Polynomial optimization from an algebraic geometric viewpoint

François Glineur

UCL/CORE
glineur@core.ucl.ac.be

MOPTA'03

July 29$^{\text{th}}$ 2003

Plan

Single variable case

◇ Newton-type methods, bisection techniques
◇ Sum of squares reformulation, companion matrix

Algebraic geometry

◇ Division, ideals, varieties and Groebner bases
◇ Elimination method
◇ Stetter and Möller method (companion matrix)
◇ Hanzon and Jibetean method

Other methods

◇ Resultant methods, sum of squares relaxations

Concluding remarks

Introduction

Definitions

A *monomial* with n variables

$$x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \quad \text{with } x \in \mathbb{R}^n \text{ and } \alpha \in \mathbb{N}^n$$

A *polynomial* is a finite linear combination of monomials with coefficients in a field k

$$f = \sum_{\alpha \in A} a_\alpha x^\alpha \quad \text{with } a_\alpha \in k \text{ and } A \subset \mathbb{N}^n \text{ finite}$$

The set of polynomials over k is denoted $k[x_1, x_2, \ldots, x_n]$
Our main concern is $\mathbb{R}[x_1, x_2, \ldots, x_n]$

Polynomial optimization

Given a polynomial $f_0 \in \mathbb{R}[x_1, x_2, \ldots, x_n]$, we will mainly consider *unconstrained polynomial optimization*:

$$\min_{x \in \mathbb{R}^n} f_0(x_1, x_2, \ldots, x_n)$$

- Obviously many applications
- Global optimization problem ($\exists$ local minima)
- NP-hard problem (w.r.t. number of variables) (take $f_0 = \sum_i (x_i^2 - 1)^2 \rightarrow$ binary constraints)
- May have a finite of infinite number of solutions
- May exhibit solutions at infinity: $x_1^2 x_2^4 + x_1 x_2^2 + x_1^2$

Polynomial optimization (cont.)

◇ Strong links with *polynomial equation solving*:

find all $x \in \mathbb{R}^n$ such that $f_i(x) = 0$ for all $1 \leq i \leq m$

◇ Stationarity conditions

$$\frac{\partial f_0}{\partial x_1} = \frac{\partial f_0}{\partial x_2} = \cdots = \frac{\partial f_0}{\partial x_n} = 0$$

define a finite number of connected components: check each to find global minimum

◇ Some methods will also be applicable to the *constrained* case without much change

$$\min_{x \in \mathbb{R}^n} f_0(x) \text{ such that } f_i(x) = 0 \text{ for all } 1 \leq i \leq m$$

Single variable case

$$\min_{x\in\mathbb{R}} a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$$

Various available solution techniques:

- ◇ Newton methods and variants
- ◇ Bisection techniques
- ◇ Sum of squares convex reformulation
- ◇ Companion matrix

Utility:

- ◇ Model single variable problems
- ◇ Use as a subroutine for multiple variable case

Newton methods and variants

- ◇ Iterative schemes
- ◇ Usually very fast convergence to local minimum
- ◇ Multiple zeros might cause trouble
- ◇ Convergence only guaranteed if starting point is close to a (local) minimum

$\Rightarrow$ Local technique without globalization
not suitable to find (all) global minimizers

Bisection techniques

◇ Principle: solve stationarity condition as polynomial equation, using *Weyl's quadtree algorithm*

Principle of Weyl's quadtree algorithm to find all complex roots of polynomial equation $p(x) = \sum_{i=0}^{n} a_i x^i = 0$

a. Consider square $\{x \mid |\mathrm{Re}(x)|, |\mathrm{Im}(x)| \leq M\}$ where M is a constant such that $M > 2 \max_{i<n}(a_i/a_n)$: all roots of $p(x)$ must lie therein

b. Recursively partition square into four congruent subsquares

c. At the center of each subsquare, use computationally cheap procedure to estimate proximity to the closest root of p

d. Eliminate subsquare if test guarantees that it contains no root

e. Use Newton-type iterations to refine solutions once they are isolated

f. Complexity: all zeros are computed with accuracy $2^{-b}M$ at overall cost $\mathcal{O}\left(n^2 \log n \log(bn)\right)$

Sum of squares reformulation

- A single-variable polynomial is nonnegative *if and only if* it can be written as a sum of squares
- A polynomial $p(x) = \sum_{i=0}^{2n} a_i x^i = 0$ is a sum of squares if and only if there exists a positive semidefinite matrix $M \in \mathbb{S}^{n+1}$ such that

$$a_k = \sum_{i+j=k+2} M_{ij}$$

- Minimizing $p(x)$ is equivalent to maximizing an additional variable t with the condition that polynomial $p(x) - t$ is nonnegative (largest lower bound)

- ◇ Minimizing a single-variable polynomial can thus be reformulated as a convex semidefinite optimization problem
- ◇ It is efficiently solvable, with a polynomial complexity ($\mathcal{O}\left(\sqrt{n}\right)$ iterations but total complexity worse than quadtree)

Note that in principle, this procedure can be used to solve a single-variable polynomial equation

Companion matrix

- Again, try to solve the stationarity condition as a monic polynomial equation

$$p(x) = x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0$$

- Fact: $p(x)$ is the characteristic polynomial of

$$A = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{pmatrix}$$

- The eigenvalues of A are thus the r roots of $p(x)$, and complexity of computing these n eigenvalues is $\mathcal{O}(n^3)$ (again worse than quadtree)

Algebraic geometry

Ideals and varieties

Let $f_1, f_2, \ldots, f_s$ polynomials in $k[x_1, x_2, \ldots, x_n]$, define the *affine variety*

$$V(f_1, f_2, \ldots, f_s) = \{(x_1, x_2, \ldots, x_n) \mid f_i(x) = 0 \; \forall i\}$$

An *ideal* is a subset of $k[x_1, x_2, \ldots, x_n]$ closed under addition and multiplication by an *element* of $k[x_1, x_2, \ldots, x_n]$

Let $f_1, f_2, \ldots, f_s$ polynomials in $k[x_1, x_2, \ldots, x_n]$, they *generate* an ideal defined by

$$\langle f_1, f_2, \ldots, f_s \rangle = \{\sum_{i=1}^{s} f_i h_i \text{ with } h_i \in k[x_1, x_2, \ldots, x_n] \; \forall i\}$$

Once again, the application of algebraic geometric techniques to polynomial optimization will rely on the resolution of the stationarity conditions as a system of polynomials equations $f_1(x) = f_2(x) = \cdots = f_s(x) = 0$
$\longrightarrow$ identify elements of the affine variety $V(f_1, f_2, \ldots, f_s)$

Key property: $\langle f_1, f_2, \ldots, f_s \rangle = \langle g_1, g_2, \ldots, g_t \rangle$ is easily seen to imply $V(f_1, f_2, \ldots, f_s) = V(g_1, g_2, \ldots, g_t)$
$\longrightarrow$ look for a generating set that allows for easy identification of the underlying affine variety

Ideals/generating sets resemble vector spaces/bases but lack the *independence/unicity* property
The use of Groebner bases is an answer to this problem

Polynomials and division

- Single variable case: divide f by g

  $\exists$ unique q and r such that $f = qg + r$ ($\deg r < \deg g$)

  easy to compute (Euclidean algorithm)

  Consequence : all ideals in $k[x]$ have the form $\langle f \rangle$

- Multiple variables: first issue is to define degree (for remainder condition)

  Define an *ordering* on the monomials such that

  a. it is a total order

  b. $x^\alpha > x^\beta \Rightarrow x^{\alpha+\gamma} > x^{\beta+\gamma}$

  c. well ordered (nonempty subsets have smallest element)

- The *multidegree* of a polynomial $f = \sum_{\alpha \in A} a_\alpha x^\alpha$ is the maximal $\alpha \in A$ such that $a_\alpha \neq 0$
- Define also the *leading monomial* of a polynomial: $\mathrm{LM}(f) = x^{\mathrm{multideg} f}$

Examples

- lexicographic order (lex)

  decide which monomial is greater by looking first at the exponent of x_1, then in case of a tie at the exponent of x_2, , then in case of another tie at the exponent of x_3, etc.

◇ graded lexicographic order (grlex)

decide which monomial is greater by looking at the sum of their exponents, and use lexicographic order in case of a tie

◇ graded reverse lexicographic order (grevlex)

decide which monomial is greater by looking at the sum of their exponents, and use reverse lexicographic order in case of a tie (which means look first at x_n, then x_{n-1}, etc. but choose the opposite of the first difference observed)

Multivariable division

Divide f by $(g_1, g_2, \ldots, g_s)$ (ordered s-uple) to obtain

$f = q_1 g_1 + q_2 g_2 + \cdots + q_s g_s + r$
with multideg $f \geq$ multideg $q_i g_i \ \forall i$
and *no monomial of* r *is divisible by any* $LM(g_i)$

This can be achieved by a naive iterative scheme (try repeatedly to divide by g_1, then by g_2, etc.) but uniqueness is *not* guaranteed

Example
divide $xy^2 - x$ by $(xy + 1, y^2 - 1)$ to find
$xy^2 - x = y(xy + 1) + 0(y^2 - 1) + (-x - y)$
while one also has $xy^2 - x = 0(xy + 1) + x(y^2 - 1) + 0$

Groebner bases

Definition: the generating set $\{g_1, g_2, \ldots, g_s\}$ of an ideal $I = \langle g_1, g_2, \ldots, g_s \rangle$ is a Groebner basis if and only if

$$\langle \mathrm{LC}(g_1), \ldots, \mathrm{LC}(g_s) \rangle = \{x^\alpha \mid x^\alpha = \mathrm{LC}(f) \text{ for some } f \in I\}$$

a. In a division by the elements of a Groebner basis, the remainder is *independent* of the order of the g_i's (but not the quotients q_i)

b. This means that Groebner bases can be used to determine whether a polynomial belongs or not to a given ideal

c. For a given monomial ordering, Groebner bases are essentially unique

d. Groebner bases can be computed with *Buchberger*'s algorithm but its complexity is exponential:

polynomials of degree bounded by d can exhibit basis elements of degree $\mathcal{O}\left(2^{2^d}\right)$

e. Computational experience tends to suggest that Groebner bases using the grevlex have much less elements than those computed using the lex or the grlex order

f. However, bad behavior can still occur:

$$\langle x^{n+1} - yz^{n-1}w, xy^{n-1} - z^n, x^n z - y^n w\rangle$$

with $x > y > z > w$ grevlex leads to $z^{n^2+1} - y^{n^2}w$ in the basis

The elimination method

Let $I = \langle f_1, f_2, \ldots, f_s \rangle \subseteq k[x_1, x_2, \ldots, x_n]$ an ideal
Define the l^{th} elimination ideal

$$I_l = I \cap k[x_{l+1}, x_{l+2}, \ldots, x_n]$$

i.e. the set of all *consequences* of $f_1, f_2, \ldots, f_s$ that do not involve the first l variables $x_1, x_2, \ldots, x_l$
If $G = \{g_i\}$ is a Groebner basis for the lex-order, $G_l = G \cap k[x_{l+1}, x_{l+2}, \ldots, x_n]$ is a Groebner basis for I_l

This means in particular that G_{n-1} involves only x_n, from which we can compute potential values for x_n. Using then G_{n-2}, one computes values for x_{n-1}, then x_{n-2} etc.
Finally check the validity of all potential solutions found

The elimination method (cont.)

- ◇ Easily applicable to the constrained case (use slacks for inequality constraints)
- ◇ Requires finite number of solutions to work
- ◇ Number of potential solutions can grow exponentially with the number of variables
- ◇ Computation of the lex-order Groebner basis can be very slow in practice

The following method will tackle this last drawback by allowing any Groebner basis to be used

The Stetter-Möller method (1993)

Let $I = \langle f_1, f_2, \ldots, f_s \rangle \subseteq k[x_1, x_2, \ldots, x_n]$

Consider the space $k[x_1, x_2, \ldots, x_n]/I$, i.e. the *quotient space* of the whole set of polynomials by the ideal: this means we do not distinguish between polynomials if they differ by a member of the ideal
One can show that this is a vector space whose dimension is *finite* if and only if the affine variety $V(I)$ is finite

Choosing now a Groebner basis for the ideal I, one can check that each class of this quotient space can be *represented* by its *remainder* when divided by the Groebner basis

The Stetter-Möller method (cont.)

Using a basis of $k[x_1, x_2, \ldots, x_n]/I$ (which can be easily done by inspection of the Groebner basis), one can write down A_i, the matrix corresponding to the linear operator describing multiplication by x_i in the quotient space $k[x_1, x_2, \ldots, x_n]/I$

One has then the following key result:

The eigenvalues of A_i correspond to the values that variable x_i takes on the affine variety $V(I)$

This allows to readily compute all solutions to the polynomial system of equations

The Stetter-Möller method (cont.)

- Easily applicable to the constrained case (use slacks for inequality constraints)
- Requires finite number of solutions to work
- Number of potential solutions can grow exponentially with the number of variables
- Key step is identification of a basis of $k[x_1, x_2, \ldots, x_n]/I$, which requires the computation of *any* Groebner basis (this can still be difficult)

The next method will try to alleviate the need for the explicit computation of any type of Groebner basis

The Hanzon-Jibetean method (2001)

Instead of minimizing $p(x_1, x_2, \ldots, x_n)$, consider

$$q_\lambda(x_1, x_2, \ldots, x_n) = p(x_1, x_2, \ldots, x_n) + \lambda(x_1^{2m} + x_2^{2m} + \ldots + x_n^{2m})$$

where λ is a parameter and m an integer such that $2m$ is greater than the total degree of p

We have

$$\inf_{x \in \mathbb{R}^n} p(x) = \lim_{\lambda \to 0} \min_{x \in \mathbb{R}^n} q_\lambda(x)$$

$\longrightarrow$ minimize the (parameterized) polynomial $q(x)$

The Hanzon-Jibetean method (cont.)

Key result: the polynomials arising from the stationarity conditions of q_λ *always constitute a Groebner basis* for the ideal they generate

$\longrightarrow$ implicit Groebner basis $\longrightarrow$ Stetter-Möller method

- Limit of the eigenvalues $\lambda \to 0$ can be evaluated using a power series technique
- Not applicable to the constrained case! (breaks implicit Groebner basis)
- Allows for infinite number of solutions (finds minimum-norm solution one in each connected component)!

Resultant methods

◇ Resultant methods are bases on the fact that a sufficient condition for that fact that two polynomials in $k[x]$ share a common root is the vanishing of the determinant of a certain matrix called the Sylvester matrix (easy to build from the polynomial coefficients)

◇ This result can be used to solve two polynomial equations involving two variables by considering one of the variables to be a parameter and solving the polynomial equation obtained by equating the resulting Sylvester determinant to zero

◇ This can be applied recursively to solve any multivariable equation

Sum of square methods

- In the multivariable case, sum of squares methods only provide relaxations since there exists nonnegative polynomials than cannot be expressed as a sum of squares: $x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2$ (Motzkin)
- Use some more theory, it is however possible to build a sequence of semidefinite relaxations of increasing size whose optima converge to the global optimum
- These methods also provide a proof for the lower bound they compute
- Practical computational efficiency (using the first few relaxations) seems promising (GloptiPoly, D. Henrion)

Concluding remarks

- Many different methods for the single variable case. Weyl's quadtree seems the best complexity-wise but the fact that it beats the convex reformulation is surprising
- Why is it more efficient to concentrate on the stationarity conditions rather than optimizing directly ?
- Groebner bases techniques are powerful but their worst-case complexity is awful and practical performance sometimes too slow
- Sum of squares relaxation vs. Hanzon & Jibetean comparison wanted !