Front. Math. China 2017, 12(6): 1409–1426
https://doi.org/10.1007/s11464-017-0644-1

RESEARCH ARTICLE

Approximation algorithms for nonnegative polynomial optimization problems over unit spheres

Xinzhen ZHANG[1], **Guanglu ZHOU**[2], **Louis CACCETTA**[2], **Mohammed ALQAHTANI**[2]

1 School of Mathematics, Tianjin University, Tianjin 300072, China
2 Department of Mathematics and Statistics, Curtin University, Perth, Australia

Abstract We consider approximation algorithms for nonnegative polynomial optimization problems over unit spheres. These optimization problems have wide applications e.g., in signal and image processing, high order statistics, and computer vision. Since these problems are NP-hard, we are interested in studying on approximation algorithms. In particular, we propose some polynomial-time approximation algorithms with new approximation bounds. In addition, based on these approximation algorithms, some efficient algorithms are presented and numerical results are reported to show the efficiency of our proposed algorithms.

Keywords Approximation algorithm, polynomial optimization, approximation bound
MSC 65K10, 90C25, 90C30

1 Introduction

Let $\mathbb{R}$ be the real field, and let $\mathbb{R}^n_+$ be the nonnegative orthant in $\mathbb{R}^n$, that is, the subset of vectors with nonnegative coordinates. A d-th order *tensor* $\mathscr{T}$ is defined as

$$\mathscr{T} = (t_{i_1 i_2 \cdots i_d}), \quad t_{i_1 i_2 \cdots i_d} \in \mathbb{R}, \, 1 \leqslant i_j \leqslant n_j, \, 1 \leqslant j \leqslant d. \tag{1}$$

In this regard, a vector is a first-order tensor and a matrix is a second-order tensor. Tensors of order more than two are called higher-order tensors. In this paper, we always suppose that $d > 2$. $\mathscr{T}$ is called nonnegative if $t_{i_1 i_2 \cdots i_d} \geqslant 0$.

Received July 26, 2016; accepted February 15, 2017
Corresponding author: Guanglu ZHOU, E-mail: G.Zhou@curtin.edu.au

When $\mathscr{T}$ is nonnegative, we write it as $\mathscr{T} \geqslant 0$. For $x^i \in \mathbb{R}^{n_i}$, $i = 1, 2, \ldots, d$, let $\mathscr{T} x^1 x^2 \cdots x^d$ be the following multi-linear function defined by tensor $\mathscr{T}$:

$$\mathscr{T} x^1 x^2 \cdots x^d = \sum_{1 \leqslant i_j \leqslant n_j,\, 1 \leqslant j \leqslant d} t_{i_1 i_2 \cdots i_d} x^1_{i_1} x^2_{i_2} \cdots x^d_{i_d}. \tag{2}$$

A d-th order n-dimensional *square tensor* $\mathscr{A}$ is defined as

$$\mathscr{A} = (a_{i_1 i_2 \cdots i_d}), \quad a_{i_1 i_2 \cdots i_d} \in \mathbb{R},\, 1 \leqslant i_1, i_2, \ldots, i_d \leqslant n. \tag{3}$$

Tensor $\mathscr{A}$ is called *symmetric* if its entries $a_{i_1 i_2 \cdots i_d}$ are invariant under any permutation of their indices $\{i_1, i_2, \ldots, i_d\}$ [13]. For $x \in \mathbb{R}^n$, let $F_{\mathscr{A}}$ be a d-th degree homogeneous polynomial defined by

$$F_{\mathscr{A}}(x) := \mathscr{A} x^d = \sum_{1 \leqslant i_1, i_2, \ldots, i_d \leqslant n} a_{i_1 i_2 \cdots i_d} x_{i_1} x_{i_2} \cdots x_{i_d}. \tag{4}$$

Assume that $n, m \geqslant 2$ are positive integers. A fourth order (n, m)-dimensional *rectangular tensor* is defined as

$$\mathscr{B} = (b_{ijkl}), \quad b_{ijkl} \in \mathbb{R},\, 1 \leqslant i, j \leqslant n,\, 1 \leqslant k, l \leqslant m. \tag{5}$$

We say that $\mathscr{B}$ is a *partially symmetric* rectangular tensor [11], if

$$b_{ijkl} = b_{jikl} = b_{ijlk} = b_{jilk}, \quad 1 \leqslant i, j \leqslant n,\, 1 \leqslant k, l \leqslant m.$$

Let

$$G_{\mathscr{B}}(x, y) := \mathscr{B} xxyy = \sum_{i,j=1}^{n} \sum_{k,l=1}^{m} b_{ijkl} x_i x_j y_k y_l, \quad x \in \mathbb{R}^n,\, y \in \mathbb{R}^m. \tag{6}$$

In this paper, we consider the following two models:

$$\begin{aligned} &\text{(P1)} \quad \max F_{\mathscr{A}}(x) \\ &\qquad\quad \text{s.t. } \|x\| = 1,\, x \in \mathbb{R}^n; \\ &\text{(P2)} \quad \max G_{\mathscr{B}}(x, y) \\ &\qquad\quad \text{s.t. } \|x\| = 1,\, \|y\| = 1,\, x \in \mathbb{R}^n,\, y \in \mathbb{R}^m. \end{aligned}$$

(P1) and (P2) are both homogeneous polynomial optimization problems and they have wide applications in signal processing, biomedical engineering, and investment science; see [2,3,9,14,17,18,20–22]. Since polynomial functions are non-convex in most cases, (P1) and (P2) are NP-hard problems, see [2,11,22]. Hence, they are difficult to solve theoretically as well as numerically. Motivated by this, in this paper, we will focus on approximation algorithms for (P1) and (P2). A quality measure of approximation is defined as follows.

Definition 1.1 [11] Let the optimization problem

$$\begin{aligned}&\max\ g(x)\\&\text{s.t. } x \in \omega \subset \mathbb{R}^n\end{aligned} \tag{7}$$

be NP-hard and Algorithm $\mathscr{M}$ be a polynomial time approximation algorithm to solve (7). Algorithm $\mathscr{M}$ is said to have a relative approximation bound $C \in (0, 1]$ if for any instance of (7), Algorithm $\mathscr{M}$ can find a lower bound g for (7) such that

$$\begin{cases} Cg_{\max} \leqslant g \leqslant g_{\max}, & g_{\max} \geqslant 0, \\ g \leqslant g_{\max} \leqslant Cg, & g_{\max} < 0, \end{cases}$$

where $g_{\max}$ is the optimal value of (7). Furthermore, Algorithm $\mathscr{M}$ is called a C-bound approximation algorithm.

When Algorithm $\mathscr{M}$ returns a feasible solution $\overline{x}$ with objective value $\overline{g} = g(\overline{x})$ such that

$$\begin{cases} Cg_{\max} \leqslant \overline{g} \leqslant g_{\max}, & g_{\max} \geqslant 0, \\ \overline{g} \leqslant g_{\max} \leqslant C\overline{g}, & g_{\max} < 0, \end{cases}$$

the feasible solution $\overline{x}$ is said to be a C-bound approximation solution of the maximization model (7).

Clearly, in this definition, the closer C is to 1, the better the approximation algorithm would be.

Recently, a number of approximation algorithms for (P1) and (P2) have been proposed; see [2,3,9,11,15,18,19,23]. In the second column of Table 1, we summarize some approximation bounds for (P1) and (P2) obtained in [2,11,15, 19,23]. In [12], using the Sum-of-Squares (SOS) approach proposed by Lasserre [9], approximate solutions of (P1) were considered. In [5,6], some specially structured polynomial optimization problem with suitable sign structure has been studied via the SOS approach. The SOS approach gives tight bounds for general polynomial optimization, but it is expensive for large scale problems.

In this paper, we study (P1) and (P2), where $\mathscr{A}$ and $\mathscr{B}$ are nonnegative. When $\mathscr{A} \geqslant 0$ and $\mathscr{B} \geqslant 0$, (P1) and (P2) are nonnegative polynomial optimization problems over unit spheres. In Section 2, we present some approximation algorithms for (P1) and (P2) with some improved quality bounds. Table 1 summarizes the approximation bounds obtained in this paper. Clearly, when $d \geqslant 4$, our approximation bounds are new and better than the existing ones. In Section 3, some efficient algorithms for (P1) and (P2) are proposed, and numerical results are reported.

Some words about notation. For a vector $x = [x_1, x_2, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^n$, the 2-norm is denoted by $\|x\|$ and we use $|x|$ to denote the vector $[|x_1|, |x_2|, \ldots, |x_n|]^{\mathrm{T}}$. We use e to denote the vector of ones and e_i to denote the vector with its ith entry being 1 and other entries being 0. For a d-th order n-dimensional square tensor $\mathscr{A}$, and $x \in \mathbb{R}^n$, let $\mathscr{A}x^{d-1}$ be an n-dimensional vector defined by

Table 1 New approximation bounds for (P1) and (P2)

problem	existing quality bounds	new quality bounds
(P1)	$d!d^{-d}n^{-(d-2)/2}$ [2]	
	$\Omega\Big(\frac{(\log n)^{(d-2)/2}}{n^{(d-2)/2}}\Big)$ [15]	$d!d^{-d}n^{-\frac{d-2}{4}}$ ($\mathscr{A}\geqslant 0$, $d\geqslant 4$ even)
	$n^{-(d-2)/2}$ ($\mathscr{A}\geqslant 0$) [23]	$d!d^{-d}n^{-(d-1)/4}$ ($\mathscr{A}\geqslant 0$, $d\geqslant 5$ odd)
(P2)	$\frac{1}{2\max\{n,m\}^2}$ [11]	$\frac{1}{\sqrt{\min\{n,m\}}}$ ($\mathscr{B}\geqslant 0$)
	$\Omega\Big(\frac{\log\min\{n,m\}}{\min\{n,m\}}\Big)$ [15]	$\frac{1}{\sqrt{\min\{n,m\}}}$ ($\mathscr{B}\geqslant 0$)
	$\frac{1}{2\max\{n,m\}(\max\{n,m\}-1)}$ [19]	$\frac{1}{\sqrt{\min\{n,m\}}}$ ($\mathscr{B}\geqslant 0$)
	$\frac{1}{\sqrt{nm}}$ ($\mathscr{B}\geqslant 0$) [23]	$\frac{1}{\sqrt{\min\{n,m\}}}$ ($\mathscr{B}\geqslant 0$)

$$\mathscr{A}x^{d-1}=\Bigg(\sum_{1\leqslant i_2,i_3,\ldots,i_d\leqslant n}a_{ii_2i_3\cdots i_d}x_{i_2}x_{i_3}\cdots x_{i_d}\Bigg)_{1\leqslant i\leqslant n}. \tag{8}$$

For a fourth order (n,m)-dimensional partially symmetric rectangular tensor $\mathscr{B}$, $x\in\mathbb{R}^n$, and $y\in\mathbb{R}^m$, let $\mathscr{B}xyy$ be an n-dimensional vector defined by

$$\mathscr{B}xyy=\Bigg(\sum_{j=1}^{n}\sum_{k,l=1}^{m}b_{ijkl}x_jy_ky_l\Bigg)_{1\leqslant i\leqslant n}, \tag{9}$$

and let $\mathscr{B}xxy$ be an m-dimensional vector defined by

$$\mathscr{B}xxy=\Bigg(\sum_{i,j=1}^{n}\sum_{l=1}^{m}b_{ijkl}x_ix_jy_l\Bigg)_{1\leqslant k\leqslant m}. \tag{10}$$

2 Approximation solutions for (P1) and (P2)

In this section, we will present some new approximation bounds for (P1) and (P2). We first give some lemmas which will be used later.

For a symmetric matrix M, let $\sigma(M)$ denote the spectrum of M, the set of all eigenvalues of M. The *spectral radius* of M, denoted by $\rho(M)$, is the maximum distance of an eigenvalue from the origin, i.e.,

$$\rho(M)=\max\{|\lambda|:\lambda\in\sigma(M)\}.$$

Throughout this paper, it is assumed that the eigenvectors of a matrix are unit vectors.

Lemma 2.1 *For an $n \times n$ nonnegative matrix M, there holds that*

$$\rho(M) \leqslant \max_i \sum_{j=1}^{n} M_{ij}.$$

This lemma is an immediate corollary of the Gershgorin Circle Theorem, so the proof is omitted here.

Lemma 2.2 *Suppose that the $n \times n$ matrix M is nonnegative and symmetric. Then, for any unit vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$, there exists a unit vector $u \in \mathbb{R}^n$ such that*

$$x^{\mathrm{T}} M y \leqslant u^{\mathrm{T}} M u.$$

Proof Since M is an $n \times n$ symmetric nonnegative matrix, by [1, Theorem 4.1], we have

$$\max\{x^{\mathrm{T}} M y \colon \|x\| = \|y\| = 1\} = \max\{x^{\mathrm{T}} M x \colon \|x\| = 1\}.$$

Let u be a nonnegative eigenvector of M associated with the spectral radius of M, $\rho(M)$. Then, u is a global optimal solution of the problem

$$\max\{x^{\mathrm{T}} M x \colon \|x\| = 1\}.$$

Therefore, for any unit vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$, $x^{\mathrm{T}} M y \leqslant u^{\mathrm{T}} M u$. □

Lemma 2.3 (P1) *and* (P2) *have nonnegative global optimal solutions.*

Proof Since unit spheres are closed and bounded, (P1) and (P2) have global optimal solutions. Suppose that $x^* \in \mathbb{R}^n$ is a global optimal solution of (P1), and let

$$|x^*| = [|x_1^*|, |x_2^*|, \ldots, |x_n^*|]^{\mathrm{T}}.$$

Then we have $\|x^*\| = 1$ and

$$F_{\mathscr{A}}(x^*) \geqslant F_{\mathscr{A}}(x), \quad \forall\, x \in \mathbb{R}^n,\ \|x\| = 1.$$

Because the 2-norm of $|x^*|$ is 1, we have

$$F_{\mathscr{A}}(|x^*|) \leqslant F_{\mathscr{A}}(x^*).$$

Since $\mathscr{A}$ is a nonnegative tensor, we obtain

$$F_{\mathscr{A}}(|x^*|) \geqslant F_{\mathscr{A}}(x^*).$$

Hence,

$$F_{\mathscr{A}}(|x^*|) = F_{\mathscr{A}}(x^*),$$

which implies that $|x^*|$ is a global optimal solution of (P1). Similarly, we can show that (P2) has a nonnegative global optimal solution. □

Now, we present the following approximation algorithm for (P2). Suppose that $n \leqslant m$ in (P2) and the fourth order (n, m)-dimensional partial symmetric tensor $\mathscr{B}$ is nonnegative.

Algorithm 2.1 **Step 0** Input a nonnegative fourth order (n, m)-dimensional tensor $\mathscr{B} = (b_{ijkl})$.

Step 1 Compute $M_i = (\sum_{j=1}^{n} b_{ijkl})_{1 \leqslant k,l \leqslant m}$, $i = 1, 2, \dots, n$. Then, for $i = 1, 2, \dots, n$, compute the largest eigenvalue λ_i^* and the associated nonnegative eigenvector y^{i*}.

Step 2 Let $\lambda_j^* = \max\{\lambda_i^*, 1 \leqslant i \leqslant n\}$, $\overline{y} = y^{j*}$, and $M = \mathscr{B}\overline{y}^2$. Then, compute the largest eigenvalue λ^* of M and the associated nonnegative eigenvector $\overline{x}$. Output $(\overline{x}, \overline{y})$.

Since we can find the largest eigenvalue and the associated eigenvector for symmetric nonnegative matrices in polynomial time, Algorithm 2.1 is a polynomial-time approximation algorithm for (P2).

Theorem 2.1 *Let*

$$G_{\max} := \max \ G_{\mathscr{B}}(x, y) \quad \text{s.t. } \|x\| = 1, \|y\| = 1, x \in \mathbb{R}^n, y \in \mathbb{R}^m.$$

Then, Algorithm 2.1 *produces a* $(\min\{n, m\})^{-1/2}$*-bound approximation solution for* (P2)*.*

Proof By Lemma 2.3, (P2) has a nonnegative global optimal solution pair. For any $y \in \mathbb{R}^m$, let $\mathscr{B}y^2$ be a matrix defined with its elements

$$(\mathscr{B}y^2)_{ij} = \sum_{k,l=1}^{m} b_{ijkl} y_k y_l, \quad 1 \leqslant i, j \leqslant n.$$

Then, by the fact that an optimal solution pair of problem (P2) can be nonnegative, we have

$$\begin{aligned} G_{\max} &= \max_{\|x\|=\|y\|=1,\, x \in \mathbb{R}^n_+,\, y \in \mathbb{R}^m_+} G_{\mathscr{B}}(x, y) \\ &= \max_{\|y\|=1,\, y \in \mathbb{R}^m_+} \max_{\|x\|=1,\, x \in \mathbb{R}^n_+} x^{\mathrm{T}}(\mathscr{B}y^2)x \\ &\leqslant \max_{\|y\|=1,\, y \in \mathbb{R}^m_+} \rho(\mathscr{B}y^2) \\ &\leqslant \max_{\|y\|=1,\, y \in \mathbb{R}^m_+} \max_i \sum_{1 \leqslant j \leqslant n,\, 1 \leqslant k,l \leqslant m} b_{ijkl} y_k y_l \quad \text{(by Lemma 2.1)} \\ &= \max_{\|y\|=1,\, y \in \mathbb{R}^m_+} \max_i y^{\mathrm{T}}(M_i)y \\ &\leqslant \max_{\|y\|=1,\, y \in \mathbb{R}^m_+} \max_i \rho(M_i) \\ &= \max_i \rho(M_i). \end{aligned}$$

Without loss of generality, let $\rho(M_1) = \max_i \rho(M_i)$, and

$$\rho(M_1) = \overline{y}^{\mathrm{T}} M_1 \overline{y} = \sum_{1 \leqslant j \leqslant n,\, 1 \leqslant k,l \leqslant m} b_{1jkl} \overline{y}_k \overline{y}_l$$

for some $\overline{y} \in \mathbb{R}^m_+$. We have

$$\rho(M_1) = e_1^{\mathrm{T}} (\mathscr{B}\overline{y}^2) e = \sqrt{n}\, e_1^{\mathrm{T}} (\mathscr{B}\overline{y}^2) \frac{e}{\sqrt{n}}.$$

Since $\mathscr{B}\overline{y}^2$ is a symmetric matrix, by Lemma 2.2, we have

$$\rho(M_1) = \sqrt{n}\, e_1^{\mathrm{T}} (\mathscr{B}\overline{y}^2) \frac{e}{\sqrt{n}} \leqslant \sqrt{n}\, \overline{x}^{\mathrm{T}} (\mathscr{B}\overline{y}^2) \overline{x} = \sqrt{n}\, G_{\mathscr{B}}(\overline{x}, \overline{y}) \leqslant \sqrt{n}\, G_{\max},$$

where $\overline{x}$ is a nonnegative eigenvector of $\mathscr{B}\overline{y}^2$ associated with the spectral radius of $\mathscr{B}\overline{y}^2$. Hence,

$$G_{\max} \leqslant \rho(M_1) \leqslant \sqrt{n}\, G_{\mathscr{B}}(\overline{x}, \overline{y}) \leqslant \sqrt{n}\, G_{\max}.$$

Therefore,

$$\frac{1}{\sqrt{n}} G_{\max} \leqslant G_{\mathscr{B}}(\overline{x}, \overline{y}) \leqslant G_{\max}.$$

This implies that $(\overline{x}, \overline{y})$ is an $n^{-1/2}$-bound approximation solution of (P2). □

Lemma 2.4 *Suppose that $\mathscr{A}$ in problem* (P1) *is a nonnegative d-th order n-dimensional symmetric tensor. For any nonnegative unit vectors $x^{(i)} \in \mathbb{R}^n$, $i = 1, 2, \ldots, d$, we can find a nonnegative unit vector $\overline{x} \in \mathbb{R}^n$ such that*

$$d! d^{-d} \mathscr{A} x^{(1)} x^{(2)} \cdots x^{(d)} \leqslant \mathscr{A} \overline{x}^d = F_{\mathscr{A}}(\overline{x}).$$

Proof Let

$$\begin{aligned} \overline{y} &= \frac{\sum_{i=1}^d \xi_i^* x^{(i)}}{\| \sum_{i=1}^d \xi_i^* x^{(i)} \|} \\ &= \arg\max \left\{ \mathscr{A} \left(\frac{\sum_{i=1}^d \xi_i x^{(i)}}{\| \sum_{i=1}^d \xi_i x^{(i)} \|} \right)^d : \xi_i \in \{-1, 1\},\, i = 1, 2, \ldots, n \right\}. \end{aligned} \tag{11}$$

By [2, Lemma 1], we have

$$d! d^{-d} \mathscr{A} x^{(1)} x^{(2)} \cdots x^{(d)} \leqslant \mathscr{A} \overline{y}^d.$$

Let $\overline{x} = |\overline{y}|$. Then we obtain

$$d! d^{-d} \mathscr{A} x^{(1)} x^{(2)} \cdots x^{(d)} \leqslant \mathscr{A} \overline{y}^d \leqslant \mathscr{A} \overline{x}^d.$$ □

Let $\mathscr{A}$ be a nonnegative fourth order n-dimensional symmetric tensor, and consider the following three optimization problems:

$$\begin{aligned} \max\ & \mathscr{A}x^4 \\ \text{s.t.}\ & x^{\mathrm{T}}x = 1,\ x \in \mathbb{R}^n; \end{aligned} \tag{12}$$

$$\begin{aligned} \max\ & \mathscr{A}x^2y^2 \\ \text{s.t.}\ & x^{\mathrm{T}}x = 1,\ y^{\mathrm{T}}y = 1,\ x \in \mathbb{R}^n,\ y \in \mathbb{R}^n; \end{aligned} \tag{13}$$

$$\begin{aligned} \max\ & \mathscr{A}xyzw \\ \text{s.t.}\ & x^{\mathrm{T}}x = 1,\ y^{\mathrm{T}}y = 1,\ z^{\mathrm{T}}z = 1,\ w^{\mathrm{T}}w = 1,\ x, y, z, w \in \mathbb{R}^n. \end{aligned} \tag{14}$$

Lemma 2.5 *Problems* (12)–(14) *have the same optimal values.*

Proof By [20, Theorem 2.1] or [1, Theorem 4.1], the result holds. □

Next, we give an approximation algorithm for (P1) with $d = 4$.

Algorithm 2.2 Step 0 Input a nonnegative fourth order n-dimensional symmetric tensor $\mathscr{A}$.

Step 1 Compute an approximation solution $(\overline{u}, \overline{v})$ of (13) by Algorithm 2.1.

Step 2 Compute a nonnegative unit vector $\overline{x} \in \mathbb{R}^n$ such that $4!\,4^{-4}\mathscr{A}\overline{u}^2\overline{v}^2 \leqslant \mathscr{A}\overline{x}^4$, and output $\overline{x}$.

By Theorem 2.1, Lemmas 2.4 and 2.5, the following result holds.

Theorem 2.2 *Algorithm* 2.2 *produces a* $4!\,4^{-4}n^{-1/2}$*-bound approximation solution for* (P1) *with* $d = 4$.

Proof Let $g_{\max}$ be the optimal value of problem (12). By Lemma 2.5, $g_{\max}$ is also the optimal value of its bi-quadratic relaxation problem (13). It follows from Theorem 2.1 that we can find nonnegative unit vectors $\overline{u}, \overline{v} \in \mathbb{R}^n$ such that

$$g_{\max} \leqslant \sqrt{n}\, \mathscr{A}\overline{u}^2\overline{v}^2.$$

By Lemma 2.4, we can find a nonnegative unit vector $\overline{x} \in \mathbb{R}^n$ in constant time such that

$$4!\,4^{-4}\mathscr{A}\overline{u}^2\overline{v}^2 \leqslant \mathscr{A}\overline{x}^4.$$

Hence,

$$g_{\max} \leqslant \sqrt{n}\, \mathscr{A}\overline{u}^2\overline{v}^2 \leqslant \frac{\sqrt{n}}{4!\,4^{-4}}\, \mathscr{A}\overline{x}^4.$$

Therefore,

$$\frac{4!\,4^{-4}}{\sqrt{n}}\, g_{\max} \leqslant \mathscr{A}\overline{x}^4 \leqslant g_{\max}.$$

This means that $\overline{x}$ is a $4!\,4^{-4}n^{-1/2}$-bound approximation solution of problem (12). □

To extend Theorem 2.2 to the general case of (P1), we need the results of (P1) with $d = 3$ which are stated as follows.

Theorem 2.3 [23] *Suppose that* $\mathscr{A}$ *in problem* (P1) *is a nonnegative third order* n*-dimensional symmetric tensor. Then, there exists an* $n^{-1/2}$*-bound approximation algorithm for* (P1).

Theorem 2.4 [2,22] *Suppose that* $\mathscr{A}$ *in problem* (P1) *is a third order* n*-dimensional symmetric nonnegative tensor. Then there exists a* $3!\,3^{-3}n^{-1/2}$*-bound approximation solution for* (P1).

In addition, by [22, Theorem 5.1], a $3!\,3^{-3}n^{-1/2}$-bound approximation solution for (P1) with $d = 3$ can be obtained by the following approximation algorithm.

Algorithm 2.3 **Step 0** Input a nonnegative third order n-dimensional symmetric tensor $\mathscr{A} = (a_{ijk})$.

Step 1 Compute $M_i = (a_{ijk})_{1\leqslant j,k\leqslant n}, i = 1, 2, \ldots, n$. Then, for $i = 1, 2, \ldots, n$, compute the largest eigenvalue λ_i of M_i and the associated nonnegative eigenvector y^i. Let $\lambda_{i^*} = \max\{\lambda_i, 1 \leqslant i \leqslant n\}$ and y^{i^*} be the nonnegative eigenvector associated with λ_{i^*}.

Step 2 Find a nonnegative unit vector $\overline{x} \in \mathbb{R}^n$ such that $3!\,3^{-3}\mathscr{A}e_{i^*}(y^{i*})^2 \leqslant \mathscr{A}\overline{x}^3$, and output $\overline{x}$.

Now, we are ready to present our main result.

Theorem 2.5 *Suppose that* $\mathscr{A}$ *is a nonnegative symmetric tensor and* $d \geqslant 3$*. Then, when* d *is even, there exists an* $n^{-(d-2)/4}$*-bound approximation algorithm for* $(P1)$*; when* d *is odd, there exists an* $n^{-(d-1)/4}$*-bound approximation algorithm for* (P1)*. Furthermore, when* d *is even, there exists a* $d!d^{-d}n^{-(d-2)/4}$*-bound approximation solution for* (P1)*; when* d *is odd, there exists a* $d!d^{-d}\cdot n^{-(d-1)/4}$*-bound approximation solution for* (P1).

Proof Let $f_{\max}(\mathscr{A})$ be the optimal value of (P1). We first prove the desired result by induction when $d \geqslant 4$ and $d = 2k$ is even.

For case of $k = 2$, the result holds by Theorem 2.1. That is, we can find nonnegative unit vectors $x^1, x^2 \in \mathbb{R}^n$ in polynomial time such that

$$f_{\max}(\mathscr{A}) \leqslant \sqrt{n}\,\mathscr{A}(x^1)^2(x^2)^2.$$

Assume that our result holds for $k-1$ ($k \geqslant 2$). That is, $d = 2(k-1)$ and nonnegative unit vectors $x^1, x^2, \ldots, x^{k-1} \in \mathbb{R}^n$ can be found in polynomial time such that

$$f_{\max}(\mathscr{A}) \leqslant n^{(d-2)/4}\mathscr{A}(x^1)^2(x^2)^2\cdots(x^{k-1})^2.$$

Now, we show that the result holds for k. For this case, $d = 2k$. Similar to Lemma 2.5, we have the equivalence between problem (P1) and the following multi-homogeneous optimization problem:

$$\begin{aligned} &\max\ \mathscr{A}x^2y^{d-2} \\ &\quad \text{s.t. } x^{\mathrm{T}}x = 1,\ y^{\mathrm{T}}y = 1. \end{aligned}$$

Based on this equivalence and Lemma 2.1, we obtain that

$$\begin{aligned}
f_{\max}(\mathscr{A}) &= \max_{y^{\mathrm{T}}y=1}\max_{x^{\mathrm{T}}x=1}\mathscr{A}x^2y^{2k-2}\\
&= \max_{y^{\mathrm{T}}y=1}\rho(\mathscr{A}y^{2k-2})\\
&\leqslant \max_{y^{\mathrm{T}}y=1}\max_{i}\sum_{j=1}^{n}(\mathscr{A}y^{2k-2})_{ij}\\
&= \max_{y^{\mathrm{T}}y=1}\max_{i}(\mathscr{A}y^{2k-2}e)_i\\
&= \max_{1\leqslant i\leqslant n}\max_{y^{\mathrm{T}}y=1}(\mathscr{A}y^{2k-2}e)_i\\
&= \sqrt{n}\max_{1\leqslant i\leqslant n}\max_{y^{\mathrm{T}}y=1}\Big(\mathscr{A}y^{2k-2}\frac{e}{\sqrt{n}}\Big)_i\\
&= \sqrt{n}\max_{1\leqslant i\leqslant n}\Big\{\max_{y^{\mathrm{T}}y=1}\mathscr{A}\frac{e}{\sqrt{n}}e_iy^{2k-2}\Big\}.
\end{aligned}$$

For $i=1,2,\dots,n$, $\mathscr{G}_i:=\mathscr{A}\frac{e}{\sqrt{n}}e_i$ is a $(2k-2)$-nd order n-dimensional symmetric nonnegative tensor. Then there holds

$$f_{\max}(\mathscr{A})\leqslant\sqrt{n}\max_{1\leqslant i\leqslant n}\big\{\max_{y^{\mathrm{T}}y=1}\mathscr{G}_iy^{2k-2}\big\}.$$

For $i=1,2,\dots,n$, by our assumption, we can find nonnegative unit vectors $y^{(i,1)},y^{(i,2)},\dots,y^{(i,k-1)}\in\mathbb{R}^n$ in polynomial time such that

$$f_{\max}(\mathscr{G}_i)=\max_{y^{\mathrm{T}}y=1}\mathscr{G}_iy^{2k-2}\leqslant n^{(k-2)/2}\mathscr{G}_i(y^{(i,1)})^2(y^{(i,2)})^2\cdots(y^{(i,k-1)})^2.$$

Hence, we obtain that

$$\begin{aligned}
f_{\max}(\mathscr{A}) &\leqslant \sqrt{n}\max_{1\leqslant i\leqslant n}\{n^{(k-2)/2}\mathscr{G}_i(y^{(i,1)})^2(y^{(i,2)})^2\cdots(y^{(i,k-1)})^2\}\\
&= n^{(k-1)/2}\max_{1\leqslant i\leqslant n}\{\mathscr{G}_i(y^{(i,1)})^2(y^{(i,2)})^2\cdots(y^{(i,k-1)})^2\}\\
&= n^{(k-1)/2}\max_{1\leqslant i\leqslant n}\Big\{\mathscr{A}\frac{e}{\sqrt{n}}e_i(y^{(i,1)})^2(y^{(i,2)})^2\cdots(y^{(i,k-1)})^2\Big\}.
\end{aligned}$$

Suppose that i_0 is the index such that

$$\begin{aligned}
&\mathscr{A}\frac{e}{\sqrt{n}}e_{i_0}(y^{(i_0,1)})^2(y^{(i_0,2)})^2\cdots(y^{(i_0,k-1)})^2\\
&\quad= \max_{1\leqslant i\leqslant n}\Big\{\mathscr{A}\frac{e}{\sqrt{n}}e_i(y^{(i,1)})^2(y^{(i,2)})^2\cdots(y^{(i,k-1)})^2\Big\}.
\end{aligned}$$

Then there holds

$$\begin{aligned}
f_{\max}(\mathscr{A}) &\leqslant n^{(k-1)/2}\mathscr{A}\frac{e}{\sqrt{n}}e_{i_0}(y^{(i_0,1)})^2(y^{(i_0,2)})^2\cdots(y^{(i_0,k-1)})^2\\
&= n^{(d-2)/4}\mathscr{A}\frac{e}{\sqrt{n}}e_{i_0}(y^{(i_0,1)})^2(y^{(i_0,2)})^2\cdots(y^{(i_0,k-1)})^2.
\end{aligned}$$

Hence, we have proven that there exists an $n^{-(d-2)/4}$-bound when $d \geqslant 4$ and d is even by induction. Furthermore, by Lemma 2.4, we can find a nonnegative unit vector $\overline{x} \in \mathbb{R}^n$ in constant time such that

$$d!d^{-d}\mathscr{A}\frac{e}{\sqrt{n}}e_{i_0}(y^{(i_0,1)})^2(y^{(i_0,2)})^2\cdots(y^{(i_0,k-1)})^2 \leqslant \mathscr{A}\overline{x}^d.$$

So, we have

$$f_{\max}(\mathscr{A}) \leqslant n^{(d-2)/4}\mathscr{A}\frac{e}{\sqrt{n}}e_{i_0}(y^{(i_0,1)})^2(y^{(i_0,2)})^2\cdots(y^{(i_0,k-1)})^2 \leqslant \frac{n^{(d-2)/4}}{d!d^{-d}}\mathscr{A}\overline{x}^d.$$

Therefore,

$$d!d^{-d}n^{-(d-2)/4}f_{\max}(\mathscr{A}) \leqslant \mathscr{A}\overline{x}^d \leqslant f_{\max}(\mathscr{A}).$$

This means that $\overline{x}$ is a $d!d^{-d}n^{-(d-2)/4}$-bound approximation solution of (P1).

Similarly, we can prove that there exists a $d!d^{-d}n^{-(d-1)/4}$-bound approximation solution for (P1) when $d \geqslant 3$ and d is odd. □

From the proof of Theorem 2.5, we present a polynomial-time approximation algorithm for (P1) when $d \geqslant 3$ as follows.

Algorithm 2.4 **Step 0** Input a nonnegative d-th order n-dimensional symmetric tensor $\mathscr{A}$. Here, $d \geqslant 3$. If $d = 2k+2$, $k \geqslant 2$, then go to Step 1. If $d = 2k+1$, $k \geqslant 1$, then go to Step 3.

Step 1 Compute the matrices $\mathscr{A}(e/\sqrt{n})^k e_{i_1}e_{i_2}\cdots e_{i_k}$, $1 \leqslant i_1, i_2, \dots, i_k \leqslant n$. For each matrix $\mathscr{A}(e/\sqrt{n})^k e_{i_1}e_{i_2}\cdots e_{i_k}$, compute the largest eigenvalue $\lambda^{(i_1,i_2,\dots,i_k)}$ and the associated nonnegative eigenvector $y^{(i_1,i_2,\dots,i_k)}$. Let

$$\lambda^{(i_1^*,i_2^*,\dots,i_k^*)} = \max\{\lambda^{(i_1,i_2,\dots,i_k)} : 1 \leqslant i_1, i_2, \dots, i_k \leqslant n\}, \tag{15}$$

and let $y^{(i_1^*,i_2^*,\dots,i_k^*)}$ be the nonnegative eigenvector associated with $\lambda^{(i_1^*,i_2^*,\dots,i_k^*)}$.

Step 2 Compute a nonnegative unit vector $\overline{x} \in \mathbb{R}^n$ such that

$$d!d^{-d}\mathscr{A}\left(\frac{e}{\sqrt{n}}\right)^k e_{i_1^*}e_{i_2^*}\cdots e_{i_k^*}(y^{(i_1^*,i_2^*,\dots,i_k^*)})^2 \leqslant \mathscr{A}\overline{x}^d.$$

Output $\overline{x}$, and stop.

Step 3 Compute the matrices $\mathscr{A}(e/\sqrt{n})^{k-1} e_{i_1}e_{i_2}\cdots e_{i_k}$, $1 \leqslant i_1, i_2, \dots, i_k \leqslant n$. For each matrix $\mathscr{A}(e/\sqrt{n})^{k-1} e_{i_1}e_{i_2}\cdots e_{i_k}$, compute the largest eigenvalue $\lambda^{(i_1,i_2,\dots,i_k)}$ and the associated nonnegative eigenvector $y^{(i_1,i_2,\dots,i_k)}$. Let $\lambda^{(i_1^*,i_2^*,\dots,i_k^*)}$ be defined as in (15), and let $y^{(i_1^*,i_2^*,\dots,i_k^*)}$ be the nonnegative eigenvector associated with $\lambda^{(i_1^*,i_2^*,\dots,i_k^*)}$.

Step 4 Compute a nonnegative unit vector $\overline{x} \in \mathbb{R}^n$ such that

$$d!d^{-d}\mathscr{A}\left(\frac{e}{\sqrt{n}}\right)^{k-1} e_{i_1^*}e_{i_2^*}\cdots e_{i_k^*}(y^{(i_1^*,i_2^*,\dots,i_k^*)})^2 \leqslant \mathscr{A}\overline{x}^d.$$

Output $\overline{x}$, and stop.

Clearly, when $d = 3$, Algorithm 2.4 reduces to Algorithm 2.3; when $d = 4$, Algorithm 2.4 becomes to Algorithm 2.2.

3 Efficient algorithms for (P1) and (P2)

In this section, we present some efficient methods for solving (P1) and (P2), based on the approximation algorithms proposed in Section 2 and some local search procedures for (P1) and (P2) [8,16]. Usually, the solutions obtained by the approximation algorithms in Section 2 may be not local optimal solutions for (P1) and (P2). Starting from these solutions, by the local search procedures for (P1) and (P2) [8,16], we can obtain some further improved solutions for (P1) and (P2); see Tables 2 and 3.

Table 2 Numerical comparison of Algorithms 2.4 and 3.1

n	instance	Algorithm 2.4, Value 1	Algorithm 3.1, Value 2
9	1	0.0684	0.0689
	2	0.3255	0.3255
	3	0.4045	0.4045
	4	0.0353	0.0366
	5	0.0647	0.0652
	6	0.2999	0.3000
	7	0.0750	0.0755
	8	0.0655	0.0661
	9	1.3449	1.3449
	10	0.4587	0.4588
12	1	0.0445	0.0454
	2	0.1789	0.1790
	3	0.0848	0.0852
	4	0.4533	0.4533
	5	0.0472	0.0480
	6	0.2581	0.2582
	7	0.4031	0.4032
	8	0.5213	0.5213
	9	0.0527	0.0535
	10	0.4419	0.4419

Now, we propose an algorithm for (P1) as follows.

Algorithm 3.1 Initial Step Input a nonnegative symmetric tensor $\mathscr{A}$ and $c > 0$. By Algorithm 2.4, obtain an initial point $x^{(0)} \in \mathbb{R}^n_+$. Let $f_0 = \mathscr{A}[x^{(0)}]^d$ and $k := 0$.

Iterative Step **for** $k = 1, 2, \dots$ **do**

$$x^{(k)} = \frac{\mathscr{A}[x^{(k-1)}]^{(d-1)} + cx^{(k-1)}}{\|\mathscr{A}[x^{(k-1)}]^{(d-1)} + cx^{(k-1)}\|}, \quad f_k = \mathscr{A}[x^{(k)}]^d.$$

end for

Table 3 Numerical comparison of Algorithms 2.1 and 3.2

(n,m)	instance	Algorithm 2.1, Value 3	Algorithm 3.2, Value 4
$(6,9)$	1	0.3709	0.3710
	2	0.2246	0.2247
	3	0.0011	0.0013
	4	0.9864	0.9864
	5	0.1359	0.1362
	6	0.0210	0.0232
	7	0.0322	0.0336
	8	0.3765	0.3765
	9	0.0500	0.0508
	10	0.2502	0.2502
$(6,12)$	1	0.0001	0.0102
	2	0.1926	0.1927
	3	0.0003	0.0142
	4	0.0765	0.0771
	5	0.1426	0.1428
	6	0.0511	0.0519
	7	0.1307	0.1309
	8	0.0451	0.0461
	9	0.0115	0.0154
	10	0.2532	0.2533

Algorithm 3.1 includes two parts: the initial step and the iterative step. It is easy to see that the initial step returns an approximation solution with a bound defined in Theorem 2.5. The iterative step is the SS-HOPM [8], so according to [8, Theorem 4.4], we have the following convergence result.

Theorem 3.1 *Suppose that* $\{x^{(k)}\}$ *is an infinite sequence generated by Algorithm* 3.1. *If* $c > \beta(\mathscr{A})$, *where*

$$\beta(\mathscr{A}) = (d-1)\max_{x\in\Sigma}\rho(\mathscr{A}x^{d-2}), \quad \Sigma = \{x\in\mathbb{R}^n\colon \|x\| = 1\},$$

then $\{x^{(k)}\}$ *converges to a KKT point of* (P1).

For Algorithm 3.1, in practice, we may stop the iteration when $|f_k - f_{k-1}| \leqslant 10^{-5}$. For (P2), we state an algorithm in the following.

Algorithm 3.2 **Initial Step** Input a nonnegative fourth order (n,m)-dimensional partial symmetric tensor $\mathscr{B}$ and $C_1, C_2 > 0$. By Algorithm 2.1, obtain an initial point $(x^{(0)}, y^{(0)}) \in \mathbb{R}^n_+ \times \mathbb{R}^m_+$. Let $g_0 = \mathscr{B}x^{(0)}x^{(0)}y^{(0)}y^{(0)}$ and $k := 0$.

Iterative Step **for** $k = 1, 2, \ldots$ **do**

$$x^{(k)} = \frac{\mathscr{B}x^{(k-1)}y^{(k-1)}y^{(k-1)} + C_1x^{(k-1)}}{\|\mathscr{B}x^{(k-1)}y^{(k-1)}y^{(k-1)} + C_1x^{(k-1)}\|},$$

$$y^{(k)} = \frac{\mathscr{B}x^{(k)}x^{(k)}y^{(k-1)} + C_2y^{(k-1)}}{\|\mathscr{B}x^{(k)}x^{(k)}y^{(k-1)} + C_2y^{(k-1)}\|},$$

$$g_k = \mathscr{B}x^{(k)}x^{(k)}y^{(k)}y^{(k)}.$$

end for

Similar as Algorithm 3.1, Algorithm 3.2 also includes two parts: the initial step and the iterative step. The initial step returns an approximation solution with a bound defined in Theorem 2.1. The iterative step is [16, Algorithm 5.1]. For Algorithm 3.2, in practice, we may stop the iteration when $|g_k - g_{k-1}| \leqslant 10^{-5}$. By [16, Theorem 5.1], we have the following convergence result for Algorithm 3.2.

Theorem 3.2 *Suppose that* $\{x^{(k)}, y^{(k)}\}$ *is an infinite sequence generated by Algorithm* 3.2. *If*

$$C_1, C_2 \geqslant \tau := \max\{|\mathscr{B}x^2y^2| \colon \|x\| = \|y\| = 1\},$$

then any accumulation point of the sequence is a KKT point of (P2).

In the following, we will first compare Algorithms 3.1 and 3.2 with the approximation algorithms in Section 2. Then we are going to compare our algorithms proposed in this section with two competing methods. One of them is the SOS method [9,10], based on which, Henrion et al. [4] developed a specialized Matlab toolbox known as GloptiPoly3, and the other one is the ADM method [7]. GloptiPoly3 is designed for general polynomial optimization problems and it can produce global minimizers and global maximizers. The ADM method [7] is a local search algorithm for general polynomial optimization problems and theoretically it can produce local minimizers and local maximizers. The main purpose of comparing Algorithms 3.1 and 3.2 with GloptiPoly3 is to show that Algorithms 3.1 and 3.2 may produce global solutions for large scale problems. All algorithms are implemented in MATLAB (R2010b) and all the numerical computations are conducted using an Intel 3.30 GHz computer with 8 GB of RAM. All test problems are randomly generated.

In Tables 2 and 3, we report the performance comparison of Algorithms 3.1 and 3.2 with the approximation algorithms in Section 2. In Table 2, we report our numerical results of Algorithms 2.4 and 3.1 for some randomly generated tensor $\mathscr{A}$ with $d = 4$. Value 1 and Value 2 denote the values of $F_{\mathscr{A}}(x)$ at the final iteration of Algorithms 2.4 and 3.1, respectively. In Table 3, Value 3 and Value 4 denote the values of $G_{\mathscr{B}}(x, y)$ at the final iteration of Algorithms 2.1 and 3.2, respectively. From these two tables, clearly, we can see Algorithms 3.1 and 3.2 can produce solutions with bigger objective function values than the approximation algorithms in Section 2.

Now, we report our numerical results of Algorithm 3.1, the ADM method [7], and GloptiPoly3 [4] for solving (P1). We tested these algorithms for some randomly generated tensor $\mathscr{A}$ with $d = 4$. Our numerical results are reported in Tables 4 and 5. In these tables, for Algorithm 3.1, 'value' denotes the value of $F_{\mathscr{A}}(x)$ at the final iteration, and 'time' denotes the total computer time in seconds used to solve the problem. For the ADM method, 'best value' denotes the best solution value among the ones generated by 5 different starting points,

Table 4 Numerical results of Algorithm 3.1, ADM [7], and GloptiPoly3 [4] for small size problems

n	instance	Algorithm 3.1		ADM		GloptiPoly3	
		value	time	best value	aver. time	value	time
6	1	17.9380	0.0192	17.9380	0.0403	17.9380	0.5821
	2	18.1761	0.0070	18.1761	0.0264	18.1761	0.0959
	3	18.0296	0.0038	18.0296	0.0267	18.0296	0.0878
	4	17.8157	0.0048	17.8157	0.0266	17.8157	0.1172
	5	17.7139	0.0037	17.7139	0.0268	17.7139	0.0992
	6	17.8932	0.0047	17.8932	0.0267	17.8932	0.1001
	7	18.0886	0.0036	18.0886	0.0268	18.0886	0.1015
	8	18.0627	0.0049	18.0627	0.0265	18.0627	0.1006
	9	17.9229	0.0047	17.9229	0.0265	17.9229	0.0874
	10	17.8302	0.0048	17.8302	0.0268	17.8302	0.0991
9	1	40.1582	0.0118	40.1582	0.0269	40.1582	0.8595
	2	40.2443	0.0120	40.2443	0.0270	40.2443	0.8655
	3	40.3084	0.0119	40.3084	0.0271	40.3084	0.8513
	4	40.3257	0.0132	40.3257	0.0271	40.3257	0.8642
	5	39.7725	0.0119	39.7725	0.0270	39.7725	0.7725
	6	40.7858	0.0118	40.7858	0.0313	40.7858	0.7262
	7	40.4937	0.0118	40.4937	0.0269	40.4937	0.7612
	8	40.6219	0.0118	40.6219	0.0316	40.6219	0.6264
	9	40.2984	0.0119	40.2984	0.0273	40.2984	0.7618
	10	40.4987	0.0118	40.4987	0.0228	40.4987	0.6870
12	1	72.0432	0.0287	72.0432	0.0276	72.0432	6.5892
	2	72.4487	0.0289	72.4487	0.0232	72.4487	6.3997
	3	72.1980	0.0284	72.1980	0.0230	72.1980	6.4412
	4	71.8164	0.0387	71.8164	0.0339	71.8164	6.5011
	5	71.8175	0.0286	71.8175	0.0232	71.8175	7.4755
	6	72.2376	0.0408	72.2376	0.0339	72.2376	6.4816
	7	71.8316	0.0283	71.8316	0.0277	71.8316	7.5805
	8	72.2826	0.0281	72.2826	0.0283	72.2826	6.5865
	9	72.0180	0.0285	72.0180	0.0277	72.0180	7.5763
	10	71.8804	0.0281	71.8804	0.0234	71.8804	6.4136
15	1	112.7589	0.0578	112.7589	0.0290	112.7589	72.0025
	2	112.7157	0.0604	112.7157	0.0249	112.7157	71.9587
	3	112.5376	0.0579	112.5376	0.0247	112.5376	71.6209
	4	112.6509	0.0766	112.6509	0.0319	112.6509	71.6807
	5	112.1491	0.0568	112.1491	0.0244	112.1491	72.3085
	6	112.9461	0.0679	112.9461	0.0245	112.9461	71.8191
	7	112.4412	0.0794	112.4412	0.0289	112.4412	72.0612
	8	112.2322	0.0801	112.2322	0.0269	112.2322	73.6233
	9	112.1184	0.0677	112.1184	0.0291	112.1184	71.8953
	10	112.4726	0.0787	112.4726	0.0269	112.4726	71.9910

and 'aver. time' denotes the average computer time in seconds used to solve the problem. For GloptiPoly3 [4], 'value' denotes the value of $F_{\mathscr{A}}(x)$ at the final iteration, 'time' denotes the total computer time in seconds used to solve the problem, and for GloptiPoly3, all the global solutions are found. From Table 4, we can see that these three algorithms can solve all the test problems with same optimal values. In terms of the computer time, we can see that GloptiPoly3 is most time consuming as it involves additional moment matrix rank condition

Table 5 Numerical results of Algorithm 3.1 and ADM [7] for large size problems

n	instance	Algorithm 3.1		ADM	
		value	time	best value	aver. time
40	1	799.2684	0.1259	799.2684	0.2440
	2	799.7059	0.0590	799.7059	0.2430
	3	800.1427	0.0588	800.1427	0.2926
	4	800.2174	0.0587	800.2174	0.2423
	5	799.6355	0.0585	799.6355	0.2428
	6	799.9708	0.0579	799.9708	0.2440
	7	799.8903	0.0588	799.8903	0.2424
	8	799.8400	0.0576	799.8400	0.2912
	9	800.5227	0.0580	800.5227	0.2921
	10	800.0133	0.0583	800.0133	0.2903
60	1	1799.7780	0.2597	1799.7780	1.3086
	2	1799.9830	0.2591	1799.9830	1.3140
	3	1800.1720	0.2603	1800.1720	1.0960
	4	1799.7120	0.2625	1799.7120	1.3206
	5	1799.7816	0.2608	1799.7816	1.1023
	6	1800.2263	0.2611	1800.2263	1.3252
	7	1800.2267	0.2611	1800.2267	1.3221
	8	1800.1004	0.2628	1800.1004	1.3224
	9	1800.3416	0.2611	1800.3416	1.1045
	10	1799.7447	0.2618	1799.7447	1.1048
80	1	3199.7798	1.6405	3199.7798	4.9238
	2	3199.9943	1.6386	3199.9943	4.9683
	3	3200.4058	1.6463	3200.4058	4.9511
	4	3200.0421	1.6422	3200.0421	4.9599
	5	3200.0610	1.6439	3200.0610	5.9472
	6	3199.8731	1.6448	3199.8731	4.9741
	7	3200.2557	1.6484	3200.2557	4.9761
	8	3199.5153	1.6472	3199.5153	4.9966
	9	3200.3246	1.6449	3200.3246	4.9830
	10	3200.1149	1.6480	3200.1149	4.9834
100	1	4999.6408	2.0561	4999.6408	10.4039
	2	4999.7645	2.0431	4999.7645	8.7602
	3	4999.6792	2.0447	4999.6792	10.5746
	4	5000.1390	2.0577	5000.1390	8.7747
	5	4999.8006	2.0579	4999.8006	8.8198
	6	5000.4760	2.0637	5000.4760	10.6242
	7	5000.0296	2.0612	5000.0296	8.8647
	8	4999.9768	2.0789	4999.9768	8.8659
	9	5000.0183	2.0620	5000.0183	8.8939
	10	5000.0126	2.0682	5000.0126	10.6267

check for extracting global maximizers while Algorithm 3.1 and the ADM method need less computer time. Table 5 shows that Algorithm 3.1 and the ADM method perform well for these large size test problems.

In addition, we tested Algorithm 3.2 and GloptiPoly3 [4] for some randomly generated tensor $\mathscr{B}$. Our numerical results reported in Table 6, where 'value' denotes the value of $G_{\mathscr{B}}(x, y)$ at the final iteration, and 'time' denotes the total computer time in seconds used to solve the problem. These results show that Algorithm 3.2 can produce global solutions for these test problems.

Table 6 Numerical results of Algorithm 3.2 and GloptiPoly3 [4] for small size problems

(n,m)	instance	Algorithm 3.2		GloptiPoly3	
		value	time	value	time
$(3,6)$	1	3.8066	0.5600	3.8066	4.1583
	2	1.3513	0.0070	1.3513	0.9007
	3	0.0739	0.0070	0.0739	1.1097
	4	4.1287	0.0073	4.1287	0.6044
	5	2.2616	0.0070	2.2616	0.6213
	6	2.1449	0.0068	2.1449	0.7088
	7	1.0516	0.0072	1.0516	0.7839
	8	1.5155	0.0071	1.5155	0.8563
	9	2.2108	0.0063	2.2108	0.6108
	10	2.3806	0.0066	2.3806	0.6888
$(3,9)$	1	3.6203	0.0075	3.6203	8.4651
	2	4.9392	0.0073	4.9392	6.4375
	3	4.6096	0.0065	4.6096	8.5439
	4	1.9927	0.0071	1.9927	8.4526
	5	2.7925	0.0073	2.7925	6.3307
	6	3.8466	0.0069	3.8466	7.3962
	7	5.8244	0.0103	5.8244	7.4653
	8	3.3822	0.0064	3.3822	6.3533
	9	0.4579	0.0069	0.4579	10.4191
	10	2.3821	0.0071	2.3821	7.3675
$(6,9)$	1	7.6415	0.0702	7.6415	72.7478
	2	2.8294	0.0067	2.8294	71.8820
	3	2.1927	0.0064	2.1927	65.3537
	4	5.0578	0.0063	5.0578	72.5861
	5	11.4202	0.0164	11.4202	73.2697
	6	14.9455	0.0066	14.9455	83.5311
	7	3.3829	0.0063	3.3829	63.3064
	8	12.1780	0.0064	12.1780	71.9839
	9	4.1406	0.0064	4.1406	76.4998
	10	2.5000	0.0113	2.5000	76.2772
$(6,12)$	1	8.1447	0.1898	8.1447	465.3645
	2	8.6276	0.0068	8.6276	524.4079
	3	12.8403	0.0064	12.8403	523.8832
	4	5.6678	0.0070	5.6678	458.4797
	5	5.5411	0.0172	5.5411	461.5319
	6	9.0097	0.0064	9.0097	460.4975
	7	7.4475	0.0080	7.4475	527.9661
	8	6.8387	0.0067	6.8387	526.6338
	9	14.2982	0.0088	14.2982	525.0086
	10	10.1592	0.0067	10.1592	525.4176

Acknowledgements The authors would like to thank the reviewers for their insightful comments which help to improve the presentation of the paper. The first author's work was supported by the National Natural Science Foundation of China (Grant No. 11471242) and the work of the second author was supported by the National Natural Science Foundation of China (Grant No. 11601261).

References

1. Chen B, He S, Li Z, Zhang S. Maximum block improvement and polynomial optimization. SIAM J Optim, 2012, 22: 87–107
2. He S, Li Z, Zhang S. Approximation algorithms for homogeneous polynomial optimization with quadratic constraints. Math Program, 2010, 125: 353–383
3. He S, Luo Z, Nie J, Zhang S. Semidefinite relaxation bounds for indefinite homogeneous quadratic optimization. SIAM J Optim, 2008, 19: 503–523
4. Henrion D, Lasserre J B, Loefberg J. GloptiPoly 3: Moments, optimization and semidefinite programming. Optim Methods Softw, 2009, 24: 761–779
5. Hu S, Li G, Qi L. A tensor analogy of Yuan's alternative theorem and polynomial optimization with sign structure. J Optim Theory Appl, 2016, 168: 446–474
6. Hu S, Li G, Qi L, Song Y. Finding the maximum eigenvalue of essentially nonnegative symmetric tensors via sum of squares programming. J Optim Theory Appl, 2013, 158: 717–738
7. Jiang B, Ma S, Zhang S. Alternating direction method of multipliers for real and complex polynomial optimization models. Optimization, 2014, 63: 883–898
8. Kolda T G, Mayo J R. Shifted power method for computing tensor eigenvalues. SIAM J Matrix Anal Appl, 2012, 32: 1095–1124
9. Lasserre J B. Global optimization with polynomials and the problem of moments. SIAM J Optim, 2001, 11: 796–817
10. Laurent M. Sum of squares, moment matrices and optimization over polynomials. In: Putinar M, Sullivant S, eds. Emerging Applications of Algebra Geometry. IMA Volumes in Mathematics and its Applications, Vol 149. Berlin: Springer, 2009, 157–270
11. Ling C, Nie J, Qi L, Ye Y. Bi-quadratic optimization over unit spheres and semidefinite programming relaxations. SIAM J Optim, 2009, 20: 1286–1310
12. Nie J. Sum of squares methods for minimizing polynomial functions over spheres and hypersurfaces. Front Math China, 2012, 7: 321–346
13. Qi L. Eigenvalues of a real supersymmetric tensor. J Symbolic Comput, 2005, 40: 1302–1324
14. Qi L, Teo K L. Multivariate polynomial minimization and its applications in signal processing. J Global Optim, 2003, 26: 419–433
15. So A M -C. Deterministic approximation algorithms for sphere contained homogeneous polynomial optimization problems. Mathe Program, 2011, 129: 357–382
16. Wang Y, Caccetta L, Zhou G. Convergence analysis of a block improvement method for polynomial optimization over unit spheres. Numer Linear Algebra Appl, 2015, 22: 1059–1076
17. Zhang L, Qi L. Linear convergence of an algorithm for computing the largest eigenvalue of a nonnegative tensor. Numer Linear Algebra Appl, 2012, 19: 830–841
18. Zhang L, Qi L, Xu Y. Finding the largest eigenvalue of an irreducible nonnegative tensor and linear convergence for weakly positive tensors. J Comput Math, 2012, 30: 24–33
19. Zhang X, Ling C, Qi L. Semidefinite relaxation bounds for bi-quadratic optimization problems with quadratic constraints. J Global Optim, 2010, 49: 293–311
20. Zhang X, Ling C, Qi L. The best rank-1 approximation of a symmetric tensor and related spherical optimization problems. SIAM J Matrix Anal Appl, 2012, 33: 806–821
21. Zhang X, Ling C, Qi L, Wu E X. The measure of diffusion skewness and kurtosis in magnetic resonance imaging. Pac J Optim, 2010, 6: 391–404
22. Zhang X, Qi L, Ye Y. The cubic spherical optimization problems. Math Comp, 2012, 81: 1513–1525
23. Zhou G, Caccetta L, Teo K, Wu S -Y. Nonnegative polynomial optimization over unit spheres and convex programming relaxations. SIAM J Optim, 2012, 22: 987–1008