniques. The numerical experiments are also implemented to show the effectiveness of these different relaxation methods.

There is a trade-off in choosing these various sparse relaxation methods. The sparse SOS relaxation (14) is the best in quality (it has the same quality as the dense SOS relaxations), but it is the most expensive one in these relaxations. The sparse SOS relaxation (16) has the least quality, but it is the cheapest one and can solve big problems. In practice, to solve GCD problems of big size, we suggest to apply the relaxation (16) to find one approximate solution, and then apply local methods like STLN to refine the solution.

The GCD problem can also be equivalently formulated as an unconstrained rational function optimization (7). This formulation is faster than the polynomial SOS program (2) when there are only few variables and the degree of GCD is very small. However, the problem (7) is very difficult to solve when the GCD problem has big size. It is also an interesting work to exploit the special structures of (7) and get more effective methods.

The strength of SOS relaxation methods is that they do not require an initial guess of solutions and can always return a lower bound of the global minimum. When this lower bound is achieved, we immediately know the global solution is found. Our
preliminary experiments show that these SOS relaxation methods work well in solving the GCD problems. They often return global solutions.

Our proposed sparse SOS relaxation methods are based on the nonlinear least squares formulation (1). Since the GCD problem can also be equivalently formulated as (8), it is also possible to exploit special structure of (8). An interesting future work is to get more effective semidefinite relaxations for (9)-(11) based on their structures.

Acknowledgement: The authors would like to thank Erich Kaltofen, Zhengfeng Yang and Kartik Sivaramakrishnan for a number of interesting discussions on SDP for approximate polynomial computation. We are also grateful to the IMA in Minneapolis for hosting us during our collaboration on this project.

References


