EXPLOITING SPARSITY AND SYMMETRY IN SEMIDEFINITE PROGRAMMING WITH APPLICATIONS IN POLYNOMIAL OPTIMIZATION

Kartik Sivaramakrishnan
Department of Mathematics
North Carolina State University
kksivara@ncsu.edu
http://www4.ncsu.edu/∼kksivara

Symbolic Computation Seminar
North Carolina State University
November 29, 2006

Contents

Semidefinite programming

$$\begin{array}{lll} \text{(SDP)} & \max & C \bullet X \\ & \text{s.t.} & \mathcal{A}(X) = b \\ & & X \succeq 0 \end{array} \qquad \begin{array}{lll} \text{(SDD)} & \min & b^T y \\ & \text{s.t.} & \mathcal{A}^T y - S = C \\ & & S \succeq 0 \end{array}$$

- **Notation**
  - $X, S, C \in \mathcal{S}^n$, $b \in \mathbb{R}^m$
  - $A \bullet B = \text{trace}(AB) = \sum_{i,j=1}^n A_{ij} B_{ij}$ (Frobenius inner product)
  - The operator $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ and its adjoint $\mathcal{A}^T : \mathbb{R}^m \to \mathcal{S}^n$ are

$$\mathcal{A}(X) = \begin{pmatrix} A_1 \bullet X \\ \vdots \\ A_m \bullet X \end{pmatrix}, \quad \mathcal{A}^\top y = \sum_{i=1}^m y_i A_i$$

where $A_i \in \mathcal{S}^n$, $i = 1, \ldots, m$

Polynomial preliminaries 1

1. $p(x) = \sum_{\alpha \in Z_+^n} p_\alpha x_1^{\alpha_1} \dots x_n^{\alpha_n} = \sum_{\alpha \in Z_+^n} p_\alpha x^\alpha$ is a multivariate polynomial.
2. The degree of the monomial $x^\alpha = x_1^{\alpha_1} \dots x_n^{\alpha_n}$ is $|\alpha| = \sum_{i=1}^{n} \alpha_i$.
3. The degree of the polynomial $p(x)$ is the maximum degree of a monomial x^α for which $p_\alpha \neq 0$.
4. **Example:** $x_1^2 x_2^3 x_3 - x_2^4 + 2x_1x_3 - 1$ is a polynomial in 3 variables with degree 6.
5. Let $S_n^d \;:=\; \{\alpha \in Z_+^n : \;|\alpha| \leq d\}, \;\; |S_n^d| = \binom{n+d}{d}$.
6. One can identify a polynomial $p(x)$ of degree d with its sequence of coefficients $p = (p_\alpha)_{\alpha \in S_n^d}$.
7. The set $\mathbb{R}_d[x_1, \dots, x_n]$ of polynomials of degree $\leq d$ with real coefficients is isomorphic to $\mathbb{R}^{\binom{n+d}{d}}$.

Polynomial preliminaries 2

1. A polynomial $p(x)$ is nonnegative if $p(x) \geq 0$, $\forall x \in \mathbb{R}^n$.
   Let $\mathcal{P}_{n,d} = \{p_\alpha : \ p(x) = \sum_\alpha p_\alpha x^\alpha \geq 0, \forall x \in \mathbb{R}^n$, deg$(p(x)) \leq d\}$.
2. A polynomial $p(x)$ has a sum of squares (SOS) if

$$p(x) \ = \ \sum_i p_i(x)^2.$$

   for some polynomials p_i. It is clear that $p(x)$ has even degree $2d$, and each $p_i(x)$ has degree $\leq d$. Let
   $\Sigma_{n,d} = \{p_\alpha : \ p(x) = \sum_\alpha p_\alpha x^\alpha$ has an SOS, deg$(p(x)) \leq d\}$.
3. **Example:** $x_1^2 + x_2^2 + 2x_1x_2 + x_3^6 = (x_1 + x_2)^2 + (x_3^3)^2$ is an SOS.
4. The sets $\mathcal{P}_{n,d}$ and $\Sigma_{n,d}$ are both closed convex cones.

When does a nonnegative polynomial have an SOS decomposition?

1. An SOS polynomial is nonnegative but not every nonnegative polynomial is an SOS.
2. Hilbert in 1888 showed that $\mathcal{P}_{n,d} = \Sigma_{n,d}$, i.e., every nonnegative polynomial of degree d in n variables is an SOS when
   - $n = 1$ ($d \geq 2$ and even)
   - $d = 2$ (all n)
   - $n = 2, d = 4$

   In all other cases, $\Sigma_{n,d} \subset \mathcal{P}_{n,d}$ (see Chapter 6 in *Squares* by Rajwade).
3. The Motzkin polynomial $x_1^4 x_2^2 + x_1^2 x_2^4 - 3x_1^2 x_2^2 + 1$ is nonnegative but *not* a SOS.

Applications of semidefinite programming in polynomial optimization

1. Given a polynomial $p(x)$ in n variables and even degree $2d$.

$$\begin{aligned} p_{\min} &= \inf_{x \in \mathbb{R}^n} p(x) \\ &= \sup\{t : \ p(x) - t \geq 0 \ \forall x \in \mathbb{R}^n\}. \end{aligned}$$

   The constraint requires $p(x) - t \in \mathcal{P}_{n,2d}$.

2. A constrained version of this problem is

$$p_{sos} = \sup\{t : \ p(x) - t \text{ is SOS}\}.$$

   Since $p(x) - t \in \Sigma_{n,2d} \subseteq \mathcal{P}_{n,2d}$, we have $p_{\min} \geq p_{sos}$.

3. The second problem can be reformulated as a semidefinite programming problem in standard form where the size of the matrix size is $\binom{n+d}{d}$ and $m = \binom{n+2d}{2d}$ (Parrilo, Lasserre).

4. One can generate a hierarchy of SDPs (of increasing size) whose objective values converge to the global minimum value of the polynomial.

Difficulties with solving large SDPs

1. Interior point methods (IPMs) can solve a semidefinite program to a prescribed accuracy in a polynomial number of arithmetic operations (Nesterov and Nemirovskii).
2. Excellent IPM software for solving semidefinite programs are currently available (CSDP, SeDuMi, SDPT3, SDPA).
3. In each iteration of an IPM, one has to
   (a) Compute the Schur matrix of size m; this requires $O(mn^3 + m^2n^2)$ flops.
   (b) Factorize and store this dense matrix; a Cholesky factorization requires $O(m^3)$ flops.
4. Limited to SDPs with matrix size $n = 5000$ and $m = 10,000$.
5. Kartik is currently exploring decomposition techniques to solve large scale block-angular semidefinite programs in a parallel and distributed computing environment.

Semidefinite programming with block angular structure

$$\begin{array}{lrcl} \max & \sum_{i=1}^r C_i \bullet X_i & & \\ \text{s.t.} & \sum_{i=1}^r \mathcal{A}_i(X_i) & = & b \\ & X_i & \in & \mathcal{C}_i, \ \ i = 1, \ldots, r \end{array}$$

- **Notes**
  - $X_i, C_i \in \mathcal{S}^{n_i}$, $b \in \mathbb{R}^m$.
  - $\mathcal{C}_i = \{X_i : \mathcal{B}_i(X_i) = d_i, X_i \succeq 0\}$ are compact semidefinite feasibility sets (described by LMIs).
  - The objective function and coupling constraints are **block separable**.
  - If the coupling constraints were absent, then we only have to solve r independent problems of the form

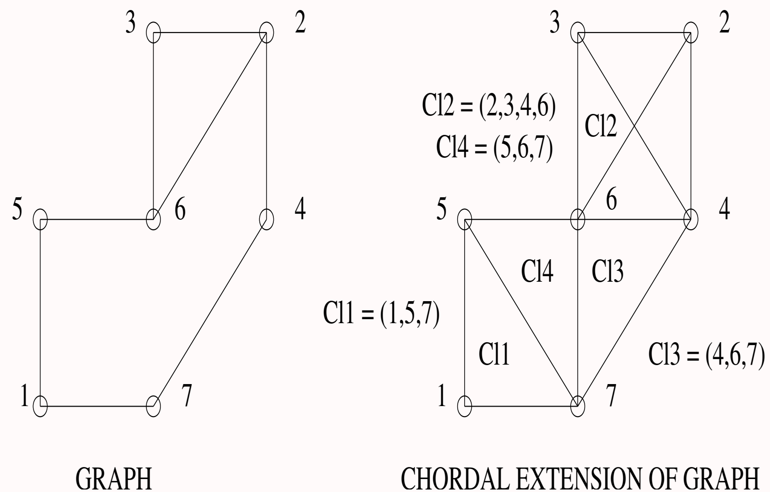
$$\max_{X_i \in \mathcal{C}_i} \ C_i \bullet X_i$$

Exploiting sparsity to get block-angular SDPs: 1

Consider the SDP

$$\begin{array}{lrcl} \max & L \bullet X & & \\ \text{s.t.} & X_{ii} & = & 1, \;\; i = 1, \ldots, n, \\ & X & \succeq & 0, \end{array}$$

where L is the adjacency matrix of the graph



Exploiting sparsity to get block-angular SDPs: 2

Using **matrix completion**, one can reformulate the earlier SDP as

$$
\begin{array}{lrcl}
\max & \multicolumn{3}{l}{\sum_{k=1}^{4} (L^k \bullet X^k)} \\
\text{s.t.} & X^1_{23} - X^4_{13} & = & 0, \\
 & X^2_{34} - X^3_{12} & = & 0, \\
 & X^3_{23} - X^4_{23} & = & 0, \\
 & X^k_{ii} & = & 1, \;\; i = 1, \ldots, |C_k|, \;\; k = 1, \ldots, 4, \\
 & X^k & \succeq & 0, \;\; k = 1, \ldots, 4,
\end{array}
$$

which is in **block-angular** form.

Exploiting sparsity to get block-angular SDPs: 3

1. Construct the aggregate sparsity graph $G = (V, E)$ from data matrices $A_0 = C$ and A_i, $i = 1, \dots, m$ of SDP. We have $V = \{1, \dots, n\}$ and
$$E \;=\; \{(i,j) \in V \times V : \exists k \in \{0, 1, \dots, m\} \text{ s.t. } (A_k)_{ij} \neq 0\}$$
2. Construct a minimal **chordal** extension $G' = (V, E')$ of $G = (V, E)$.
3. Find maximal cliques Cl_i, $i = 1, \dots, k$ in $G' = (V, E')$. We have
$$X \succeq 0 \;\Leftrightarrow\; X_{Cl_i, Cl_i} \succeq 0, \;\; i = 1, \dots, k$$
4. Each block in the block-diagonal SDP corresponds to a maximal clique.
5. Introduce additional equality constraints for common nodes and edges in the cliques ; for instance if cliques Cl_k and Cl_l share a common edge $\{i, j\}$ then $X^k_{ij} = X^l_{ij}$ etc.
6. In some special cases, the resulting block-diagonal SDP has a block-angular form.

Exploiting symmetry to get block-angular SDPs: 1

1. We are given a finite group G, elements $g \in G$ and its representation $\rho(g) : G \rightarrow GL(\mathcal{R}^n)$ (the set of real invertible matrices of size n). Also, assume $\rho(g)$ to be an orthogonal matrix.

2. Consider the associated representation $\sigma(g)$ of G given by $\sigma(g)X = \rho(g)^T X \rho(g)$ where X is a symmetric psd matrix of size n (congruence transformation on X).

3. We say that a SDP is *invariant* w.r.t group action $\sigma(g)$ if

   (a) If X is feasible in the SDP then $\sigma(g)X$ is also feasible $\forall g \in G$.

   (b) We have $C \bullet X = C \bullet \sigma(g)X$, $\forall g \in G$.

4. In this case, one can add the additional linear constraints $X = \sigma(g)X$, i.e., $\rho(g)X - X\rho(g) = 0$, $\forall g \in G$ (fixed point subspace) to the SDP without changing its optimal value (Gatermann-Parrilo).

Exploiting symmetry: 2

1. Consider the SDP

$$\begin{array}{lrcl} \min & -X_{11} + X_{22} + X_{33} & & \\ \text{s.t.} & X_{12} & = & X_{13} \\ & \text{trace}(X) & = & 1 \\ & X & \succeq & 0. \end{array}$$

2. SDP unchanged under a simultaneous permutation of the last two rows and columns of X, i.e., invariant under the permutation group S_2.

3. Add the additional constraint $X_{22} - X_{33} = 0$ (fixed point subspace) to the SDP without changing the optimal objective value.

4. Apply a symmetry-adapted orthogonal transformation (using irreducible representations of S_2) to get an equivalent block-angular SDP

$$\begin{array}{lrcl} \min & -X^1_{11} + X^1_{22} + X^2 & & \\ \text{s.t.} & \text{trace}(X^1) + X^2 & = & 1 \\ & X^1 & \succeq & 0 \\ & X^2 & \geq & 0 \end{array}$$

containing a semidefinite block of size 2 and a linear block of size 1.

The Lagrangian dual problem

- The Lagrangian dual problem is

$$\min_y \;\; \theta(y) = b^T y + \textstyle\sum_{i=1}^r \theta_i(y)$$

  where

$$\theta_i(y) \;\; = \;\; \max_{X_i \in \mathcal{C}_i} \;\; (C_i - \mathcal{A}_i^T y) \bullet X_i$$

- Dual is an unconstrained **convex** but **nonsmooth** problem.
- Given y^k, we have $\theta(y^k) = b^T y^k + \sum_{i=1}^r (C_i - \mathcal{A}_i^T y^k) \bullet X_i^k$

  and a subgradient $g(y^k) = (b - \sum_{i=1}^r \mathcal{A}_i(X_i^k))$ where

$$X_i^k \;\; = \;\; \operatorname*{argmax}_{X_i \in \mathcal{C}_i} \;\; (C - \mathcal{A}_i^T y^k) \bullet X_i$$

  (these can be computed in parallel!)

Solving the Lagrangian dual

1. Construct a model $\theta^k(y)$ an **underestimate** for $\theta(y)$

$$\theta^k(y) \; = \; b^T y + \textstyle\sum_{i=1}^r \max\limits_{j=1,\ldots,J^k(i)} (C_i - \mathcal{A}_i^T y) \bullet X_i^j$$

   from the function values and subgradient information.

2. The **regularized** master problem then is

$$\min_y \;\; \theta^k(y) + \tfrac{u^k}{2} ||y - x^k||^2$$

   where $u^k \geq 0$ and x^k is our current center (best iterate so far!)

3. The dual to this quadratic program (much easier to solve!) is

$$\max \;\; -\frac{1}{2u^k} || \sum_{i=1}^{r} \sum_{j=1}^{J^k(i)} \mathcal{A}_i(X_i^j)\lambda_i^j - b||^2 + \sum_{i=1}^{r} \sum_{j=1}^{J^k(i)} ((C_i - \mathcal{A}_i^T x^k) \bullet X_i^j)\lambda_i^j$$

$$\text{s.t.} \quad \sum_{j=1}^{J^k(i)} \lambda_i^j \; = \; 1, \;\; i = 1, \ldots, r$$

$$\lambda_i^j \; \geq \; 0, \;\; i = 1, \ldots, r, \;\; j = 1, \ldots, J^k(i).$$

Kartik's approach in a nutshell

(a) Solve **large scale structured semidefinite programs (SDP)** arising in polynomial optimization. Exploit the sparsity and the symmetry in these SDPs (arising from the underlying polynomial!)

(b) The technique is to **iteratively** solve an SDP between a **mixed conic master problem** over linear and smaller semidefinite cones; and **distributed subproblems** (smaller SDPs) in a parallel high performance computing environment.

(c) Improve the **scalability** of interior point methods (IPMs) by applying them instead on the smaller master problem, and subproblems (which are solved in parallel!)

This is our **conic interior point decomposition** scheme.

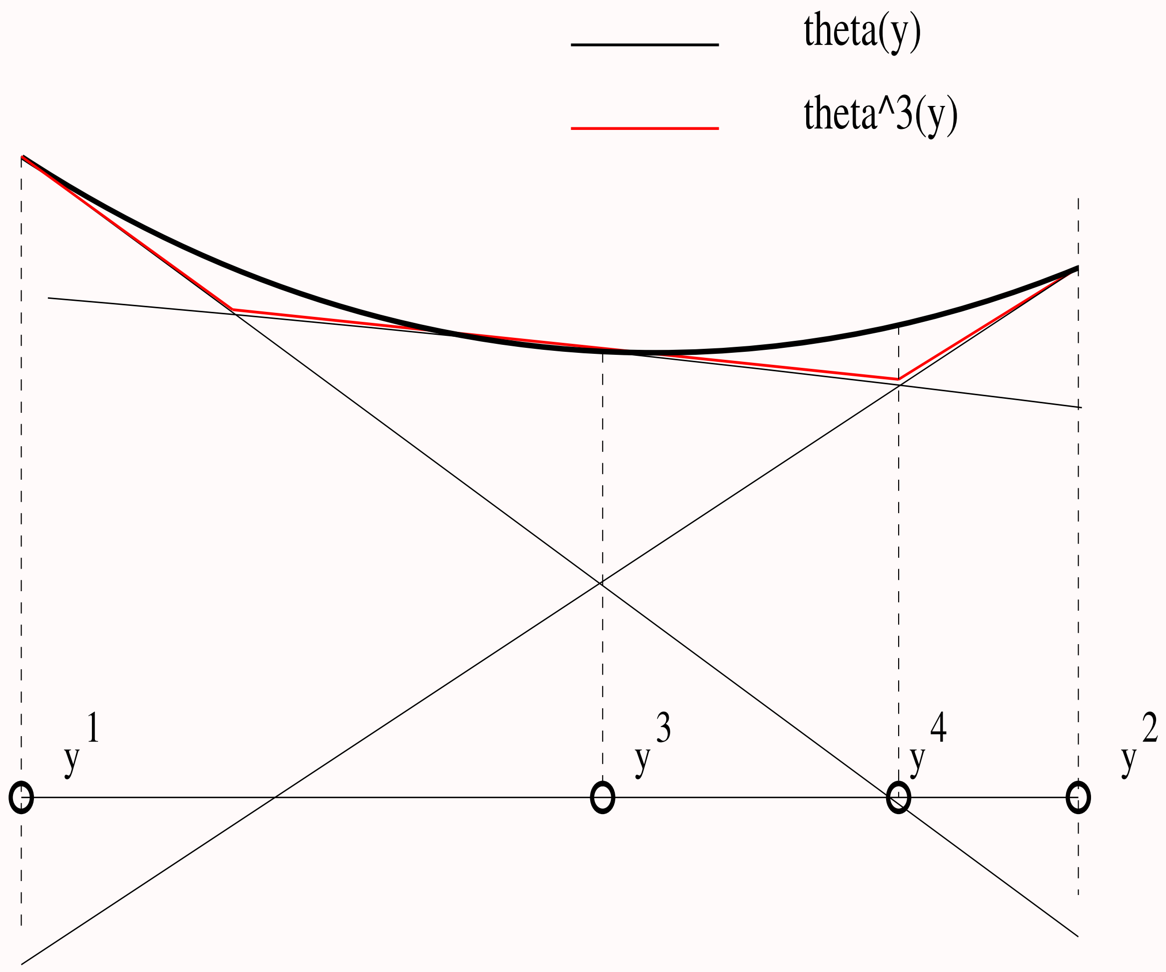


Figure 1: Solving the Lagrangian dual problem

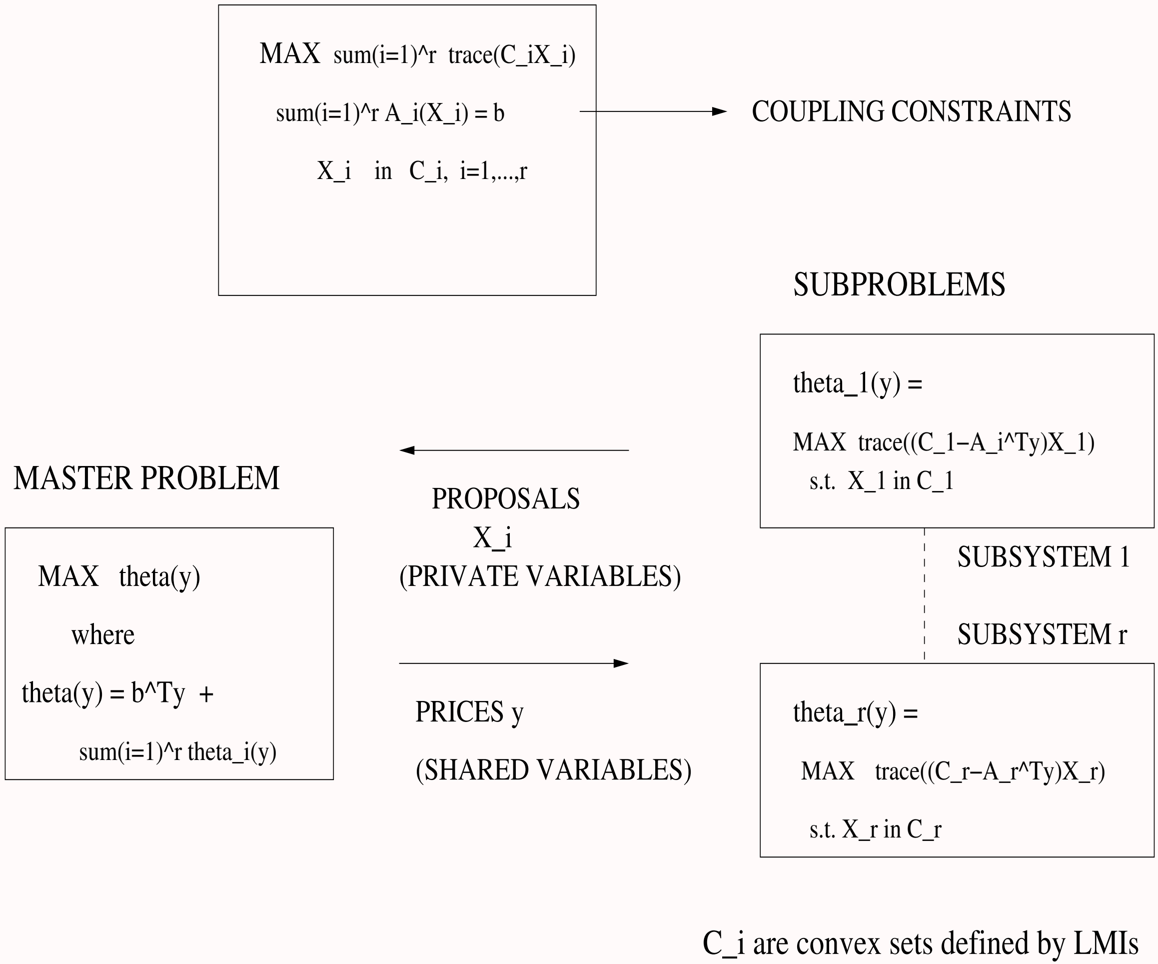


Figure 2: Decomposition by prices

Thank you for your attention!.

Questions, Comments, Suggestions ?

The slides from this talk are available online at

http://www4.ncsu.edu/~kksivara/publications/kartik-symb.pdf

My publications and talks can be downloaded from

http://www4.ncsu.edu/~kksivara/publications

Bibliography: 1

1. K. Sivaramakrishnan, *A parallel conic interior point decomposition for block angular semidefinite programs*, submitted.
http://www4.ncsu.edu/∼kksivara/publications/parallel-conic-blockangular.pdf

2. K. Sivaramakrishnan, G. Plaza, and T. Terlaky, *A conic interior point decomposition approach for large scale semidefinite programming*, submitted.
http://www4.ncsu.edu/∼kksivara/publications/conic-ipm-decomposition.pdf

3. A. Ruszczyński, *Nonlinear Optimization*, Princeton University Press, 2006.

4. M. Fukuda, M. Kojima, K. Murota, and K. Nakata, *Exploiting sparsity in semidefinite programming via matrix completion I: General framework*, SIAM Journal on Optimization, 11(2000), pp. 647-674.

6. K. Fujisawa, M. Fukuda, and K. Nakata, *Preprocessing sparse semidefinite programs via matrix completion*, Optimization Methods and Software, 21(2006), pp. 17-39.

7. P.A. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Mathematical Programming, 96(2003), pp. 293-320.

8. K. Gatermann and P.A. Parrilo, *Symmetry groups, semidefinite programs, and sums of squares*, Journal of Pure and Applied Algebra 192(2004), pp. 95-128.

Bibliography: 3

9. P.A. PARRILO, *Exploiting algebraic structure in sums of squares programs*, In Positive Polynomials in Control, LNCSI 312, Springer, 2005, pp. 181-194.

10. E. DE KLERK, *Exploiting algebraic symmetry in semidefinite programming: a review*, Talk at Delft, the Netherlands, October, 2005.
`http://www4.ncsu.edu/~kksivara/etienne-symmetry.pdf`

11. H. WAKI, S. KIM, M. KOJIMA, AND M. MURAMATSU, *Sums of squares and semidefinite programming relaxations for polynomial problems with structured sparsity*.
`http://www.optimization-online.org/DB_HTML/2004/10/988.html`

12. J.B. LASSERRE, *Global optimization with polynomials and the problem of moments*, SIAM Journal on Optimization, 11(2001), pp. 796-817.

13. A. RAJWADE, *Squares*, Volume 171 of London Math Society Lecture Note Series, Cambridge University Press, 1993.

14. S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press 2004.

15. K. SIVARAMAKRISHNAN, *Solving polynomial optimization problems via real algebraic geometry and semidefinite programming*, Talk in the Symbolic Computation seminar, NCSU, January 2006.
`http://www4.ncsu.edu/~kksivara/publications/ncsu-symb.pdf`.