SDP relaxation method for detecting P-tensors

Li Li* and Xiao Wang

Department of Mathematics, Tianjin University, China.

Abstract   P-tensor and P_0-tensor are introduced in tensor complementarity problem, which have wide applications in many fields such as game theory, tensor complementarity problem. In this paper, we discuss how to check whether a given symmetric tensor is $P(P_0)$-tensor or not. For a symmetric tensor, it is a $P(P_0)$-tensor is equivalent to the positivity(nonnegativity) of a polynomial optimization problem. For such polynomial optimization problem, a SDP relaxation method is proposed. By the proposed method, the $P(P_0)$-tensor can be detected by solving a finite number of SDP relaxations. Furthermore, numerical examples are reported to show the efficiency of the proposed algorithm.

Keywords   P-tensor, P_0-tensor, SDP relaxation

AMS 2010 subject classifications 15A18, 90C22, 90C30

DOI: 10.19139/soic.v5i4.324

1. Introduction

For positive integers m and $n_1, n_2, \cdots, n_m$, an m-order and $(n_1, n_2, \cdots, n_m)$-dimensional real tensor is an array in the space $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$. Every tensor $\mathcal{A}$ from this space can be indexed as

$$\mathcal{A} = (a_{i_1 i_2 \cdots i_m}), \ \ 1 \le i_j \le n_j, \ \ j = 1, 2, \cdots m. \tag{1}$$

When $n_1 = \cdots = n_m = n$, $\mathcal{A}$ is called an m-order n-dimensional square tensor. In such case, the tensor space $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$ is denoted as $\mathtt{T}^m(\mathbb{R}^n)$. A tensor in $\mathtt{T}^m(\mathbb{R}^n)$ is said to be *symmetric* if its entries are invariant under permutations of indices $(i_1, i_2, \ldots, i_m)$. The subspace of symmetric tensors in $\mathtt{T}^m(\mathbb{R}^n)$ is denoted as $\mathtt{S}^m(\mathbb{R}^n)$. Using the notation as in Qi [16], for $\mathcal{A} \in \mathtt{T}^m(\mathbb{R}^n)$ and $x := (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, we denote

$$\begin{cases} \mathcal{A}x^m & := \displaystyle\sum_{1 \le i_1, \cdots, i_m \le n} a_{i_1 i_2 \cdots i_m} x_{i_1} x_{i_2} \cdots x_{i_m}, \\ \mathcal{A}x^{m-1} & := \left(\displaystyle\sum_{1 \le i_2, \cdots, i_m \le n} a_{j i_2 \cdots i_m} x_{i_2} \cdots x_{i_m} \right)_{j=1,\ldots,n}. \end{cases} \tag{2}$$

Note that $\mathcal{A}x^{m-1} \in \mathbb{R}^n$. Denote $[p] = \{1, 2, \ldots, p\}$ for every positive integer p in this paper. With these notations, the definition of $P(P_0)$-tensor is presented, introduced in [18].

Definition 1
For $\mathcal{A} \in \mathtt{T}^m(\mathbb{R}^n)$, $\mathcal{A}$ is said to be a

---

*Correspondence to: Li Li (Email: lili2015@tju.edu.cn). Department of Mathematics, Tianjin University. 135 Yaguan Road, Tianjin, China (300354).

(i) P_0-tensor if and only if for any nonzero vector $x \in \mathbb{R}^n$, there exists $i \in [n]$ such that $x_i \neq 0$ and

$$x_i(\mathcal{A}x^{m-1})_i \geq 0; \tag{3}$$

(ii) P-tensor if and only if for any nonzero vector $x \in \mathbb{R}^n$,

$$\max_{i \in [n]} x_i(\mathcal{A}x^{m-1})_i > 0. \tag{4}$$

This definition can be regarded as an extension of $P(P_0)$-matrix, which plays important roles in linear complementarity problems and variational inequalities, see [4, 6, 14].

For symmetric $P(P_0)$-tensors, some properties of tensor are presented in [18]. It is shown that a given symmetric tensor is $P(P_0)$-tensor if and only if its smallest Z(H)-eigenvalue is positive(nonnegative). Furthermore, there does not exist odd order P-tensor and nonzero P_0-tensor. For even order symmetric tensor, it is a $P(P_0)$-tensor if and only if it is positive (semi-)definite. For computing the smallest Z(H)-eigenvalue, Qi et al. [17] discuss the case of $(m, n) = (3, 2)$. Later shifted power methods are proposed in [8, 19]. Recently, SDP relaxation method is applied to compute Z(H)-eigenvalues in [1].

Motivated by these fact, in this paper, we propose a numerical method by solving SDP relaxations to check whether the given symmetric tensor is $P(P_0)$-tensor or not. Furthermore, it is a P-tensor if the minimum of SDP relaxation is positive; it is a P_0-tensor but not P-tensor if the optimal value of the polynomial optimization problem is zero; it is not a P_0-tensor if the optimal value of the polynomial optimization problem is negative.

This paper is organized as follows. Section 2 gives preliminaries on polynomial optimization. Section 3 presents SDP relaxation method of polynomial optimization problem to check symmetric $P(P_0)$-tensor. Numerical examples are presented in Section 4.

2. Preliminaries

In this section, we review some basics in polynomial optimization. We refer to [10, 11] for surveys in polynomial optimization.

In the space $\mathbb{R}^n$, the symbol $\| \cdot \|$ denotes the standard Euclidean norm. Let $\mathbb{R}[x]$ be the ring of polynomials with real coefficients and in variables $x := (x_1, \ldots, x_n)$, and let $\mathbb{R}[x]_d$ be the set of real polynomials in x whose degrees are at most d.

For a polynomial tuple $h = (h_1, h_2, \cdots, h_s)$, the ideal generated by h is the set

$$I(h) := h_1 \cdot \mathbb{R}[x] + h_2 \cdot \mathbb{R}[x] + \cdots + h_s \cdot \mathbb{R}[x].$$

The k-th truncation of $I(h)$ is the set

$$I_k(h) := h_1 \cdot \mathbb{R}[x]_{k-deg(h_1)} + \cdots + h_s \cdot \mathbb{R}[x]_{k-deg(h_s)}.$$

The complex and real algebraic varieties of h are respectively defined as

$$\mathcal{V}_{\mathbb{C}}(h) := \{x \in \mathbb{C}^n \mid h(x) = 0\}, \quad \mathcal{V}_{\mathbb{R}}(h) := \mathcal{V}_{\mathbb{C}}(h) \cap \mathbb{R}^n.$$

A polynomial p is said to be sum of squares (SOS) if there exist $p_1, p_2, \cdots p_r \in \mathbb{R}[x]$ such that $p = p_1^2 + p_2^2 + \cdots + p_r^2$. The set of all SOS polynomials is denoted as $\Sigma[x]$. For a given degree m, denote

$$\Sigma[x]_m := \Sigma[x] \cap \mathbb{R}[x]_m.$$

The quadratic module generated by a polynomial tupe $g = (g_1, \cdots, g_t)$ is the set

$$Q(g) := \Sigma[x] + g_1 \cdot \Sigma[x] + \cdots + g_t \cdot \Sigma[x].$$

The k-th truncation of the quadratic module $Q(g)$ is the set

$$Q_k(g) := \Sigma[x]_{2k} + g_1 \cdot \Sigma[x]_{2k-deg(g_1)} + \cdots + g_t \cdot \Sigma[x]_{2k-deg(g_t)}.$$

Note that if $g = \emptyset$ is an empty tuple, then $Q(g) = \Sigma[x]$ and $Q_k(g) = \Sigma[x]_{2k}$.

Let $\mathbb{N}$ be the set of nonnegative integers. For $x := (x_1, \ldots, x_n)$, $\alpha := (\alpha_1, \ldots, \alpha_n)$ and a degree d, denote

$$x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}, \quad |\alpha| := \alpha_1 + \cdots + \alpha_n, \quad \mathbb{N}_d^n := \{\alpha \in \mathbb{N}^n : |\alpha| \le d\}.$$

Denote by $\mathbb{R}^{\mathbb{N}_d^n}$ the space of all real vectors y that are indexed by $\alpha \in \mathbb{N}_d^n$. For $y \in \mathbb{R}^{\mathbb{N}_d^n}$, we can write it as

$$y = (y_\alpha), \quad \alpha \in \mathbb{N}_d^n.$$

For $f = \sum_{\alpha \in \mathbb{N}_d^n} f_\alpha x^\alpha \in \mathbb{R}[x]_d$ and $y \in \mathbb{R}^{\mathbb{N}_d^n}$, we define the operation

$$\langle f, y \rangle := \sum_{\alpha \in \mathbb{N}_d^n} f_\alpha y_\alpha. \tag{5}$$

For an integer $t \le d$ and $y \in \mathbb{R}^{\mathbb{N}_d^n}$, denote the t-th truncation of y as

$$y|_t := (y_\alpha)_{\alpha \in \mathbb{N}_t^n}. \tag{6}$$

Let $q \in \mathbb{R}[x]$ with $\deg(q) \le 2k$. For each $y \in \mathbb{R}\mathbb{N}_{2k}^n$, $\langle qp^2, y \rangle$ is a quadratic form in $vec(p)$, the coefficient vector of the polynomial p with $\deg(qp^2) \le 2k$. Let $L_q^{(k)}(y)$ be the symmetric matrix such that

$$\langle qp^2, y \rangle = vec(p)^T \left(L_q^{(k)}(y) \right) vec(p). \tag{7}$$

The matrix $L_q^{(k)}(y)$ is called the k-th localizing matrix of q generated by y. It is linear in y. For instance, when $n = 2$, $k = 2$ and $q = x_1 x_2 - x_1^2 - x_2^2$,

$$L_{x_1 x_2 - x_1^2 - x_2^2}^{(2)}(y) = \begin{pmatrix} y_{11} - y_{20} - y_{02} & y_{21} - y_{30} - y_{12} & y_{12} - y_{21} - y_{03} \\ y_{21} - y_{30} - y_{12} & y_{31} - y_{40} - y_{22} & y_{22} - y_{31} - y_{13} \\ y_{12} - y_{21} - y_{03} & y_{22} - y_{31} - y_{13} & y_{13} - y_{22} - y_{04} \end{pmatrix}.$$

If $q = (q_1, \ldots, q_r)$ is a tuple of polynomials, we then define

$$L_q^{(k)}(y) := \left(L_{q_1}^{(k)}(y), \ldots, L_{q_r}^{(k)}(y) \right). \tag{8}$$

When $q = 1$ (the constant 1 polynomial), $L_1^{(k)}(y)$ is called the k-th moment matrix generated by y, and we denote

$$M_k(y) := L_1^{(k)}(y). \tag{9}$$

For instance, when $n = 2$ and $k = 2$,

$$M_2(y) = \begin{pmatrix} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{pmatrix}.$$

For a degree d, denote the monomial vector

$$[x]_d := \begin{bmatrix} 1 & x_1 & \cdots & x_n & x_1^2 & x_1 x_2 & \cdots & x_n^2 & \cdots & x_1^m & \cdots & x_n^m \end{bmatrix}^T. \tag{10}$$

3. Checking symmetric P(P$_0$)-tensor

In this section, we propose a numerical method to check whether a given symmetric tensor is P(P$_0$)-tensor or not. From [18], for a symmetric tensor, it is a P(P$_0$)-tensor if and only if it is positive (semi-)definite. Furthermore, the positive (semi-)definiteness is equivalent to positivity(nonnegativity) of the smallest Z(H)-eigenvalue. Motivated by this, we propose a numerical method to check the P(P$_0$)-tensor. Before proceeding, we recall the definition of Z(H)-eigenvalues.

Definition 2
For $\mathcal{A} \in \mathrm{T}^m(\mathbb{R}^n)$, a number $\lambda \in \mathbb{R}$ is called a Z-eigenvalue of $\mathcal{A}$ if there exists a vector $u \in \mathbb{R}^n$ such that

$$\mathcal{A}u^{m-1} = \lambda u, \quad u^T u = 1. \tag{11}$$

(The superscript ^{T} denotes the transpose.) Such u is called a Z-eigenvector associated with λ, and (λ, u) is called a Z-eigenpair. Furthermore, a number $\alpha \in \mathbb{R}$ is called an H-eigenvalue of $\mathcal{A}$ if there exists $0 \neq v \in \mathbb{R}^n$ such that

$$\mathcal{A}v^{m-1} = \alpha v^{[m-1]}. \tag{12}$$

(The symbol $v^{[m-1]}$ denotes the vector such that $(v^{[m-1]})_i = (v_i)^{m-1}$ for $i = 1, \ldots, n$). Such v is called an H-eigenvector associated with α, and (α, v) is called an H-eigenpair.

From [13], the smallest Z(H)-eigenvalue is the optimal value of the following optimization problem

$$\begin{aligned} f^s := \min \quad & \mathcal{A}x^m \\ \text{s.t.} \quad & x^\top x^{[m']} = 1, \end{aligned} \tag{13}$$

where $x^{[k]} = (x_i^k) \in \mathbb{R}^n$. f^s is the smallest Z-eigenvalue when $m' = 1$, and f^s is the smallest H-eigenvalue when $m' = m - 1$. The computation of (13) for $m' = 1$ and $m' = m - 1$ are similar, hence we only adopt $m' = 1$ for cleanness in the following.

The first order optimality condition of (13) with $m' = 1$ can be written as following

$$\left\{ \begin{array}{l} m\mathcal{A}x^{m-1} - \lambda x = 0, \\ x^\top x = 1, \end{array} \right.$$

for some $\lambda \in \mathbb{R}$. It is clear to see that $\lambda = m\mathcal{A}x^m$. Based on this observation, we consider the following optimization problem

$$\begin{aligned} \min \quad & \mathcal{A}x^m \\ \text{s.t.} \quad & \mathcal{A}x^{m-1} = (\mathcal{A}x^m)x, \\ & x^\top x = 1. \end{aligned} \tag{14}$$

It is clear to see that problems (13) and (14) are equivalent, that is, they have the same optimal solution. Hence, it suffices to consider problem (14). For convenience, we introduce the following notations

$$\left\{ \begin{array}{ll} f(x) & := \mathcal{A}x^m, \\ h(x) & := \{x^\top x - 1, \ \mathcal{A}x^{m-1} - (\mathcal{A}x^m)x\}. \end{array} \right. \tag{15}$$

Now problem (14) can be rewritten as

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0. \end{aligned} \tag{16}$$

Lasserre's hierarchy ([9]) of semidefinite relaxations for problem (16) is

$$\begin{aligned} \rho_k^{(1)} := \quad \min \quad & \langle f, y \rangle \\ \text{s.t.} \quad & L_h^{(k)}(y) = 0, \\ & \langle 1, y \rangle = 1, \ M_k(y) \succeq 0, \ y \in \mathbb{R}^{\mathbb{N}_{2k}^n}, \end{aligned} \tag{17}$$

where $k = k_0, k_0 + 1, \ldots$ and $k_0 = \lceil \frac{m+1}{2} \rceil$. Here $\lceil t \rceil$ is the smallest integer that is larger than or equal to t. Matrix $X \succeq 0$ means that X is positive semidefinite, and matrices $L_h^{(k)}(y)$, $M_k(y)$ are defined in (8) and (9). The dual optimization problem of (17) is

$$\rho_k^{(2)} := \begin{array}{cc} \max & \theta \\ \text{s.t.} & f - \theta \in I_{2k}(h). \end{array} \tag{18}$$

As in [9], it can be shown that for all k

$$\rho_k^{(2)} \le \rho_k^{(1)} \le f^s$$

and the sequences $\{\rho_k^{(1)}\}$ and $\{\rho_k^{(2)}\}$ are monotonically increasing. Furthermore, suppose y^* is a minimizer of (17). If there exists a real integer $t \in [k_0, k]$ such that

$$rank M_{t-k_0}(y^*) = rank M_t(y^*), \tag{19}$$

then $\rho_k^{(1)} = f^s$, and we can get $r := rank M_t(y^*)$ global optimizers of (16) (cf. [12]). It is clear to have the following results.

Theorem 1
Tensor $\mathcal{A}$ is a P-tensor if $\rho_k^{(1)} > 0$, and tensor $\mathcal{A}$ is a P_0-tensor but not a P-tensor if and only if $f^s = 0$. Furthermore, tensor $\mathcal{A}$ is not a P_0-tensor if and only if $f^s < 0$.

Based on this result, we present our numerical algorithm here.

Algorithm 1
To check the membership problem of $P(P_0)$-tensor

Step 0: For tensor $\mathcal{A}$, write polynomial tuples f and h as in (15). Let $k = \lceil \frac{m+1}{2} \rceil$.
Step 1: Solve the hierarchy of (17) for k and get $\rho_k^{(1)}$ with minimizer y_k^*.
Step 2: Check $\rho_k^{(1)}$. If $\rho_k^{(1)} > 0$, then $\mathcal{A}$ is a P-tensor and stop; if $\rho_k^{(1)} = 0$ and rank condition (19) with y_k^* holds for some t, then $\mathcal{A}$ is a P_0-tensor but not P-tensor and stop; and if $\rho_k^{(1)} < 0$, rank condition (19) is satisfied for some t, then $\mathcal{A}$ is not a P_0-tensor and stop. If rank condition (19) with y_k^* fails, let $k := k + 1$ and go to Step 1.

Theorem 2
Let $\mathcal{A} \in S^m(\mathbb{R}^n)$. Then we have:

1. $\lim_{k \to \infty} \rho_k^{(1)} = f^s$ when tensor $\mathcal{A}$ is not a P-tensor.
2. If $V_{\mathbb{R}}(h)$ is finite, $\rho_k^{(1)} = f^s$ and rank condition (19) is satisfied for some k. Hence, Algorithm 1 terminates in finitely many steps.

The proof can be seen from Theorem 3.1 in [13] and omitted here.

4. Numerical Examples

In this section, we give numerical examples for how to check whether a given tensor is $P(P_0)$-tensor or not. The computation is implemented in MATLAB 7.10 in a Dell Linux Desktop with 8GB memory and Intel(R) CPU 2.8GHz. The software Gloptipoly 3 [7] is used to solve the semidefinite relaxations. For convenience, we use the following notation: for any $i_1, i_2, \ldots, i_m \in [n]$, we use $\pi(i_1 i_2 \cdots i_m)$ to denote a permutation of $i_1 i_2 \cdots i_m$, and $S_{\pi(i_1 i_2 \cdots i_m)}$ to denote the set of all these permutations. $\rho_k^{(1)} = 0$ if $|\rho_k^{(1)}| < 1e - 5$ in the following numerical examples. For cleanness, only four decimal digits are displayed.

Table 1

a	times	$P(P_0)$-tensor?
-2	0.26s	not P_0
$-\sqrt[4]{15}$	0.29s	P_0 but not P
-1	0.16s	P
0	0.16s	P
1	0.16s	P
$\sqrt[4]{15}$	0.29s	P_0 but not P
2	0.27s	not P_0

Example 1
([5], Example 3.1) Consider the nonnegative tensor $\mathcal{A} \in S_{3,3}$ such that

$$a_{111} = 100, a_{222} = 3, a_{333} = 1,$$
$$a_{112} = a_{113} = a_{122} = a_{133} = 1,$$
$$a_{223} = 3, a_{233} = 2.5, a_{123} = 0.$$

By [5], $\mathcal{A}$ is not a P_0-tensor. Using Algorithm 1, it is obtained that $f^s = -100.0034$. The detection takes about 0.15 seconds to assert that the tensor is not a P_0-tensor.

Example 2
([2], Example 5.8) Consider the tensor $\mathcal{A} \in S_{6,3}$ is given by

$$\begin{cases} a_{333333} = 1, \\ \sum_{i_1 i_2 i_3 i_4 i_5 i_6 \in S_\pi(111122)} a_{i_1 i_2 i_3 i_4 i_5 i_6} = 1, \\ \sum_{i_1 i_2 i_3 i_4 i_5 i_6 \in S_\pi(112222)} a_{i_1 i_2 i_3 i_4 i_5 i_6} = 1, \\ \sum_{i_1 i_2 i_3 i_4 i_5 i_6 \in S_\pi(112233)} a_{i_1 i_2 i_3 i_4 i_5 i_6} = -3. \end{cases}$$

The corresponding polynomial of the tensor $\mathcal{A}$ is

$$f(x_1, x_2, x_3) = x_1^4 x_2^2 + x_1^2 x_2^4 + x_3^6 - 3 x_1^2 x_2^2 x_3^2.$$

This is the famous Motzkin polynomial, which is nonnegative but not a sum of squares. By Algorithms 1, it is obtained that $f^s = 0$. The detection takes about 0.60 seconds to assert that the tensor is a P-tensor but not P_0-tensor. This confirms the fact that the Motzkin polynomial $f(x_1, x_2, x_3)$ is positive semi-definite from [2].

Example 3
([3], Example 4.1) Consider tensor $\mathcal{A} \in S_{6,4}$ such that

$$a_{111111} = a_{222222} = \frac{11}{4}, \quad a_{333333} = a_{444444} = 1,$$
$$\sum_{i_1 i_2 i_3 i_4 i_5 i_6 \in S_\pi(111222)} a_{i_1 i_2 i_3 i_4 i_5 i_6} = -\frac{3}{4},$$
$$\sum_{i_1 i_2 i_3 i_4 i_5 i_6 \in S_\pi(334444)} a_{i_1 i_2 i_3 i_4 i_5 i_6} = \frac{2}{5}.$$

The corresponding polynomial of the tensor $\mathcal{A}$ is

$$f(x_1, x_2, x_3, x_4) = \frac{11}{4} x_1^6 + \frac{11}{4} x_2^6 + x_3^6 + x_4^6 - \frac{15}{2} x_1^3 x_2^3 + 6 x_3^2 x_4^4.$$

By [3], -1 is the minimum H-eigenvalue of $\mathcal{A}$, that is, $\mathcal{A}$ is not a P_0-tensor. By Algorithm 1, it is obtained that $f^s = -0.2500$. The detection takes about 1.8 seconds to assert that the tensor is not a P_0-tensor.

Example 4
Consider the tensor $\mathcal{A} \in S_{4,3}$ with the entries $a_{ijkl} = 0$ except

$$a_{1111} = 2, \ a_{2222} = 3, \ a_{3333} = 5, a_{1123} = \frac{a}{3}.$$

Similar to [15], a is chosen different values. In table 1, we report the results, where "times" denotes the elapsed time.

5. Conclusion

In this paper, we proposed a SDP relaxation method for checking whether a given symmetric tensor is $P(P_0)$-tensor or not. We determine $P(P_0)$-tensor by solving a finite number of SDP relaxations. As a prospect, whether such method can be applied to check a nonsymmetric $P(P_0)$-tensor is our future research.

Acknowledgement

REFERENCES

1. C. Cui, Y. Dai and J. Nie, *All real eigenvalues of symmetric tensors*, SIAM Journal on Matrix Analysis and Applications, vol. 35, no. 4, pp. 1582–1601, 2014.
2. D. Cartwright and B. Sturmfels, *The number of eigenvalues of a tensor*, Linear Algebra and Its Applications, vol. 438, no. 2, pp. 942–952, 2013.
3. H. Chen, Y. Chen, G. Li and L. Qi, *Positive semi-definiteness and extremal H-eigenvalues of extended essentially nonnegative tensors*, arXiv: 1511. 02328v1.
4. R.W. Cottle, J.S. Pang and R.E. Stone, *The linear complementarity problem*, Academic Press, Boston 1992.
5. W. Ding, Z. Luo and L. Qi, *P-tensor, P₀-tensor, and tensor complementarity problem*, arXiv: 1507. 06731v1.
6. F. Facchinei and J.S. Pang, *Finite dimensionl variatioal inequalities and complementarity problems*, Springer, New York 2003.
7. D. Henrion, J.B. Lasserre and Y. Lofberg, *GloptiPoly 3: Moments, Optimization and Semidfinite Programming*, Optimization, Methods and Softwares, vol. 24, no. 4-5, pp. 761–779, 2009.
8. T.G. Kolda and J.R. Mayo, *Shifted power mthod for computing tensor eigenpairs*, SIAM Journal on Matrix Analysis and Applications, vol. 32, no. 4, pp. 1095–1124, 2011.
9. J.B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM Journal on Optimization, vol. 11, no. 3, pp. 796–817, 2001.
10. J.B. Lasserre, *Moments, positive polynomials and their applications*, Imperial College Press, 2009.
11. M. Laurent, *Sums of squares, moment matrices and optimization over polynomials*, Emerging Applications of Algebraic Geometry, IMA Volumes in Mathematics and its Applications (Eds. M. Putinar and S. Sullivant), Springer, vol. 149, pp. 157–270, 2009.
12. J. Nie, *The hierarchy of local minimums in polynomial optimizaion*, Mathematical Programming, to appear.
13. J. Nie and X. Zhang, *Real eigenvalues of nonsymmetric tensors*, arXiv:1503.06881.
14. L. Qi, D. Sun, and G. Zhou, *A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities*, Mathematical Programming, vol. 87, no. 1, pp. 1–35, 2000.
15. L. Qi, *Eigenvalues of a real supersymmetric tensor*, Journal of Symbolic Computation, vol. 40, no. 6, pp. 1302–1324, 2005.
16. L. Qi, *Eigenvalues and invariants of tensors*, Journal of Mathematics Analysis and Applications, vol. 325, no. 2, pp. 1363–1377, 2007.
17. L. Qi, F. Wang and Y. Wang, *Z-eigenvalue methods for a global polynomial optimization problem*, Mathematical Programming, vol. 118, no. 2, pp. 301–316, 2009.
18. Y. Song and L. Qi, *Properties of some classes of structured tensors*, Journal of Optimization Theory and Applications, vol. 165, no. 3, pp. 854–873, 2015.
19. X. Zhang, L. Qi and Y. Ye, *The cubic spherical optimization problems*, Mathematics of Computation, vol. 81, pp. 1513–1525, 2012.