SIAM J. MATRIX ANAL. APPL.
Vol. 35, No. 3, pp. 1155–1179

SEMIDEFINITE RELAXATIONS FOR BEST RANK-1 TENSOR APPROXIMATIONS*

JIAWANG NIE† AND LI WANG†

Abstract. This paper studies the problem of finding best rank-1 approximations for both symmetric and nonsymmetric tensors. For symmetric tensors, this is equivalent to optimizing homogeneous polynomials over unit spheres; for nonsymmetric tensors, this is equivalent to optimizing multiquadratic forms over multispheres. We propose semidefinite relaxations, based on sum of squares representations, to solve these polynomial optimization problems. Their special properties and structures are studied. In applications, the resulting semidefinite programs are often large scale. The recent Newton-CG augmented Lagrangian method by Zhao, Sun, and Toh [*SIAM J. Optim.*, 20 (2010), pp. 1737–1765] is suitable for solving these semidefinite relaxations. Extensive numerical experiments are presented to show that this approach is efficient in getting best rank-1 approximations.

Key words. form, polynomial, relaxation, rank-1 approximation, semidefinite program, sum of squares, tensor

AMS subject classifications. 15A18, 15A69, 90C22

DOI. 10.1137/130935112

1. Introduction.

Let m and $n_1, \ldots, n_m$ be positive integers. A tensor of order m and dimension $(n_1, \ldots, n_m)$ is an array $\mathcal{F}$ that is indexed by integer tuples $(i_1, \ldots, i_m)$ with $1 \le i_j \le n_j$ $(j = 1, \ldots, m)$, i.e.,

$$\mathcal{F} = (\mathcal{F}_{i_1,\ldots,i_m})_{1\le i_1\le n_1,\ldots,1\le i_m\le n_m}.$$

The space of all such tensors with real (resp., complex) entries is denoted as $\mathbb{R}^{n_1\times\cdots\times n_m}$ (resp., $\mathbb{C}^{n_1\times\cdots\times n_m}$). Tensors of order m are called m-tensors. When m equals 1 or 2, they are regular vectors or matrices. When $m = 3$ (resp., 4), they are called cubic (resp., quartic) tensors. A tensor $\mathcal{F} \in \mathbb{R}^{n_1\times\cdots\times n_m}$ is symmetric if $n_1 = \cdots = n_m$ and

$$\mathcal{F}_{i_1,\ldots,i_m} = \mathcal{F}_{j_1,\ldots,j_m} \quad \forall \quad (i_1, \ldots, i_m) \sim (j_1, \ldots, j_m),$$

where $\sim$ means that $(i_1, \ldots, i_m)$ is a permutation of $(j_1, \ldots, j_m)$. We define the norm of a tensor $\mathcal{F}$ as

$$\|\mathcal{F}\| = \left(\sum_{i_1=1}^{n_1} \cdots \sum_{i_m=1}^{n_m} |\mathcal{F}_{i_1,\ldots,i_m}|^2\right)^{1/2}. \tag{1.1}$$

For $m = 1$, $\|\mathcal{F}\|$ is the vector 2-norm, and for $m = 2$, $\|\mathcal{F}\|$ is the matrix Frobenius norm.

Every tensor can be expressed as a linear combination of outer products of vectors. For vectors $u^1 \in \mathbb{C}^{n_1}, \ldots, u^m \in \mathbb{C}^{n_m}$, their outer product $u^1 \otimes \cdots \otimes u^m$ is the tensor in $\mathbb{C}^{n_1\times\cdots\times n_m}$ such that for all $1 \le i_j \le n_j$ $(j = 1, \ldots, m)$

$$(u^1 \otimes \cdots \otimes u^m)_{i_1,\ldots,i_m} = (u^1)_{i_1} \cdots (u^m)_{i_m}.$$

*Received by the editors September 3, 2013; accepted for publication (in revised form) by T. Kolda May 23, 2014; published electronically August 28, 2014. The research was partially supported by NSF grant DMS-0844775.
http://www.siam.org/journals/simax/35-3/93511.html
†Department of Mathematics, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093 (njw@math.ucsd.edu, liw022@math.ucsd.edu).

For every tensor $\mathcal{F}$ of order m, there exist tuples $(u^{i,1},\dots,u^{i,m})$ $(i=1,\dots,r)$, with each $u^{i,j}\in\mathbb{C}^{n_j}$, such that

$$\mathcal{F}=\sum_{i=1}^{r}u^{i,1}\otimes\cdots\otimes u^{i,m}. \tag{1.2}$$

The smallest r in the above is called the rank of $\mathcal{F}$ and is denoted as $\operatorname{rank}\mathcal{F}$. When $\operatorname{rank}\mathcal{F}=r$, (1.2) is called a rank decomposition of $\mathcal{F}$, and we say that $\mathcal{F}$ is a rank-r tensor.

Tensor problems have wide applications in chemometrics, signal processing, and high order statistics [6]. For the theory and applications of tensors, we refer to Comon et al. [7], Kolda and Bader [15], and Landsberg [17]. When $m\geq 3$, determining ranks of tensors and computing rank decompositions are NP-hard (cf. Hillar and Lim [11]). We refer to Brachat et al. [1], Bernardi et al. [2], and Oeding and Ottaviani [25] for tensor decompositions. When $r>1$, the problem of finding best rank-r approximations may be ill-posed, because a best rank-r approximation might not exist, as discovered by De Silva and Lim [10]. However, a best rank-1 approximation always exists. It is also NP-hard to compute best rank-1 approximations. This paper studies best real rank-1 approximations for real tensors (i.e., tensor entries are real numbers). We begin with some reviews on this subject.

1.1. Nonsymmetric tensor approximations. Given a tensor $\mathcal{F}\in\mathbb{R}^{n_1\times\cdots\times n_m}$, we say that a tensor $\mathcal{B}$ is a best rank-1 approximation of $\mathcal{F}$ if it is a minimizer of the least squares problem

$$\min_{\mathcal{X}\in\mathbb{R}^{n_1\times\cdots\times n_m},\operatorname{rank}\mathcal{X}=1},\qquad \|\mathcal{F}-\mathcal{X}\|^2. \tag{1.3}$$

This is equivalent to a homogeneous polynomial optimization problem, as shown in [9]. For convenience of description, we define the homogeneous polynomial

$$F(x^1,\dots,x^m):=\sum_{1\leq i_1\leq n_1,\dots,1\leq i_m\leq n_m}\mathcal{F}_{i_1,\dots,i_m}(x^1)_{i_1}\cdots(x^m)_{i_m}, \tag{1.4}$$

which is in $x^1\in\mathbb{R}^{n_1},\dots,x^m\in\mathbb{R}^{n_m}$. Note that $F(x^1,\dots,x^m)$ is a multilinear form (a form is a homogeneous polynomial), since it is linear in each x^j. De Lathauwer, De Moor, and Vandewalle [9] proved the following result.

THEOREM 1.1 (see [9, Theorem 3.1]). *For a tensor $\mathcal{F}\in\mathbb{R}^{n_1\times\cdots\times n_m}$, the rank-1 approximation problem* (1.3) *is equivalent to the maximization problem*

$$\begin{cases}\max\limits_{x^1\in\mathbb{R}^{n_1},\dots,x^m\in\mathbb{R}^{n_m}}, & |F(x^1,\dots,x^m)|\\ \text{s.t.} & \|x^1\|=\cdots=\|x^m\|=1,\end{cases} \tag{1.5}$$

that is, a rank-1 tensor $\lambda\cdot(u^1\otimes\cdots\otimes u^m)$, with $\lambda\in\mathbb{R}$ and each $\|u^i\|=1$, is a best rank-1 approximation for $\mathcal{F}$ if and only if $(u^1,\dots,u^m)$ is a global maximizer of (1.5) *and $\lambda=F(u^1,\dots,u^m)$. Moreover, it also holds that*

$$\|\mathcal{F}-\lambda\cdot(u^1\otimes\cdots\otimes u^m)\|^2=\|\mathcal{F}\|^2-\lambda^2.$$

By Theorem 1.1, to find a best rank-1 approximation, it is enough to find a global maximizer of the multilinear optimization problem (1.5). There exist methods on finding rank-1 approximation, like the alternating least squares (ALS), truncated high

order singular value decomposition (HOSVD), higher order power method (HOPM), and quasi-Newton methods. We refer to Zhang and Golub [33], De Lathauwer, De Moor and Vandewalle [8, 9], Savas and Lim [29], and the references therein. An advantage of these methods is that they can be easily implemented. These kinds of methods typically generate a sequence that converges to a locally optimal rank-1 approximation or even just a stationary point. Even for the lucky cases that they get globally optimal rank-1 approximations, it is usually very difficult to verify the global optimality by these methods.

1.2. Symmetric tensor approximations. Let $\mathtt{S}^m(\mathbb{R}^n)$ be the space of real symmetric tensors of order m and in dimension n. Given $\mathcal{F} \in \mathtt{S}^m(\mathbb{R}^n)$, we say that $\mathcal{B}$ is a best rank-1 approximation of $\mathcal{F}$ if it is a minimizer of the optimization problem

$$\min_{\mathcal{X} \in \mathbb{R}^{n \times \cdots \times n},\, \operatorname{rank} \mathcal{X} = 1} \quad \|\mathcal{F} - \mathcal{X}\|^2. \tag{1.6}$$

When $\mathcal{F}$ is symmetric, Zhang, Ling, and Qi [35] showed that (1.6) has a global minimizer that belongs to $\mathtt{S}^m(\mathbb{R}^n)$, i.e., (1.6) has an optimizer that is a symmetric tensor. It is possible that a best rank-1 approximation of a symmetric tensor might not be symmetric. But there is always at least one global minimizer of (1.6) that is a symmetric rank-1 tensor. Therefore, for convenience, by best rank-1 approximation for symmetric tensors, we mean best symmetric rank-1 approximation.

A symmetric tensor in $\mathtt{S}^m(\mathbb{R}^n)$ is rank-1 if and only if it equals $\lambda \cdot (u \otimes \cdots \otimes u)$ for some $\lambda \in \mathbb{R}$ and $u \in \mathbb{R}^n$. For convenience, denote $u^{\otimes m} := u \otimes \cdots \otimes u$ (u is repeated m times). In the spirit of Theorem 1.1 and the work [35], (1.6) is equivalent to the optimization problem

$$\max_{x \in \mathbb{R}^n} \quad |f(x)| \quad \text{s.t.} \quad \|x\| = 1, \tag{1.7}$$

where $f(x) := F(x, \cdots, x)$. Therefore, if u is a global maximizer of (1.7) and $\lambda = f(u)$, then $\lambda \cdot u^{\otimes m}$ is a best rank-1 approximation of $\mathcal{F}$. Clearly, to solve (1.7), we need to solve two maximization problems:

$$\text{(I)} \quad \max_{x \in \mathbb{S}^{n-1}} f(x), \qquad \text{(II)} \quad \max_{x \in \mathbb{S}^{n-1}} -f(x), \tag{1.8}$$

where $\mathbb{S}^{n-1} := \{x \in \mathbb{R}^n : \|x\| = 1\}$ is the $n-1$ dimensional unit sphere. Suppose u^+, u^- are global maximizers of (I), (II) in (1.8), respectively. By Theorem 1.1, if $|f(u^+)| \geq |f(u^-)|$, $f(u^+) \cdot (u^+)^{\otimes m}$ is the best rank-1 approximation; otherwise, $f(u^-) \cdot (u^-)^{\otimes m}$ is the best.

For an introduction to symmetric tensors, we refer to Comon et al. [7]. Finding best rank-1 approximations for symmetric tensors is also NP-hard when $m \geq 3$. There exist methods for computing rank-1 approximations for symmetric tensors. When HOPM is directly applied, it is often unreliable for attaining a good symmetric rank-1 approximation, as pointed out in [9]. To get good symmetric rank-1 approximations, Kofidis and Regalia [14] proposed a symmetric higher order power method (SHOPM), and Zhang, Ling, and Qi [35] proposed a modified power method. These methods can be easily implemented. Like for nonsymmetric tensors, they often generate a locally optimal rank-1 approximation or even just a stationary point. Even for the lucky cases in which a globally optimal rank-1 approximation is found, these methods typically have difficulty verifying its global optimality. The problem (1.7) is related to extreme Z-eigenvalues of symmetric tensors. Recently, Hu, Huang, and Qi [12] proposed a method for computing extreme Z-eigenvalues for symmetric tensors of even orders. It solves a sequence of semidefinite relaxations based on sum of squares representations.

1.3. Contributions of the paper. In this paper, we propose a new approach for computing best rank-1 tensor approximations, i.e., using semidefinite program (SDP) relaxations. As we have seen, for nonsymmetric tensors, the problem is equivalent to optimizing a multilinear form over multispheres, i.e., (1.5); for symmetric tensors, it is equivalent to optimizing a homogeneous polynomial over the unit sphere, i.e., (1.7). Recently, there has been extensive work on solving polynomial optimization problems by using semidefinite relaxations and sum of squares representations, e.g., Lasserre [18], Parrilo and Sturmfels [26], and Nie and Wang [24]. These relaxations are often tight, and generally they are able to get global optimizers, which can be verified mathematically. We refer to Lasserre's book [19] and Laurent's survey [20] for an overview for the work in this area.

For a nonsymmetric tensor $\mathcal{F}$, to get a best rank-1 approximation is equivalent to solving the multilinear optimization problem (1.5). When $\mathcal{F}$ is symmetric, to get a best rank-1 approximation for $\mathcal{F}$, we need to solve the homogeneous optimization problem (1.7). We solve these polynomial optimization problems by using semidefinite relaxations, based on sum of squares representations. In applications, the resulting semidefinite programs are often large scale. The traditional interior point methods for semidefinite programs are generally too expensive for solving them. The recent Newton-CG augmented Lagrangian method by Zhao, Sun, and Toh [32] is very efficient for solving such big semidefinite programs, as shown in [24]. In the paper, we use this method to compute best rank-1 approximations for tensors.

The paper is organized as follows. In section 2, we show how to find best rank-1 approximations by using semidefinite relaxations and we propose numerical algorithms based on them. Their special structures and properties are also studied. In section 3, we present extensive numerical experiments to show the efficiency of these semidefinite relaxations. In section 4, we discuss this approach and future work.

Notation. The symbol $\mathbb{N}$ (resp., $\mathbb{R}$, $\mathbb{C}$) denotes the set of nonnegative integers (resp., real numbers, complex numbers). For two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots n_m}$, define their inner product as

$$\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{1 \le i_1 \le n_1, \ldots, 1 \le i_m \le n_m} \mathcal{X}_{i_1,\ldots,i_m} \mathcal{Y}_{i_1,\ldots,i_m}.$$

For $t \in \mathbb{R}$, $\lceil t \rceil$ denotes the smallest integer that is not smaller than t. For integer $n > 0$, $[n]$ denotes the set $\{1, \cdots, n\}$. For $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, $|\alpha| := \alpha_1 + \cdots + \alpha_n$. Denote $\mathbb{N}^n_m = \{\alpha \in \mathbb{N}^n : |\alpha| = m\}$. For $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, x^α denotes $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. Let $\mathbb{R}[x]$ be the ring of polynomials with real coefficients in x. A polynomial p is said to be sum of squares (SOS) if $p = p_1^2 + \cdots + p_k^2$ for some $p_1, \ldots, p_k \in \mathbb{R}[x]$. The symbol $\Sigma_{n,m}$ denotes the cone of SOS forms in $(x_1, \ldots, x_n)$ and of degree m. For a finite set T, $|T|$ denotes its cardinality. The symbol e_i denotes the ith unit vector, i.e., e_i is the vector whose ith entry equals one and all others equal zero. For a matrix A, A^T denotes its transpose. For a symmetric matrix W, $W \succeq 0$(resp., $\succ 0$) means that W is positive semidefinite (resp., definite). For any vector $u \in \mathbb{R}^N$, $\|u\| = \sqrt{u^T u}$ denotes the standard Euclidean 2-norm.

2. Semidefinite relaxations and algorithms. To find best rank-1 tensor approximations is equivalent to solving some homogeneous polynomial optimization problems with sphere constraints. In this section, we show how to solve them by using semidefinite relaxations based on sum of squares representations, and we study their properties.

2.1. Symmetric tensors of even orders. Let $\mathcal{F} \in \mathtt{S}^m(\mathbb{R}^n)$ with $m = 2d$ even. To get a best rank-1 approximation of $\mathcal{F}$, we need to solve (1.7), i.e., to maximize $|f(x)|$ over the unit sphere. For this purpose, we need to find the maximum and minimum of $f(x)$ over $\mathbb{S}^{n-1}$.

First, we consider the maximization problem:

$$f_{\max} := \max \quad f(x) \quad \text{s.t.} \quad x^T x = 1. \tag{2.1}$$

The form f is in $x := (x_1, \ldots, x_n)$, determined by the tensor $\mathcal{F}$ as

$$f(x) = \sum_{1 \le i_1, \ldots, i_m \le n} \mathcal{F}_{i_1,\ldots,i_m} x_{i_1} \cdots x_{i_m}. \tag{2.2}$$

Let $[x^d]$ be the monomial vector:

$$[x^d] := \begin{bmatrix} x_1^d & x_1^{d-1}x_2 & \cdots & x_1^{d-1}x_n & \cdots & x_n^d \end{bmatrix}^T.$$

Its length is $\binom{n+d-1}{d}$. The outer product $[x^d][x^d]^T$ is a symmetric matrix with entries being monomials of degree m. Let A_α be symmetric matrices such that

$$[x^d][x^d]^T = \sum_{\alpha \in \mathbb{N}^n_m} A_\alpha x^\alpha.$$

For $y \in \mathbb{R}^{\mathbb{N}^n_m}$, define the matrix-valued function $M(y)$ as

$$M(y) := \sum_{\alpha \in \mathbb{N}^n_m} A_\alpha y_\alpha.$$

Then $M(y)$ is a linear pencil in y (i.e., $M(y)$ is a linear matrix-valued function in y). Let f_α, g_α be the coefficients such that

$$f := \sum_{\alpha \in \mathbb{N}^n_m} f_\alpha x^\alpha, \quad g := (x^T x)^d = \sum_{\alpha \in \mathbb{N}^n_m} g_\alpha x^\alpha.$$

For $y \in \mathbb{R}^{\mathbb{N}^n_m}$, define

$$\langle f, y \rangle := \sum_{\alpha \in \mathbb{N}^n_m} f_\alpha y_\alpha, \quad \langle g, y \rangle := \sum_{\alpha \in \mathbb{N}^n_m} g_\alpha y_\alpha.$$

A semidefinite relaxation of (2.1) is (cf. [23, 24])

$$f^{\text{sdp}}_{\max} := \max_{y \in \mathbb{R}^{\mathbb{N}^n_m}} \quad \langle f, y \rangle \quad \text{s.t.} \quad M(y) \succeq 0, \quad \langle g, y \rangle = 1. \tag{2.3}$$

It can be shown that the dual of the above is

$$\min \quad \gamma \quad \text{s.t.} \quad \gamma g - f \in \Sigma_{n,m}. \tag{2.4}$$

In the above, $\Sigma_{n,m}$ denotes the cone of SOS forms of degree m and in variables $x_1, \ldots, x_n$. Clearly, $f^{\text{sdp}}_{\max} \ge f_{\max}$. When $f^{\text{sdp}}_{\max} = f_{\max}$, we say that the semidefinite relaxation (2.3) is tight.

The dual problem (2.4) is a linear optimization with SOS type constraints. Recently, Hu, Huang, and Qi [12] proposed an SOS relaxation method for computing extreme Z-eigenvalues for symmetric tensors of even orders. Since not every nonnegative form is SOS, they consider a sequence of nesting SOS relaxations. The problem (2.4) is equivalent to the lowest order relaxation in [12, section 4]. In practice, the relaxation (2.4) is often tight, and it is frequently used because of its simplicity. The SOS relaxation (2.4) was also proposed in [23] for optimizing forms over unit spheres. Its approximation quality was also analyzed there.

The feasible set of (2.3) is compact, because

$$\mathrm{Trace}(M(y)) \le \langle g, y\rangle = 1.$$

So (2.3) always has a maximizer, say, y^*. If $\mathrm{rank}\, M(y^*) = 1$, then (2.3) is a tight relaxation. In such case, there exists $v^+ \in \mathbb{S}^{n-1}$ such that $y^* = [(v^+)^m]$, because of the structure of $M(y)$. Then, it holds that

$$f(v^+) = \langle f, y^*\rangle = f_{\max}^{\mathrm{sdp}} \ge f_{\max}.$$

This implies that v^+ is a global maximizer of (2.1). The vector v^+ can be chosen numerically as follows. Let $s \in [n]$ be the index such that

$$y^*_{2de_s} = \max_{1\le i\le n} y^*_{2de_i}.$$

Then choose v^+ as the normalization:

$$\hat{u} = (y^*_{(2d-1)e_s+e_1}, \ldots, y^*_{(2d-1)e_s+e_n}), \quad v^+ = \hat{u}/\|\hat{u}\|. \tag{2.5}$$

If $\mathrm{rank}(M(y^*)) > 1$ but $M(y^*)$ satisfies a further rank condition, then (2.3) is also tight (cf. [24, section 2.2]). In computations, no matter whether (2.3) is tight or not, the vector v^+ selected as in (2.5) can be used as an approximation for a maximizer of (2.1).

Second, we consider the minimization problem:

$$f_{\min} := \min \quad f(x) \quad \text{s.t.} \quad x^T x = 1. \tag{2.6}$$

Similarly, a semidefinite relaxation of (2.6) is

$$f_{\min}^{\mathrm{sdp}} := \min_{z\in\mathbb{R}^{\mathbb{N}^n_m}} \quad \langle f, z\rangle \quad \text{s.t.} \quad M(z) \succeq 0, \quad \langle g, z\rangle = 1. \tag{2.7}$$

Its dual optimization problem can be shown to be

$$\max \quad \eta \quad \text{s.t.} \quad f - \eta g \in \Sigma_{n,m}. \tag{2.8}$$

Let z^* be a minimizer of (2.7). Similarly, if $\mathrm{rank}(M(z^*)) = 1$, then (2.7) is a tight relaxation. In such case, a global minimizer v^- can be found as follows: let $t \in [n]$ be the index such that

$$z^*_{2de_t} = \max_{1\le i\le n} z^*_{2de_i},$$

then choose v^- as the normalization:

$$\tilde{u} = (z^*_{(2d-1)e_t+e_1}, \ldots, z^*_{(2d-1)e_t+e_n}), \quad v^- = \tilde{u}/\|\tilde{u}\|. \tag{2.9}$$

When $\operatorname{rank} M(z^*) > 1$, (2.7) might not be a tight relaxation. In computations, the vector v^- can be used as an approximation for a minimizer of (2.6).

Combining the above and inspired by Theorem 1.1, we get the following algorithm.

ALGORITHM 2.1. *Rank-1 approximations for even symmetric tensors.*

Input: *A symmetric tensor $\mathcal{F} \in \mathbb{S}^m(\mathbb{R}^n)$ with an even order $m = 2d$.*

Output: *A rank-1 tensor $\lambda \cdot u^{\otimes m}$ with $\lambda \in \mathbb{R}$ and $u \in \mathbb{S}^{n-1}$.*

Procedure:

Step 1. *Solve the semidefinite relaxation* (2.3) *and get an optimizer y^*.*

Step 2. *Choose v^+ as in* (2.5)*. If $\operatorname{rank} M(y^*) = 1$, let $u^+ = v^+$; otherwise, apply a nonlinear optimization method to get a better solution u^+ of* (2.1)*, by using v^+ as a starting point. Let $\lambda^+ = f(u^+)$.*

Step 3. *Solve the semidefinite relaxation* (2.7) *and get an optimizer z^*.*

Step 4. *Choose v^- as in* (2.9)*. If $\operatorname{rank} M(z^*) = 1$, let $u^- = v^-$; otherwise, apply a nonlinear optimization method to get a better solution u^- of* (2.6)*, by using v^- as a starting point. Let $\lambda^- = f(u^-)$.*

Step 5. *If $|\lambda^+| \geq |\lambda^-|$, output $(\lambda, u) := (\lambda^+, u^+)$; otherwise, output $(\lambda, u) := (\lambda^-, u^-)$.*

Remark. In Algorithm 2.1, if $\operatorname{rank} M(y^*) = \operatorname{rank} M(z^*) = 1$, then the output $\lambda \cdot u^{\otimes m}$ is a best rank-1 approximation of $\mathcal{F}$. If this rank condition is not satisfied, then $\lambda \cdot u^{\otimes m}$ might not be best. The approximation qualities of semidefinite relaxations (2.3) and (2.7) are analyzed in [23, section 2].

When $m = 2$ or $(n, m) = (3, 4)$, the semidefinite relaxations (2.3) and (2.7) are always tight. This is because every bivariate, or ternary quartic, nonnegative form is always SOS (cf. [28]). For other cases of (n, m), this is not necessarily true. However, this does not occur very often in applications. In our numerical experiments, the semidefinite relaxations (2.3) and (2.7) are often tight.

We want to know when the ranks of $M(y^*)$ and $M(z^*)$ are equal to one. Clearly, when they have rank one, the relaxations (2.3) and (2.7) must be tight. Interestingly, the reverse is often true, although it might be false sometimes (for very few cases). This fact was observed in the field of polynomial optimization. However, we are not able to find suitable references for explaining this fact. Here, we give a natural geometric interpretation for this phenomena, for lack of suitable references. Let $\partial\Sigma_{n,m}$ be the boundary of the cone $\Sigma_{n,m}$.

THEOREM 2.2. *Let $f_{\max}, f_{\max}^{\mathrm{sdp}}, f_{\min}, f_{\min}^{\mathrm{sdp}}, y^*, z^*$ be as above.*

(i) *Suppose $f_{\max} = f_{\max}^{\mathrm{sdp}}$. If $f_{\max} \cdot g - f$ is a smooth point of $\partial\Sigma_{n,m}$, then $\operatorname{rank} M(y^*) = 1$.*

(ii) *Suppose $f_{\min} = f_{\min}^{\mathrm{sdp}}$. If $f - f_{\min} \cdot g$ is a smooth point of $\partial\Sigma_{n,m}$, then $\operatorname{rank} M(z^*) = 1$.*

Proof. (i) Let μ be the uniform probability measure on the unit sphere $\mathbb{S}^{n-1}$, and let $y^\mu := \int [x^m] d\mu \in \mathbb{R}^{\mathbb{N}^n_m}$. Then, we can show that $M(y^\mu) \succ 0$ and $\langle g, y^\mu \rangle = 1$. This shows that y^μ is an interior point of (2.3). So, the strong duality holds, that is, the optimal values of (2.3) and (2.4) are equal, and (2.4) achieves the optimal value $f_{\max}^{\mathrm{sdp}}$. The form $\sigma := f_{\max} \cdot g - f$ belongs to $\Sigma_{n,m}$, because $f_{\max} = f_{\max}^{\mathrm{sdp}}$. Let u be a maximizer of f on $\mathbb{S}^{n-1}$. Then, $\sigma(u) = 0$. So, σ lies on the boundary $\partial\Sigma_{n,m}$. Let $\hat{y} = [u^m]$. Denote by $\mathbb{R}[x]_m$ the space of all forms in x and of degree m. Then

$$\mathcal{H}_{\hat{y}} := \{p \in \mathbb{R}[x]_m : \langle p, \hat{y} \rangle = 0\}$$

is a supporting hyperplane of $\Sigma_{n,m}$ through σ, because

$$\langle p, \hat{y} \rangle = p(u) \geq 0 \quad \forall\, p \in \Sigma_{n,m}$$

and $\langle \sigma, \hat{y} \rangle = \sigma(u) = 0$. Because $M(y^*) \succeq 0$ and $\langle p, y^* \rangle \geq 0$ for all $p \in \Sigma_{n,m}$, the hyperplane

$$\mathcal{H}_{y^*} := \{p \in \mathbb{R}[x]_m : \langle p, y^* \rangle = 0\}$$

also supports $\Sigma_{n,m}$ through σ. Since σ is a smooth point of $\partial\Sigma_{n,m}$, there is a unique supporting hyperplane $\mathcal{H}$ through σ. So, $y^* = \hat{y} = [u^m]$. This implies that $M(y^*) = M([u^m]) = [u^d][u^d]^T$ has rank one, where $d = m/2$.

(ii) The proof is the same as for (i). ∎

By Theorem 2.2, when semidefinite relaxations (2.3) and (2.7) are tight, we often have $\operatorname{rank} M(y^*) = \operatorname{rank} M(z^*) = 1$. This fact was observed in our numerical experiments. The reason is that the set of nonsmooth points of the boundary $\partial\Sigma_{n,m}$ has a strictly smaller dimension than $\partial\Sigma_{n,m}$.

2.2. Symmetric tensors of odd orders. Let $\mathcal{F} \in \mathbb{S}^m(\mathbb{R}^n)$ with $m = 2d-1$ odd. To get a best rank-1 approximation of $\mathcal{F}$, we need to solve the optimization problem (1.7). Since the form f, defined as in (2.2), has the odd degree m, (1.7) is equivalent to (2.1). We cannot directly apply a semidefinite relaxation to solve (2.1). For this purpose, we use a trick that is introduced in [23, section 4.2].

Let $f_{\max}, f_{\min}$ be as in (2.1), (2.6), respectively. Since f is an odd form,

$$f_{\max} = -f_{\min} \geq 0.$$

Let x_{n+1} be a new variable, in addition to $x = (x_1, \ldots, x_n)$. Let

$$\tilde{x} := (x_1, \ldots, x_n, x_{n+1}), \quad \tilde{f}(\tilde{x}) := f(x)x_{n+1}.$$

Then $\tilde{f}(\tilde{x})$ is a form of even degree $2d$. Consider the optimization problem:

$$\tilde{f}_{\max} := \max_{\tilde{x} \in \mathbb{R}^{n+1}} \quad \tilde{f}(\tilde{x}) \quad \text{s.t.} \quad \|\tilde{x}\| = 1. \tag{2.10}$$

As shown in [23, section 4.2], it holds that

$$f_{\max} = \sqrt{2d-1}\big(1 - \frac{1}{2d}\big)^{-d} \tilde{f}_{\max}.$$

Since $\tilde{f}$ is an even form, the semidefinite relaxation for (2.10) is

$$\tilde{f}_{\max}^{\text{sdp}} := \max_{y \in \mathbb{N}^{n+1}_{2d}} \quad \langle \tilde{f}, y \rangle \quad \text{s.t.} \quad M(y) \succeq 0, \quad \langle g, y \rangle = 1. \tag{2.11}$$

The vector y is indexed by $(n+1)$-dimensional integer vectors.

Let y^* be a maximizer of (2.11), which always exists because the feasible set of (2.11) is compact. If $\operatorname{rank} M(y^*) = 1$, then (2.11) is a tight relaxation, and a global maximizer v^+ of (2.10) can be chosen as in (2.5). (Note that the n in (2.5) should be replaced by $n+1$.) Write v^+ as

$$v^+ = (\hat{v}, \hat{t}), \quad \|\hat{v}\|^2 + \hat{t}^2 = 1.$$

If $\hat{v} = 0$ or $\hat{t} = 0$, then $\tilde{f}_{\max} = f_{\max} = 0$ and the zero tensor is the best rank-1 approximation of $\mathcal{F}$. So, we consider the general case $0 < |\hat{t}| < 1$. Note that $\operatorname{sign}(\hat{t}) \cdot \hat{v}$ is a global maximizer of f on the sphere $\|x\|_2^2 = 1 - \hat{t}^2$. Let

$$\hat{u} = \operatorname{sign}(\hat{t}) \cdot \hat{v} / \sqrt{1 - \hat{t}^2}. \tag{2.12}$$

Then $\hat{u}$ is a global maximizer of f on $\mathbb{S}^{n-1}$. When $\operatorname{rank} M(y^*) > 1$, the above $\hat{u}$ might not be a global maximizer, but it can be used as an approximation for a maximizer.

Combining the above, we get the following algorithm.

ALGORITHM 2.3. *Rank-1 approximations for odd symmetric tensors.*

Input: *A symmetric tensor $\mathcal{F} \in \mathtt{S}^m(\mathbb{R}^n)$ with an odd order $m = 2d-1$.*

Output: *A rank-1 symmetric tensor $\lambda \cdot u^{\otimes m}$ with $\lambda \in \mathbb{R}$ and $u \in \mathbb{S}^{n-1}$.*

Procedure:

Step 1. *Solve the semidefinite relaxation* (2.11) *and get an optimizer y^*.*

Step 2. *Choose v^+ as in* (2.5) *and $\hat{u}$ as in* (2.12). *(The n in* (2.5) *should be replaced by $n+1$.)*

Step 3. *If $\operatorname{rank} M(y^*) = 1$, let $u = \hat{u}$; otherwise, apply a nonlinear optimization method to get a better solution u of* (2.1)*, by using $\hat{u}$ as a starting point. Let $\lambda = f(u)$. Output (λ, u).*

Remark. In Algorithm 2.3, if $\operatorname{rank} M(y^*) = 1$, then the output $\lambda \cdot u^{\otimes m}$ is a best rank-1 approximation of the tensor $\mathcal{F}$. If $\operatorname{rank} M(y^*) > 1$, then $\lambda \cdot u^{\otimes m}$ is not necessarily the best. The approximation quality of the semidefinite relaxation (2.11) is analyzed in [23, section 4]. In our numerical experiments, we often have $\operatorname{rank} M(y^*) = 1$. A similar version of Theorem 2.2 is true for Algorithm 2.3. We omit it for simplicity.

2.3. Nonsymmetric tensors. Let $\mathcal{F} \in \mathbb{R}^{n_1 \times \cdots \times n_m}$ be a nonsymmetric tensor of order m. Let $F(x^1, \ldots, x^m)$ be the multilinear form defined as in (1.4). To get a best rank-1 approximation of $\mathcal{F}$ is equivalent to solving the multilinear optimization problem (1.5). Here we show how to solve it by using semidefinite relaxations.

Without loss of generality, assume $n_m = \max_j n_j$. Since $F(x^1, \ldots, x^m)$ is linear in x^m, we can write it as

$$F(x^1, \ldots, x^m) = \sum_{j=1}^{n_m} (x^m)_j F_j(x^1, \ldots, x^{m-1}),$$

where each $F_j(x^1, \ldots, x^{m-1})$ is also a multilinear form. Let $m' = m-1$, and

$$F^{sq} := \sum_{j=1}^{n_m} F_j(x^1, \ldots, x^{m'})^2.$$

By the Cauchy–Schwartz inequality, it holds that

$$|F(x^1, \ldots, x^m)| \le F^{sq}(x^1, \ldots, x^{m'})^{1/2} \|x^m\|.$$

The equality occurs in the above if and only if x^m is proportional to

$$\big(F_1(x^1, \ldots, x^{m'}), \ldots, F_{n_m}(x^1, \ldots, x^{m'})\big).$$

Therefore, (1.5) is equivalent to

$$\left\{ \begin{array}{lll} F_{\max} : & = \max\limits_{x^1, \ldots, x^{m'}}, & F^{sq}(x^1, \ldots, x^{m'}) \\ & \text{s.t.} & \|x^1\| = \cdots = \|x^{m'}\| = 1. \end{array} \right. \tag{2.13}$$

Now we present the semidefinite relaxations for solving (2.13). The outer product

$$\mathcal{K}(x) := x^1 \otimes \cdots \otimes x^{m'}$$

is a vector of length $n_1 \cdots n_{m'}$. Denote

$$\Omega := \{(\imath, \jmath) : \imath, \jmath \in [n_1] \times \cdots \times [n_{m'}]\}.$$

Expand the outer product of $\mathcal{K}(x)$ as

$$\mathcal{K}(x)\mathcal{K}(x)^T = \sum_{(\imath,\jmath)\in\Omega} B_{\imath,\jmath} (x^1)_{\imath_1}(x^1)_{\jmath_1} \cdots (x^{m'})_{\imath_{m'}}(x^{m'})_{\jmath_{m'}},$$

where each $B_{\imath,\jmath}$ is a constant symmetric matrix. For $w \in \mathbb{R}^{\Omega}$, define

$$K(w) := \sum_{(\imath,\jmath)\in\Omega} B_{\imath,\jmath} w_{\imath,\jmath}.$$

Clearly, $K(w)$ is a linear pencil in $w \in \mathbb{R}^{\Omega}$. Write

$$F^{sq} = \sum_{(\imath,\jmath)\in\Omega} G_{\imath,\jmath} (x^1)_{\imath_1}(x^1)_{\jmath_1} \cdots (x^{m'})_{\imath_{m'}}(x^{m'})_{\jmath_{m'}},$$

$$h := \|x^1\|^2 \cdots \|x^{m'}\|^2 = \sum_{(\imath,\jmath)\in\Omega} h_{\imath,\jmath} (x^1)_{\imath_1}(x^1)_{\jmath_1} \cdots (x^{m'})_{\imath_{m'}}(x^{m'})_{\jmath_{m'}}.$$

For $w \in \mathbb{R}^{\Omega}$, we denote

$$\langle F^{sq}, w\rangle := \sum_{(\imath,\jmath)\in\Omega} G_{\imath,\jmath}\, w_{\imath,\jmath}, \qquad \langle h, w\rangle := \sum_{(\imath,\jmath)\in\Omega} h_{\imath,\jmath}\, w_{\imath,\jmath}.$$

A semidefinite relaxation of (2.13) is

$$F_{\max}^{\mathrm{sdp}} := \max \quad \langle F^{sq}, w\rangle \quad \text{s.t.} \quad K(w) \succeq 0, \quad \langle h, w\rangle = 1. \tag{2.14}$$

Define $\Sigma_{n_1,\ldots,n_{m'}}$ to be the cone as

$$\Sigma_{n_1,\ldots,n_{m'}} = \left\{ L \;\middle|\; \begin{array}{c} L = L_1^2 + \cdots + L_k^2 \text{ where each } L_i \\ \text{is a multilinear form in } (x^1, \ldots, x^{m'}) \end{array} \right\}.$$

It can be shown that the dual problem of (2.14) is

$$\min \quad \gamma \quad \text{s.t.} \quad \gamma h - F^{sq} \in \Sigma_{n_1,\ldots,n_{m'}}. \tag{2.15}$$

Clearly, we always have $F_{\max}^{\mathrm{sdp}} \geq F_{\max}$. When the equality occurs, we say that (2.14) is a tight relaxation.

The feasible set of (2.14) is compact, because

$$\mathrm{Trace}(K(w)) = \langle h, w\rangle = 1.$$

Let w^* be a maximizer of (2.14). Like for the case of symmetric tensors, if $\mathrm{rank}\, K(w^*) = 1$, then (2.14) is tight, and there exist vectors $v^1, \ldots, v^{m'}$ of unit length such that $w^* = (v^1(v^1)^T) \otimes \cdots \otimes (v^{m'}(v^{m'})^T)$ and $(v^1, \ldots, v^{m'})$ is a maximizer of (2.13). They can be constructed as follows. Let $\ell \in [n_1] \times \cdots \times [n_{m'}]$ be the index such that

$$w^*_{\ell,\ell} = \max_{(\imath,\imath)\in\Omega} w^*_{\imath,\imath}.$$

Then choose v^j $(j = 1, \ldots, m')$ as

$$\hat{v}^j = \Big(w^*_{\hat{\ell}_1,\ell}, w^*_{\hat{\ell}_2,\ell}, \ldots, w^*_{\hat{\ell}_{n_j},\ell}\Big), \qquad v^j = \hat{v}^j/\|\hat{v}^j\|, \tag{2.16}$$

where $\hat{\ell}_k = \ell + (k - \ell_j) \cdot e_j$ for each $k \in [n_j]$. When $\operatorname{rank} K(w^*) > 1$, the tuple $(v^1, \ldots, v^{m'})$ as in (2.16) might not be a global maximizer of (2.13). But it can be used as an approximation for a maximizer of (2.13).

Combining the above, we get the following algorithm.

ALGORITHM 2.4. *Rank-1 approximations for nonsymmetric tensors.*
Input: *A nonsymmetric tensor* $\mathcal{F} \in \mathbb{R}^{n_1 \times \ldots \times n_m}$.
Output: *A rank-1 tensor* $\lambda \cdot (u^1 \otimes \cdots \otimes u^m)$ *with* $\lambda \in \mathbb{R}$ *and each* $u^j \in \mathbb{S}^{n_j - 1}$.
Procedure:
Step 1. *Solve the semidefinite relaxation* (2.14) *and get a maximizer* w^*.
Step 2. *Choose* $(v^1, \ldots, v^{m'})$ *as in* (2.16). *Then let*

$$\hat{v}^m := (F_1(v^1, \ldots, v^{m'}), \ldots, F_{n_m}(v^1, \ldots, v^{m'})),$$

and $v^m := \hat{v}^m/\|\hat{v}^m\|$.
Step 3. *If* $\operatorname{rank} K(w^*) = 1$, *let* $u^i = v^i$ *for* $i = 1, \ldots, m$; *otherwise, apply a nonlinear optimization method to get a better solution* $(u^1, \ldots, u^m)$ *of* (1.5), *by using* $(v^1, \ldots, v^m)$ *as a starting point.*
Step 4. *Let* $\lambda = F(u^1, \ldots, u^m)$, *and output* $(\lambda, u^1, \ldots, u^m)$.

Remark. In Algorithm 2.4, if $\operatorname{rank} K(w^*) = 1$, then the output $\lambda \cdot u^1 \otimes \cdots \otimes u^m$ is a best rank-1 approximation of $\mathcal{F}$. If $\operatorname{rank} K(w^*) > 1$, then $\lambda \cdot u^1 \otimes \cdots \otimes u^m$ is not necessarily the best. The approximation quality of the semidefinite relaxation (2.14) is analyzed in [23, section 3].

We want to know when $\operatorname{rank} K(w^*) = 1$. Clearly, for this to be true, the relaxation (2.14) must be tight, i.e., $F_{\max} = F^{\rm sdp}_{\max}$. Like for the symmetric case, the reverse is also often true, as shown in the following theorem. Let $\partial\Sigma_{n_1,\ldots,n_{m'}}$ be the boundary of the cone $\Sigma_{n_1,\ldots,n_{m'}}$.

THEOREM 2.5. *Let* $F_{\max}, F^{\rm sdp}_{\max}, w^*$ *be as above. Suppose* $F_{\max} = F^{\rm sdp}_{\max}$. *If* $F_{\max} \cdot h - F^{sq}$ *is a smooth point of* $\partial\Sigma_{n_1,\ldots,n_{m'}}$, *then* $\operatorname{rank} K(w^*) = 1$.

Proof. This can be proved in the same way as for Theorem 2.2. Let $(u^1, \ldots, u^{m'})$ be a global maximizer of (2.13). Let $\hat{w} \in \mathbb{R}^\Omega$ be the vector such that

$$\hat{w}_{\imath,\jmath} = (u^1)_{\imath_1}(u^1)_{\jmath_1} \cdots (u^m)_{\imath_m}(u^m)_{\jmath_m} \quad \forall\, (\imath, \jmath) \in \Omega.$$

The key point is the observation that $\langle p, \hat{w}\rangle = 0$ defines a unique supporting hyperplane of $\Sigma_{n_1,\ldots,n_{m'}}$ through $F_{\max} \cdot h - F^{sq}$, when it is a smooth point of the boundary $\partial\Sigma_{n_1,\ldots,n_{m'}}$. The proof proceeds as for Theorem 2.2. ∎

3. Numerical experiments. In this section, we report numerical experiments of using semidefinite relaxations to find best rank-1 tensor approximations. The computations are implemented in MATLAB 7.10 on a Dell Linux desktop with 8 GB memory and Intel 2.8 GHz CPU. In applications, the resulting semidefinite programs are often large scale. Interior point methods are not very suitable for solving such big semidefinite programs. We use the software `SDPNAL` [31] by Zhao, Sun, and Toh, which is based on the Newton-CG augmented Lagrangian method [32]. In our computations, the default values of the parameters in `SDPNAL` are used. In Algorithms 2.1, 2.3, and 2.4, if the matrices $M(y^*), M(z^*), K(w^*)$ do not have rank one, we apply the nonlinear

program solver `fmincon` in the MATLAB Optimization Toolbox to improve the rank-1 approximations obtained from semidefinite relaxations. Our numerical experiments show that these algorithms are often able to get best rank-1 approximations and `SDPNAL` is efficient in solving such large-scale semidefinite relaxations.

We report the consumed computer time in the format `hr:mn:sc` with `hr` (resp., `mn, sc`) standing for the consumed hours (resp., minutes, seconds). In our presented tables, `min` (resp., `med, max`) stands for the minimum (resp., median, maximum) of quantities like time and errors.

In our computations, the rank of a matrix A is numerically measured as follows: if the singular values of A are $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_t > 0$, then $\operatorname{rank}(A)$ is set to be the smallest r such that $\sigma_{r+1}/\sigma_r < 10^{-6}$. In the display of our computational results, only four decimal digits are shown.

3.1. Symmetric tensor examples. In this subsection, we report numerical experiments for symmetric tensors. We apply Algorithm 2.1 for even symmetric tensors, and we apply Algorithm 2.3 for odd symmetric tensors. In Algorithm 2.1, if

$$\operatorname{rank} M(y^*) = \operatorname{rank} M(z^*) = 1, \tag{3.1}$$

then the output tensor $\lambda \cdot u^{\otimes m}$ is a best rank-1 approximation. If $\operatorname{rank} M(y^*) > 1$ or $\operatorname{rank} M(z^*) > 1$, $\lambda \cdot u^{\otimes m}$ is not guaranteed to be a best rank-1 approximation. However, the quantity

$$f_{\rm ubd} := \max\{|f^{\rm sdp}_{\max}|, |f^{\rm sdp}_{\min}|\}$$

is always an upper bound of $|f(x)|$ on $\mathbb{S}^{n-1}$. No matter whether (3.1) holds, the error

$$\texttt{aprxerr} := \big||f(u)| - f_{\rm ubd}\big| / \max\{1, f_{\rm ubd}\} \tag{3.2}$$

is a measure of the approximation quality of $\lambda \cdot u^{\otimes m}$. When Algorithm 2.3 is applied for odd symmetric tensors, $f_{\rm ubd}$ and `aprxerr` are defined similarly. As in Qi [27], we define the *best rank-1 approximation ratio* of a tensor $\mathcal{F} \in \mathbb{S}^m(\mathbb{R}^n)$ as

$$\rho(\mathcal{F}) := \max_{\mathcal{X} \in \mathbb{S}^m(\mathbb{R}^n), \operatorname{rank}\mathcal{X}=1} \frac{|\langle \mathcal{F}, \mathcal{X}\rangle|}{\|\mathcal{F}\|\|\mathcal{X}\|}. \tag{3.3}$$

If $\lambda \cdot u^{\otimes m}$, with $\|u\| = 1$ and $\lambda = f(u)$, is a best rank-1 approximation of $\mathcal{F}$, then $\rho(\mathcal{F}) = |\lambda|/\|\mathcal{F}\|$. Estimates for $\rho(\mathcal{F})$ are given in Qi [27].

Example 3.1 (see [9, Example 2]). Consider the tensor $\mathcal{F} \in \mathbb{S}^3(\mathbb{R}^2)$ with entries

$$\mathcal{F}_{111} = 1.5578, \quad \mathcal{F}_{222} = 1.1226, \quad \mathcal{F}_{112} = -2.4443, \quad \mathcal{F}_{221} = -1.0982.$$

When Algorithm 2.3 is applied, we get the rank-1 tensor $\lambda \cdot u^{\otimes 3}$ with

$$\lambda = 3.1155, \quad u = (0.9264, -0.3764).$$

It takes about 0.2 second. The computed matrix $M(y^*)$ has rank one. So, we know $\lambda \cdot u^{\otimes 3}$ is a best rank-1 approximation. The error `aprxerr` = 7.3e-9, the ratio $\rho(\mathcal{F}) = 0.6203$, the residual $\|\mathcal{F} - \lambda \cdot u^{\otimes 3}\| = 3.9399$, and $\|\mathcal{F}\| = 5.0228$.

Example 3.2 (see [16, Example 3.6], [35, Example 4.2]). Consider the tensor $\mathcal{F} \in \mathbb{S}^3(\mathbb{R}^3)$ with entries

$$\begin{aligned}&\mathcal{F}_{111} = -0.1281, \mathcal{F}_{112} = 0.0516, \mathcal{F}_{113} = -0.0954, \mathcal{F}_{122} = -0.1958, \mathcal{F}_{123} = -0.1790,\\ &\mathcal{F}_{133} = -0.2676, \mathcal{F}_{222} = 0.3251, \mathcal{F}_{223} = \ \ 0.2513, \mathcal{F}_{233} = \ \ 0.1773, \mathcal{F}_{333} = \ \ 0.0338.\end{aligned}$$

When Algorithm 2.3 is applied, we get the rank-1 tensor $\lambda \cdot u^{\otimes 3}$ with

$$\lambda = 0.8730, \quad u = (-0.3921, 0.7249, 0.5664).$$

It takes about 0.2 second. The computed matrix $M(y^*)$ has rank one, so $\lambda \cdot u^{\otimes 3}$ is a best rank-1 approximation. The error `aprxerr`=1.2e-7, the ratio $\rho(\mathcal{F}) = 0.8890$, the residual $\|\mathcal{F} - \lambda \cdot u^{\otimes 3}\| = 0.4498$, and $\|\mathcal{F}\| = 0.9820$.

Example 3.3 (see [27, Example 2]). Consider the tensor $\mathcal{F} \in \mathbb{S}^3(\mathbb{R}^3)$ with entries

$$\mathcal{F}_{111} = 0.0517, \mathcal{F}_{112} = 0.3579, \mathcal{F}_{113} = 0.5298, \mathcal{F}_{122} = 0.7544, \mathcal{F}_{123} = 0.2156,$$
$$\mathcal{F}_{133} = 0.3612, \mathcal{F}_{222} = 0.3943, \mathcal{F}_{223} = 0.0146, \mathcal{F}_{233} = 0.6718, \mathcal{F}_{333} = 0.9723.$$

When Algorithm 2.3 is applied, we get the rank-1 tensor $\lambda \cdot u^{\otimes 3}$ with

$$\lambda = 2.1110, \quad u = (0.5204, 0.5113, 0.6839).$$

It takes about 0.2 second. Since the computed matrix $M(y^*)$ has rank one, $\lambda \cdot u^{\otimes 3}$ is a best rank-1 approximation. The error `aprxerr`=6.9e-8, the ratio $\rho(\mathcal{F}) = 0.8574$, the residual $\|\mathcal{F} - \lambda \cdot u^{\otimes 3}\| = 1.2672$, and $\|\mathcal{F}\| = 2.4621$.

Example 3.4 (see [16, Example 3.5], [35, Example 4.1]). Consider the tensor $\mathcal{F} \in \mathbb{S}^4(\mathbb{R}^3)$ with entries

$$\mathcal{F}_{1111} = 0.2883,\ \mathcal{F}_{1112} = -0.0031, \mathcal{F}_{1113} = 0.1973, \mathcal{F}_{1122} = -0.2485, \mathcal{F}_{1123} = -0.2939,$$
$$\mathcal{F}_{1133} = 0.3847,\ \mathcal{F}_{1222} = \ \ 0.2972, \mathcal{F}_{1223} = 0.1862, \mathcal{F}_{1233} = \ \ 0.0919, \mathcal{F}_{1333} = -0.3619,$$
$$\mathcal{F}_{2222} = 0.1241,\ \mathcal{F}_{2223} = -0.3420,\ \mathcal{F}_{2233} = 0.2127,\ \mathcal{F}_{2333} = 0.2727, \mathcal{F}_{3333} = -0.3054.$$

Applying Algorithm 2.1, we get $\lambda^+ \cdot (u^+)^{\otimes m}$ and $\lambda^- \cdot (u^-)^{\otimes m}$ with

$$\lambda^+ = 0.8893, \quad u^+ = (-0.6672, -0.2470, 0.7027),$$

$$\lambda^- = -1.0954, \quad u^- = (-0.5915, 0.7467, 0.3043).$$

It takes about 0.3 second. Since $|\lambda^+| < |\lambda^-|$, the output rank-1 tensor is $\lambda \cdot u^{\otimes 4}$ with $\lambda = \lambda^-, u = u^-$. The computed matrices $M(y^*), M(z^*)$ both have rank one, so $\lambda \cdot u^{\otimes 4}$ is a best rank-1 approximation. The error `aprxerr`=2.8e-7, the ratio $\rho(\mathcal{F}) = 0.4863$, the residual $\|\mathcal{F} - \lambda \cdot u^{\otimes 4}\| = 1.9683$, and $\|\mathcal{F}\| = 2.2525$.

Example 3.5. Consider the tensor $\mathcal{F} \in \mathbb{S}^3(\mathbb{R}^n)$ with entries

$$(\mathcal{F})_{i_1,i_2,i_3} = \frac{(-1)^{i_1}}{i_1} + \frac{(-1)^{i_2}}{i_2} + \frac{(-1)^{i_3}}{i_3}.$$

For $n = 5$, we apply Algorithm 2.3 and get the rank-1 tensor $\lambda \cdot u^{\otimes 3}$ with

$$\lambda = 9.9779, \qquad u = -(0.7313, 0.1375, 0.4674, 0.2365, 0.4146).$$

It takes about 0.3 second. The error `aprxerr`=1.4e-7. Since the computed matrix $M(y^*)$ has rank one, we know $\lambda \cdot u^{\otimes 3}$ is a best rank-1 approximation. The ratio $\rho(\mathcal{F}) = 0.8813$, the residual $\|\mathcal{F} - \lambda \cdot u^{\otimes 3}\| = 5.3498$, and $\|\mathcal{F}\| = 11.3216$.

For a range of values of n from 10 to 55, we apply Algorithm 2.3 to find rank-1 approximations. All the computed matrices $M(y^*)$ have rank one, so we get best rank-1 approximations for all of them. The computational results are listed in Table 1. For $n = 5, 10, 15, 20, 25, 30$, Algorithm 2.3 takes less than 1 minute; for $n = 35, 40, 45,$

TABLE 1
Computational results for Example 3.5.

n	Time	λ	aprxerr	$\rho(\mathcal{F}_n)$	n	Time	λ	aprxerr	$\rho(\mathcal{F}_n)$
10	0:00:02	17.80	3.5e-9	0.80	15	0:00:03	26.48	5.9e-8	0.79
20	0:00:06	34.16	9.8e-8	0.77	25	0:00:11	42.51	1.8e-7	0.77
30	0:00:30	50.14	2.5e-8	0.75	35	0:01:02	58.33	5.1e-7	0.75
40	0:04:05	65.93	1.8e-8	0.74	45	0:12:07	74.02	2.5e-8	0.74
50	0:32:45	81.59	7.7e-8	0.73	55	0:42:00	89.62	9.2e-8	0.73

TABLE 2
Computational results for Example 3.6.

n	Time	λ	aprxerr	$\rho(\mathcal{F})$	n	Time	λ	aprxerr	$\rho(\mathcal{F})$
5	0:00:01	2.357e+1	1.4e-7	0.81	10	0:00:02	7.707e+1	4.8e-9	0.73
15	0:00:05	1.651e+2	2.3e-10	0.71	20	0:00:09	2.830e+2	2.1e-7	0.69
25	0:00:18	4.353e+2	5.5e-8	0.68	30	0:00:22	6.175e+2	2.9e-9	0.68
35	0:01:37	8.342e+2	9.5e-10	0.67	40	0:04:41	1.081e+3	1.3e-7	0.67
45	0:09:07	1.362e+3	3.7e-9	0.67	50	0:23:32	1.673e+3	1.7e-8	0.67
55	0:45:48	2.018e+3	3.1e-8	0.67	60	1:04:26	2.393e+3	1.0e-7	0.66

it takes a couple of minutes; for $n = 50, 55$, it takes about half an hour. The errors aprxerr are all very tiny. The best rank-1 approximation ratios are around three-quarters.

Example 3.6. Consider the tensor $\mathcal{F} \in \mathbb{S}^4(\mathbb{R}^n)$ given as

$$(\mathcal{F})_{i_1,\cdots,i_4} = \arctan\Big((-1)^{i_1}\frac{i_1}{n}\Big) + \cdots + \arctan\Big((-1)^{i_4}\frac{i_4}{n}\Big).$$

For $n = 5$, applying Algorithm 2.1, we get two rank-1 tensors $\lambda^+ \cdot (u^+)^{\otimes m}$ and $\lambda^- \cdot (u^-)^{\otimes m}$ with

$$\begin{aligned}\lambda^+ = \ \ 13.0779, \quad u^+ &= (0.3174, 0.5881, 0.1566, 0.7260, 0.0418),\\ \lambda^- = -23.5740, \quad u^- &= (0.4403, 0.2382, 0.5602, 0.1354, 0.6459).\end{aligned}$$

It takes about 0.6 second. Since $|\lambda^+| < |\lambda^-|$, the output rank-1 tensor is $\lambda \cdot u^{\otimes 4}$ with $\lambda = -23.5740$ and $u = u^-$. The error aprxerr=1.4e-7. The computed matrices $M(y^*), M(z^*)$ both have rank one, so $\lambda \cdot u^{\otimes 4}$ is a best rank-1 approximation. The ratio $\rho(\mathcal{F}) = 0.8135$, the residual $\|\mathcal{F} - \lambda \cdot u^{\otimes 4}\| = 16.8501$, and $\|\mathcal{F}\| = 28.9769$.

For a range of values of n from 5 to 60, we apply Algorithm 2.1 to find rank-1 approximations. The computational results are listed in Table 2. All the computed matrices $M(y^*), M(z^*)$ have rank one, so we get best rank-1 approximations for all of them. For $n = 5, 10, 15, 20, 25, 30$, Algorithm 2.1 takes less than 1 minute; for $n = 35, 40, 45, 50, 55$, it takes less than 1 hour, and for $n = 60$, it takes around 1 hour. The errors aprxerr are all very tiny. The best rank-1 approximation ratios are near two-thirds.

Example 3.7. Consider the tensor $\mathcal{F} \in \mathbb{S}^5(\mathbb{R}^n)$ given as

$$(\mathcal{F})_{i_1,\cdots,i_5} = (-1)^{i_1}\ln(i_1) + \cdots + (-1)^{i_5}\ln(i_5).$$

For $n = 5$, we apply Algorithm 2.3 and get the rank-1 tensor $\lambda \cdot u^{\otimes 5}$ with

$$\lambda = 110.0083, \qquad u = -(0.3900, 0.2785, 0.5668, 0.1669, 0.6490).$$

TABLE 3
Computational results for Example 3.7.

n	Time	λ	`aprxerr`	$\rho(\mathcal{F})$	n	Time	λ	`aprxerr`	$\rho(\mathcal{F})$
5	0:00:01	1.100e+2	1.2e-7	0.77	13	0:00:17	1.790e+3	1.2e-6	0.69
6	0:00:02	2.096e+2	2.6e-7	0.81	14	0:00:29	2.283e+3	1.1e-6	0.71
7	0:00:03	2.975e+2	5.5e-8	0.74	15	0:01:18	2.697e+3	1.5e-8	0.69
8	0:00:06	4.706e+2	6.7e-8	0.77	16	0:02:12	3.327e+3	8.2e-7	0.70
9	0:00:10	6.192e+2	1.4e-7	0.72	17	0:02:47	3.854e+3	4.9e-7	0.68
10	0:00:22	8.833e+2	1.8e-8	0.74	18	0:07:54	4.637e+3	1.4e-7	0.69
11	0:00:23	1.107e+3	2.4e-7	0.70	19	0:12:46	5.289e+3	7.2e-7	0.68
12	0:00:13	1.478e+3	3.8e-7	0.72	20	0:22:30	6.237e+3	2.6e-7	0.69

It takes about 0.5 second. The computed matrix $M(y^*)$ has rank one, so $\lambda \cdot u^{\otimes 5}$ is a best rank-1 approximation. The error `aprxerr`=3.0e-7. The ratio $\rho(\mathcal{F}) = 0.7709$, the residual $\|\mathcal{F} - \lambda \cdot u^{\otimes 5}\| = 90.8818$ and $\|\mathcal{F}\| = 142.6931$.

For a range of values of n from 5 to 20, we apply Algorithm 2.3 to find rank-1 approximations. The computational results are shown in Table 3. All the computed matrices $M(y^*)$ have rank one, so we get best rank-1 approximations for all of them. The errors `aprxerr` are all tiny. The best rank-1 approximation ratios $\rho(\mathcal{F})$ are close to 0.70. Algorithm 2.3 takes from a few seconds to a couple of minutes to get best rank-1 approximations.

Example 3.8. Consider the tensor $\mathcal{F} \in \mathbf{S}^6(\mathbb{R}^3)$ with

$$\mathcal{F}_{111111} = 2, \mathcal{F}_{111122} = 1/3, \mathcal{F}_{111133} = 2/5, \mathcal{F}_{112222} = 1/3, \mathcal{F}_{112233} = 1/6,$$
$$\mathcal{F}_{113333} = 2/5, \mathcal{F}_{222222} = 2, \mathcal{F}_{222233} = 2/5, \mathcal{F}_{223333} = 2/5, \mathcal{F}_{333333} = 1,$$

and $\mathcal{F}_{i_1,\ldots,i_6} = 0$ if $(i_1, \ldots, i_6)$ is not a permutation of an index in the above. We can verify that

$$f(x) = 2\|x\|^6 - M(x),$$

where $M(x) = x_1^4x_2^2 + x_1^2x_2^4 + x_3^6 - 3x_1^2x_2^2x_3^2$ is the Motzkin polynomial, which is nonnegative everywhere but not SOS (cf. [28]). Since $0 \le M(x) \le \|x\|^6$, we can show that $f_{\max} = 2$, $f_{\min} = 1$. Applying Algorithm 2.1, we get

$$f_{\max}^{\text{sdp}} = 2.0046, \quad v^+ = (0, 1, 0), \quad f(v^+) = 2,$$
$$f_{\min}^{\text{sdp}} = 1.0000, \quad v^- = (0, 0, 1), \quad f(v^-) = 1.$$

The matrix $M(z^*)$ has rank one, so $\lambda^- = f(v^-)$ and $u^- = v^-$. The matrix $M(y^*)$ has rank 7, which is bigger than one, so we apply `fmincon` to improve v^+ but get the same point $u^+ = v^+$; let $\lambda^+ = f(u^+)$. Since $|\lambda^-| < |\lambda^+|$, the output rank-1 tensor is $\lambda \cdot u^{\otimes 6}$ with

$$\lambda = 2.0000, \qquad u = (0, 1, 0).$$

Since $f_{\max} = \lambda = f(u)$, we know $\lambda \cdot u^{\otimes 6}$ is a best rank-1 approximation by Theorem 1.1. The best rank-1 approximation ratio $\rho(\mathcal{F}) = 0.4046$.

Example 3.9 (random examples). We explore the performance of Algorithms 2.1 and 2.3 for finding best rank-1 approximations for randomly generated symmetric tensors. We generate $\mathcal{F} \in \mathbf{S}^m(\mathbb{R}^n)$ with each entry being a random variable obeying Gaussian distribution (by `randn` in MATLAB). For each generated $\mathcal{F}$, the semidefinite relaxations (2.3), (2.7), and (2.11) can be expressed in the standard dual form

TABLE 4
Computational results in Example 3.9. *Here, m is the tensor order, n is the tensor dimension, N is the length of matrices, and M is the number of variables in the semidefinite relaxations.*

(n,m)	(N,M)	Time (min,med,max)			`aprxerr` (min,med,max)
(10,3)	(66,1000)	0:00:01	0:00:01	0:00:03	(7.9e-9, 4.5e-8, 2.9e-6)
(20,3)	(231,10625)	0:00:03	0:00:08	0:00:13	(2.4e-9, 3.6e-7, 4.3e-6)
(30,3)	(496,46375)	0:01:14	0:01:29	0:02:01	(9.1e-9, 7.4e-7, 1.4e-5)
(40,3)	(861,135750)	0:06:32	0:10:04	0:13:09	(1.3e-9, 4.6e-6, 2.3e-3)
(50,3)	(1326,316250)	0:12:39	0:13:34	0:14:01	(3.2e-9, 1.3e-6, 2.0e-3)
(15,4)	(120,3060)	0:00:01	0:00:03	0:00:04	(4.0e-9, 1.1e-7, 1.3e-6)
(20,4)	(210,8854)	0:00:52	0:01:09	0:01:25	(1.2e-8, 1.8e-7, 6.3e-3)
(25,4)	(325,20475)	0:00:30	0:00:35	0:00:56	(4.7e-9, 1.3e-7, 1.0e-5)
(30,4)	(465,40919)	0:06:03	0:07:36	0:09:31	(1.2e-8, 1.1e-6, 9.6e-4)
(35,4)	(630,73815)	0:02:46	0:04:57	0:06:54	(4.1e-8, 1.6e-7, 7.4e-3)
(10,5)	(286,8007)	0:00:08	0:00:14	0:00:17	(4.3e-8, 4.1e-7, 4.1e-6)
(15,5)	(816,54263)	0:03:46	0:03:58	0:07:24	(4.4e-8, 2.5e-6, 1.1e-3)
(20,5)	(1771,230229)	0:28:14	0:30:30	0:43:27	(4.7e-7, 3.7e-6, 5.7e-6)
(10,6)	(220,5004)	0:00:11	0:00:14	0:00:20	(1.3e-7, 6.4e-7, 3.5e-2)
(15,6)	(680,38759)	0:03:14	0:04:19	0:04:53	(4.8e-8, 2.5e-3, 4.9e-2)
(20,6)	(1540,177099)	0:39:28	0:45:39	0:54:59	(2.8e-8, 6.6e-5, 1.0e-2)

$$\begin{cases} \max & b_1\mu_1 + \cdots + b_M\mu_M \\ \text{s.t.} & F_0 - \sum_{i=1}^M \mu_i F_i \succeq 0, \end{cases} \tag{3.4}$$

where F_i are constant symmetric matrices (cf. [30]). In (3.4), let N be the length of matrices F_i, and M be the number of variables. For pairs (n,m), if the semidefinite relaxation matrix length $N < 1000$, we test 50 instances of $\mathcal{F}$ randomly; otherwise if $N > 1000$, we test 10 instances of $\mathcal{F}$ randomly. For a range of values of (n,m), the computational results are shown in Table 4.

From Table 4, we can observe that Algorithms 2.1 and 2.3 generally produce accurate best rank-1 approximations in a short time. For some very big problems, like 3-tensors of dimension 40, or 4-tensors of dimension 35, we are able to get accurate best rank-1 approximations within a reasonable time. For most instances, we are able to get best rank-1 approximations, because the computed matrices $M(y^*), M(z^*)$ have rank one. For a few instances, their ranks are bigger than one, and the errors `aprxerr` are a bit relatively large, like in the order of 10^{-3} or 10^{-2}. This is probably because the semidefinite relaxations are not very tight.

Example 3.10. Here we explore the performance of SHOPM in Kofidis and Regalia [14, Algorithm 2], which is widely used in getting rank-1 approximations for symmetric tensors. SHOPM can be easily implemented. Generally, SHOPM generates a sequence of rank-1 tensors $\mu_k(v^k)^{\otimes m}$ with each $\|v^k\| = 1$ and $\mu_k = f(v^k)$. It is usually hard to check whether $\mu_k(v^k)^{\otimes m}$ converges to a best rank-1 approximation. Since semidefinite relaxations often get best rank-1 approximations, we can use them to check convergence of SHOPM. We write a straightforward MATLAB code to implement SHOPM and use the truncated HOSVD to generate starting points (cf. [9]), which is commonly used in applications. We terminate iterations of SHOPM if either $|\mu_{k+1} - \mu_k| < 10^{-8}$ or it runs over 1000 iterations. The tensor $\mathcal{F} \in \mathbb{S}^m(\mathbb{R}^n)$ we used is given as

$$(\mathcal{F})_{i_1,\cdots,i_m} = \sin\big(i_1 + \cdots + i_m\big).$$

Let $\lambda \cdot u^{\otimes m}$ be the rank-1 approximation generated by Algorithm 2.1 (for even orders) or Algorithm 2.3 (for odd orders), and let $\mu \cdot v^{\otimes m}$ be the rank-1 approximation generated by SHOPM. Their qualities are measured by the residuals

TABLE 5
Computational results for Example 3.10 *with* $m = 3$.

n	$\lvert\lambda\rvert$	$\lvert\mu\rvert$	$\mathrm{RES}_{\mathtt{sdp}}$	$\mathrm{RES}_{\mathtt{shopm}}$
10	12.12	3.01	18.79	22.16
15	22.07	13.74	34.65	38.72
20	32.98	26.99	53.96	57.19
25	44.46	33.33	76.39	81.86
30	61.23	13.03	98.75	115.46
35	74.68	74.68	125.94	125.94
40	92.39	53.88	153.18	170.58

TABLE 6
Computational results for Example 3.10 *with* $m = 4$.

n	$\lvert\lambda\rvert$	$\lvert\mu\rvert$	$\mathrm{RES}_{\mathtt{sdp}}$	$\mathrm{RES}_{\mathtt{shopm}}$
10	27.27	1.18	65.25	70.70
15	61.42	50.43	146.77	150.89
25	158.22	122.02	412.65	424.76
30	241.65	208.81	588.73	601.16
35	313.30	155.74	807.56	852.09
40	414.38	129.91	1052.75	1123.89

$$\mathrm{RES}_{\mathtt{sdp}} = \|\mathcal{F} - \lambda \cdot u^{\otimes m}\|, \qquad \mathrm{RES}_{\mathtt{shopm}} = \|\mathcal{F} - \mu \cdot v^{\otimes m}\|.$$

For $m = 3, 4$ and a range of n (we choose multiples of 5), the computational results are presented in Tables 5 and 6. For cleanness of comparison, only two decimal digits are presented. For all the cases, the computed matrices $M(y^*)$ and $M(z^*)$ are rank-1, so Algorithms 2.1 and 2.3 found best rank-1 approximations. As we can see, for such cases, SHOPM did not get the best, except $(n, m) = (35, 3)$. The computational results for the case $(n, m) = (20, 4)$ are not shown, because the software `SDPNAL` experiences numerical troubles and the semidefinite relaxations cannot be solved accurately. As for the consumed time, Algorithms 2.1 and 2.3 take up to a few minutes, while SHOPM takes up to tens of seconds. Generally, SHOPM takes less time, but it might not find best rank-1 approximations. On the other hand, the residuals $\mathrm{RES}_{\mathtt{shopm}}$, generated by SHOPM, are not significantly bigger than the residuals $\mathrm{RES}_{\mathtt{sdp}}$, generated by semidefinite relaxations. This shows that SHOPM performs reasonably well in getting good rank-1 approximations, although it might not get the best.

3.2. Nonsymmetric tensor examples. In this subsection, we report numerical results for nonsymmetric tensors. In Algorithm 2.4, if $\operatorname{rank} K(w^*) = 1$, the output $\lambda \cdot u^1 \otimes \cdots \otimes u^m$ is a best rank-1 approximation of $\mathcal{F}$. If $\operatorname{rank} K(w^*) > 1$, $\lambda \cdot u^1 \otimes \cdots \otimes u^m$ might not be the best. However, the quantity

$$F_{\mathrm{ubd}} := \sqrt{\left|F^{\mathrm{sdp}}_{\max}\right|}$$

is always an upper bound of $|F(x^1, \ldots, x^m)|$ on $\mathbb{S}^{n_1-1} \times \cdots \times \mathbb{S}^{n_m-1}$. Like in (3.2), we can measure the quality of $\lambda \cdot u^1 \otimes \cdots \otimes u^m$ by the error

$$\mathtt{aprxerr} := \left||F(u^1, \cdots, u^m)| - F_{\mathrm{ubd}}\right| / \max\{1, F_{\mathrm{ubd}}\}. \tag{3.5}$$

Like the symmetric case, we define the *best rank*-1 *approximation ratio* of a tensor $\mathcal{F} \in \mathbb{R}^{n_1 \times \cdots \times n_m}$ as (cf. Qi [27])

$$\rho(\mathcal{F}) := \max_{\mathcal{X}\in\mathbb{R}^{n_1\times\cdots\times n_m}, \operatorname{rank}\mathcal{X}=1} \frac{|\langle\mathcal{F},\mathcal{X}\rangle|}{\|\mathcal{F}\|\|\mathcal{X}\|}. \tag{3.6}$$

Clearly, if $\lambda\cdot u^1\otimes\cdots\otimes u^m$, with each $\|u^i\|=1$ and $\lambda = F(u^1,\ldots,u^m)$, is a best rank-1 approximation of $\mathcal{F}$, then $\rho(\mathcal{F}) = |\lambda|/\|\mathcal{F}\|$.

Example 3.11 (see [9, Example 3]). Consider the tensor $\mathcal{F}\in\mathbb{R}^{2\times2\times2\times2}$ with

$$\mathcal{F}_{1111} = 25.1, \quad \mathcal{F}_{1212} = 25.6, \quad \mathcal{F}_{2121} = 24.8, \quad \mathcal{F}_{2222} = 23,$$

and the resting entries are zeros. Applying Algorithm 2.4, we get the rank-1 tensor $\lambda\cdot u^1\otimes u^2\otimes u^3\otimes u^4$ with

$$\lambda = 25.6000, \quad u^1 = (1,0), \quad u^2 = (0,1), \quad u^3 = (1,0), \quad u^4 = (0,1).$$

It takes about 0.3 second. The matrix $K(w^*)$ has rank one, so $\lambda\cdot u^1\otimes u^2\otimes u^3\otimes u^4$ is a best rank-1 approximation. The error `aprxerr` =8.9e-10, the ratio $\rho(\mathcal{F}) = 0.5194$, the residual $\|\mathcal{F}-\lambda\cdot u^1\otimes u^2\otimes u^3\otimes u^4\| = 42.1195$, and $\|\mathcal{F}\| = 49.2890$.

Example 3.12 (see [27, Example 1]). Consider the tensor $\mathcal{F}\in\mathbb{R}^{3\times3\times3}$ with

$$\begin{aligned}
&\mathcal{F}_{111} = 0.4333, \mathcal{F}_{121} = 0.4278, \mathcal{F}_{131} = 0.4140, \mathcal{F}_{211} = 0.8154, \mathcal{F}_{221} = 0.0199,\\
&\mathcal{F}_{231} = 0.5598, \mathcal{F}_{311} = 0.0643, \mathcal{F}_{321} = 0.3815, \mathcal{F}_{331} = 0.8834, \mathcal{F}_{112} = 0.4866,\\
&\mathcal{F}_{122} = 0.8087, \mathcal{F}_{132} = 0.2073, \mathcal{F}_{212} = 0.7641, \mathcal{F}_{222} = 0.9924, \mathcal{F}_{232} = 0.8752,\\
&\mathcal{F}_{312} = 0.6708, \mathcal{F}_{322} = 0.8296, \mathcal{F}_{332} = 0.1325, \mathcal{F}_{113} = 0.3871, \mathcal{F}_{123} = 0.0769,\\
&\mathcal{F}_{133} = 0.3151, \mathcal{F}_{213} = 0.1355, \mathcal{F}_{223} = 0.7727, \mathcal{F}_{233} = 0.4089, \mathcal{F}_{313} = 0.9715,\\
&\mathcal{F}_{323} = 0.7726, \mathcal{F}_{333} = 0.5526.
\end{aligned}$$

Applying Algorithm 2.4, we get the rank-1 tensor $\lambda\cdot u^1\otimes u^2\otimes u^3$ with

$$\begin{aligned}
&\lambda = 2.8167, \qquad u^1 = (0.4281, 0.6557, 0.6220),\\
&u^2 = (0.5706, 0.6467, 0.5062), \quad u^3 = (0.4500, 0.7094, 0.5424).
\end{aligned}$$

It takes less than 1 second. The matrix $K(w^*)$ has rank one, so $\lambda\cdot u^1\otimes u^2\otimes u^3$ is a best rank-1 approximation. The error `aprxerr` = 3.9e-8, the ratio $\rho(\mathcal{F}) = 0.9017$, the residual $\|\mathcal{F}-\lambda\cdot u^1\otimes u^2\otimes u^3\| = 1.3510$, and $\|\mathcal{F}\| = 3.1239$.

Example 3.13 (see [13, section 4.1]). Consider the tensor $\mathcal{F}\in\mathbb{R}^{3\times3\times3}$ with

$$\begin{aligned}
&\mathcal{F}_{111} = \ 0.0072, \mathcal{F}_{121} = -0.4413, \mathcal{F}_{131} = \ 0.1941, \mathcal{F}_{211} = -0.4413, \mathcal{F}_{221} = \ 0.0940,\\
&\mathcal{F}_{231} = \ 0.5901, \mathcal{F}_{311} = \ 0.1941, \mathcal{F}_{321} = -0.4099, \mathcal{F}_{331} = -0.1012, \mathcal{F}_{112} = -0.4413,\\
&\mathcal{F}_{122} = \ 0.0940, \mathcal{F}_{132} = -0.4099, \mathcal{F}_{212} = \ 0.0940, \mathcal{F}_{222} = \ 0.2183, \mathcal{F}_{232} = \ 0.2950,\\
&\mathcal{F}_{312} = \ 0.5901, \mathcal{F}_{322} = \ 0.2950, \mathcal{F}_{332} = \ 0.2229, \mathcal{F}_{113} = \ 0.1941, \mathcal{F}_{123} = \ 0.5901,\\
&\mathcal{F}_{133} = -0.1012, \mathcal{F}_{213} = -0.4099, \mathcal{F}_{223} = \ 0.2950, \mathcal{F}_{233} = \ 0.2229, \mathcal{F}_{313} = -0.1012,\\
&\mathcal{F}_{323} = \ 0.2229, \mathcal{F}_{333} = -0.4891.
\end{aligned}$$

We apply Algorithm 2.4 and get an upper bound $F_{\text{ubd}} = 1.0000$. The computed matrix $K(w^*)$ has rank three, so we use `fmincon` to improve the solution and get the rank-1 tensor $\lambda\cdot u^1\otimes u^2\otimes u^3$ with

$$\begin{aligned}
&\lambda = 1.0000, \qquad u^1 = (0.7955, 0.2491, 0.5524),\\
&u^2 = (-0.0050, 0.9142, -0.4051), \quad u^3 = (-0.6060, 0.3195, 0.7285).
\end{aligned}$$

TABLE 7
Computational results for Example 3.14.

n	Time	λ	aprxerr	$\rho(\mathcal{F})$	n	Time	λ	aprxerr	$\rho(\mathcal{F})$
5	0:00:01	6.100	3.1e-9	0.77	10	0:00:02	14.79	5.8e-9	0.66
15	0:00:04	25.48	2.8e-8	0.62	20	0:00:20	33.70	1.2e-9	0.53
25	0:01:16	46.80	1.2e-8	0.53	30	0:02:57	64.91	8.2e-9	0.56
35	0:13:09	80.77	6.5e-10	0.55	40	0:27:07	95.09	4.5e-5	0.53

It takes less than 1 second. Since $\lambda = F(u^1, u^2, u^3) = F_{\text{ubd}}$, we know $\lambda \cdot u^1 \otimes u^2 \otimes u^3$ is a best rank-1 approximation by Theorem 1.1. The error aprxerr $= 6.0\text{e-}9$, the ratio $\rho(\mathcal{F}) = 0.5773$, the residual $\|\mathcal{F} - \lambda \cdot u^1 \otimes u^2 \otimes u^3\| = 1.4143$, and $\|\mathcal{F}\| = 1.7321$.

Example 3.14. Consider the tensor $\mathcal{F} \in \mathbb{R}^{n\times n\times n}$ given as

$$(\mathcal{F})_{i_1,i_2,i_3} = \cos\big(i_1 + 2i_2 + 3i_3\big).$$

For $n = 5$, we apply Algorithm 2.4 and get the rank-1 tensor $\lambda \cdot u^1 \otimes u^2 \otimes u^3$ with $\lambda = 6.0996$ and

$$\begin{aligned} u^1 &= (-0.4296, -0.5611, -0.1767, 0.3701, 0.5766),\\ u^2 &= (0.6210, -0.2956, -0.3750, 0.6077, -0.1308),\\ u^3 &= (-0.4528, 0.4590, -0.4561, 0.4441, -0.4231). \end{aligned}$$

It takes around 0.3 seconds. The matrix $K(w^*)$ has rank one, so $\lambda \cdot u^1 \otimes u^2 \otimes u^3$ is a best rank-1 approximation. The error aprxerr=3.1e-9, the ratio $\rho(\mathcal{F}) = 0.7728$, the residual $\|\mathcal{F} - \lambda \cdot u^1 \otimes u^2 \otimes u^3\| = 5.0093$, and $\|\mathcal{F}\| = 7.8930$.

For a range of values of n from 5 to 40, we apply Algorithm 2.4 to get rank-1 approximations. The computational results are listed in Table 7. For all of them, the matrices $K(w^*)$ have rank one, so we get best rank-1 approximations. Most errors aprxerr are very tiny. For $n = 40$, the semidefinite relaxation is very large (the matrices have length 1600, and there are 672,399 variables); it was not solved very accurately by SDPNAL.

Example 3.15. Consider the tensor $\mathcal{F} \in \mathbb{R}^{n\times n\times n\times n}$ given as

$$(\mathcal{F})_{i_1,\cdots,i_4} = \begin{cases} \sum_{j=1}^{4} \arcsin\left((-1)^{i_j}\frac{j}{i_j}\right) & \text{if all } i_j \geq j,\\ 0 & \text{otherwise.} \end{cases}$$

For $n = 5$, we apply Algorithm 2.4 and get the rank-1 tensor $\lambda \cdot u^1 \otimes u^2 \otimes u^3 \otimes u^4$ with $\lambda = 15.3155$ and

$$\begin{aligned} u^1 &= (0.6711, 0.2776, 0.4398, 0.3285, 0.4138),\\ u^2 &= (0, 0.1709, 0.6708, 0.3985, 0.6017),\\ u^3 &= (0, 0, 0.8048, 0.1805, 0.5655),\\ u^4 &= (0, 0, 0, -0.0073, -0.9999). \end{aligned}$$

It takes about 3.8 seconds. The matrix $K(w^*)$ has rank one, so $\lambda \cdot u^1 \otimes u^2 \otimes u^3 \otimes u^4$ is a best rank-1 approximation. The error aprxerr=9.2e-10, the ratio $\rho(\mathcal{F}) = 0.7076$, the residual $\|\mathcal{F} - \lambda \cdot u^1 \otimes u^2 \otimes u^3\| = 15.2957$, and $\|\mathcal{F}\| = 21.6454$.

For a range of values of n from 5 to 12, we apply Algorithm 2.4 to find rank-1 approximations. The computational results are shown in Table 8. For all of them, the

TABLE 8
Computational results for Example 3.15.

n	Time	λ	aprxerr	$\rho(\mathcal{F})$	n	Time	λ	aprxerr	$\rho(\mathcal{F})$
5	0:00:04	15.32	9.2e-10	0.71	6	0:00:06	25.39	6.7e-9	0.75
7	0:00:17	33.04	7.2e-8	0.69	8	0:00:30	44.66	3.8e-8	0.71
9	0:02:44	54.40	5.9e-8	0.68	10	0:05:56	67.35	1.1e-8	0.69
11	0:14:13	78.93	5.9e-9	0.67	12	2:00:50	93.02	2.4e-8	0.68

TABLE 9
Computational results for Example 3.16.

n	Items	Values	u^i	Vector entries
4	(N,M)	(256,9999)	u^1	$(0.5776, 0.4950, 0.4646, 0.4534)$
	λ	30.1125	u^2	$(0.3279, 0.4956, 0.5573, 0.5800)$
	time	0:00:07	u^3	$(0.7268, 0.4679, 0.3727, 0.3376)$
	aprxerr	1.1e-10	u^4	$(0.0998, 0.4636, 0.5974, 0.6467)$
	$\rho(\mathcal{F})$	0.85	u^5	$(0.8982, 0.3793, 0.1884, 0.1182)$
5	(N,M)	(625,50624)	u^1	$(0.5282, 0.4515, 0.4233, 0.4129, 0.4091)$
	λ	48.8437	u^2	$(0.2691, 0.4249, 0.4822, 0.5033, 0.5110)$
	time	0:01:20	u^3	$(0.6865, 0.4456, 0.3570, 0.3244, 0.3124)$
	aprxerr	3.3e-7	u^4	$(0.0339, 0.3734, 0.4983, 0.5442, 0.5612)$
	$\rho(\mathcal{F})$	0.83	u^5	$(0.8822, 0.3861, 0.2037, 0.1365, 0.1118)$
6	(N,M)	(1296,194480)	u^1	$(0.4915, 0.4183, 0.3915, 0.3816, 0.3779, 0.3766)$
	λ	71.9071	u^2	$(0.2266, 0.3751, 0.4298, 0.4499, 0.4573, 0.4600)$
	time	0:19:58	u^3	$(0.6557, 0.4259, 0.3413, 0.3102, 0.2988, 0.2946)$
	aprxerr	8.6e-8	u^4	$(-0.0120, 0.3124, 0.4317, 0.4757, 0.4918, 0.4977)$
	$\rho(\mathcal{F})$	0.81	u^5	$(0.8707, 0.3875, 0.2097, 0.1443, 0.1202, 0.1114)$

matrices $K(w^*)$ have rank one, so we get best rank-1 approximations. The approximation errors aprxerr are all very tiny. For $n \leq 8$, Algorithm 2.4 takes a few seconds; for $n = 9, 10, 11$, it takes a couple of minutes. For $n = 12$, it takes about 4 hours; in this case, the semidefinite relaxation is very big. (The matrices have length 1728 and there are $474,551$ variables.)

Example 3.16. Consider the tensor $\mathcal{F} \in \mathbb{R}^{n\times n\times n\times n\times n}$ given as

$$(\mathcal{F})_{i_1,i_2,i_3,i_4,i_5} = \sum_{j=1}^{5} (-1)^{j+1} \cdot j \cdot \exp\{-i_j\}.$$

For $n = 4, 5, 6$, we apply Algorithm 2.4 to get rank-1 approximations. The computational results are listed in Table 9. All the computed matrices $K(w^*)$ have rank one, so best rank-1 approximations are found for all of them. The errors aprxerr are all very tiny. In Table 9, the pair (N, M) measures the sizes of semidefinite relaxations, with N the length of matrices and M the number of variables. For $n = 4, 5$, it takes a short time; for $n = 6$, it takes about 20 minutes.

Example 3.17. Let B be the symmetric matrix

$$\begin{bmatrix}
1 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 & -1/2 \\
0 & 2 & 0 & -1/2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1/2 & 0 & 0 \\
0 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1/2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1/2 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & -1/2 & 0 \\
0 & 0 & -1/2 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\
-1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 & 1
\end{bmatrix}.$$

TABLE 10
Computational results for Example 3.18 *with* $m = 3$.

(n_1, n_2, n_3)	Time (min,med,max)			`aprxerr` (min,med,max)
$(10 \times 10 \times 10)$	0:00:02	0:00:02	0:00:03	(1.6e-9,2.2e-8,2.7e-7)
$(15 \times 15 \times 15)$	0:00:10	0:00:12	0:00:18	(5.9e-9,5.8e-7,2.1e-3)
$(20 \times 20 \times 20)$	0:00:05	0:00:48	0:01:24	(1.9e-9,5.7e-7,5.2e-3)
$(25 \times 25 \times 25)$	0:00:40	0:02:26	0:04:57	(3.2e-9,4.6e-7,5.1e-2)
$(30 \times 30 \times 30)$	0:05:48	0:07:57	0:11:31	(3.6e-8,1.6e-3,3.5e-2)
$(35 \times 35 \times 35)$	0:21:43	0:27:04	1:00:05	(1.1e-5,7.7e-3,5.7e-2)
$(40 \times 40 \times 40)$	1:10:05	1:30:24	1:36:24	(4.8e-4,9.4e-4,1.4e-2)

The eigenvalues of B are

$$\frac{2-\sqrt{5}}{2}, \quad 0, \quad \frac{3}{2}, \quad \frac{2+\sqrt{5}}{2},$$

which are all less than 3. Consider the tensor $\mathcal{F} \in \mathbb{R}^{3\times 3\times 9}$ such that

$$F^{sq}(x^1, x^2) = (x^1 \otimes x^2)^T (3I_9 - B)(x^1 \otimes x^2).$$

The bi-quadratic form $3\|x^1\|_2^2\|x^2\|_2^2 - F^{sq}(x^1, x^2)$ is nonnegative but not SOS (cf. [5]). The minimum of $(x^1 \otimes x^2)^T B(x^1 \otimes x^2)$ over $\mathbb{S}^2 \times \mathbb{S}^2$ is zero [21, Example 5.1], so $F_{\max} = 3$. We apply Algorithm 2.4. The computed matrix $K(w^*)$ has rank four, which is bigger than one. The upper bound $F_{\max}^{\rm sdp} = 3.0972$. Applying `fmincon`, we get the improved tuple (u^1, u^2, u^3) and λ as

$$u^1 = (0, 1, 0),\ u^2 = (1, 0, 0),\ u^3 = (0, 0.1246, 0, 0, 0, 0, -0.9922, 0, 0),$$
$$\lambda = F(u^1, u^2, u^3) = 1.7321 = \sqrt{F_{\max}}.$$

So, $\lambda \cdot u^1 \otimes u^2 \otimes u^3$ is a best rank-1 approximation, by Theorem 1.1. The ratio $\rho(\mathcal{F}) = 0.4083$, the residual $\|\mathcal{F} - \lambda \cdot u^1 \otimes u^2 \otimes u^3\| = 3.8730$, and $\|\mathcal{F}\| = 4.2426$.

Example 3.18 (random examples). We explore the performance of Algorithm 2.4 on randomly generated nonsymmetric tensors $\mathcal{F} \in \mathbb{R}^{n_1\times\cdots\times n_m}$. The entries of $\mathcal{F}$ are generated obeying Gaussian distributions (by `randn` in MATLAB). As in (3.4), let N be the length of matrices and M be the number of variables in the semidefinite relaxations. If $N < 1000$, we generate 50 instances of $\mathcal{F}$ randomly; if $N > 1000$, we generate 10 instances of $\mathcal{F}$ randomly. We apply Algorithm 2.4 to get rank-1 approximations. The computational results for orders $m = 3, 4, 5$ are shown in Tables 10, 11, and 12, respectively. When $n_1 = \cdots = n_m$, the sizes of the semidefinite relaxations are typically much larger than the sizes for symmetric tensors. For instance, for $(n, m) = (40, 3)$, $(N, M) = (861, 135750)$ for the symmetric case, while $(N, M) = (1600, 672399)$ for the nonsymmetric case. Typically, Algorithm 2.4 takes more time than Algorithms 2.1 and 2.3, when the input tensors have the same dimensions and orders.

We can observe from Tables 10, 11, and 12 that for most instances, Algorithm 2.4 is able to get best rank-1 approximations very accurately, within a reasonably short time. For a few cases, the errors are a bit relatively large, around 10^{-2}, which is probably because the semidefinite relaxations are not very tight.

Example 3.19. In this example, we explore the performance of HOPM proposed in [9, Algorithm 3.2], which is widely used for computing rank-1 approximations for nonsymmetric tensors. HOPM can be easily implemented. Typically, HOPM generates a sequence of rank-1 tensors $\mu_k v^{k,1} \otimes \cdots \otimes v^{k,m}$ (with each $\|v^{k,j}\| = 1$).

TABLE 11
Computational results for Example 3.18 *with* $m = 4$.

(n_1, n_2, n_3, n_4)	Time (min,med,max)			`aprxerr` (min,med,max)
(5× 5× 5× 5)	0:00:02	0:00:03	0:00:05	(1.0e-10,1.7e-8,3.1e-7)
(8× 8× 8× 8)	0:00:09	0:00:17	0:00:28	(2.3e-7,1.5e-6,1.1e-5)
(10 × 10 × 10 × 10)	0:00:57	0:01:52	0:10:24	(9.4e-8,2.0e-6,6.6e-3)
(15 × 15 × 5 × 15)	0:02:18	0:07:53	0:13:54	(1.8e-7,3.7e-7,3.7e-3)
(12 × 12 × 12 × 12)	0:07:24	0:10:47	0:39:07	(1.7e-7,3.3e-6,2.7e-2)
(20 × 20 × 5 × 20)	1:23:06	1:55:14	2:36:27	(4.0e-8,3.7e-4,2.0e-2)

TABLE 12
Computational results for Example 3.18 *with* $m = 5$.

$(n_1, n_2, n_3, n_4, n_5)$	Time (min,med,max)			`aprxerr` (min,med,max)
(5 × 5 × 5 × 5 × 5)	0:00:14	0:00:24	0:00:35	(7.2e-8,3.7e-7,3.6e-6)
(10 × 5 × 5 × 4 × 10)	0:00:57	0:01:20	0:03:27	(9.7e-8,4.7e-7,1.5e-5)
(10 × 5 × 8 × 5 × 10)	0:21:16	0:40:06	1:22:28	(2.1e-6,1.3e-4,2.2e-3)
(8 × 8 × 8 × 4 × 10)	1:11:02	1:24:29	2:53:40	(1.4e-7,2.4e-3,1.6e-2)

It is usually very hard to check whether this sequence converges to a best rank-1 approximation. Since Algorithm 2.4 often produces best rank-1 approximations, we can use it to check convergence of HOPM. We write a straightforward MATLAB code to implement HOPM and terminate its iterations if either $|\mu_{k+1} - \mu_k| < 10^{-8}$ or it runs over 1000 iterations. Like for the symmetric case, we use the truncated HOSVD (cf. [9]) to generate starting points. The tensor $\mathcal{F} \in \mathbb{R}^{n\times\cdots\times n}$ we used is given as

$$(\mathcal{F})_{i_1,\cdots,i_m} = \tan\left(i_1 - \frac{i_2}{2} + \cdots + (-1)^{m+1}\frac{i_m}{m}\right).$$

Let $\lambda \cdot u^1 \otimes \cdots \otimes u^m$ be the rank-1 approximation returned by Algorithm 2.4, and let $\mu \cdot v^1 \otimes \cdots \otimes v^m$ be the one returned by HOPM. As in Example 3.10, we measure their qualities by the residuals:

$$\mathrm{RES}_{\mathtt{sdp}} = \|\mathcal{F} - \lambda \cdot u^1 \otimes \cdots \otimes u^m\|, \qquad \mathrm{RES}_{\mathtt{hopm}} = \|\mathcal{F} - \mu \cdot v^1 \otimes \cdots \otimes v^m\|.$$

For $m = 3, 4$ and a range of n, the computational results are presented in Tables 13 and 14. For cleanness of comparisons, only two decimal digits are displayed. For all the cases, Algorithm 2.4 produced best rank-1 approximations (because the computed matrices $K(w^*)$ have rank one). HOPM does not get the best for most of them. The computational results for the case $(n, m) = (10, 4)$ are not shown, because the software `SDPNAL` experiences numerical troubles and the semidefinite relaxations cannot be solved accurately. As for the consumed time, Algorithm 2.4 takes up to a few minutes, while HOPM takes up to tens of seconds. Generally, HOPM takes less time,

TABLE 13
Computational results for Example 3.19 *with* $m = 3$.

n	$\lvert\lambda\rvert$	$\lvert\mu\rvert$	$\mathrm{RES}_{\mathtt{sdp}}$	$\mathrm{RES}_{\mathtt{hopm}}$
15	449.19	349.73	1264.41	1295.45
20	508.82	382.66	1780.56	1811.88
25	579.62	442.98	2304.82	2334.94
30	655.10	492.50	3022.60	3053.31
35	709.39	555.07	3595.36	3622.40

TABLE 14
Computational results for Example 3.19 *with* $m = 4$.

n	$\|\lambda\|$	$\|\mu\|$	$\text{RES}_{\texttt{sdp}}$	$\text{RES}_{\texttt{hopm}}$
7	180.22	171.39	596.18	598.78
8	191.73	117.53	691.98	708.37
9	233.49	211.69	904.78	910.13
11	386.04	379.96	1512.92	1514.71
12	550.21	357.85	2069.34	2111.13

but it might not find best rank-1 approximations. On the other hand, the residual $\text{RES}_{\texttt{hopm}}$, generated by HOPM, is not significantly bigger than $\text{RES}_{\texttt{sdp}}$, generated by semidefinite relaxations. This shows that HOPM still gives reasonably good rank-1 approximations, though it might not produce the best.

4. Conclusions and discussions. This paper proposes semidefinite relaxations to find the best rank-1 approximations for both symmetric and nonsymmetric tensors. Three algorithms based on semidefinite relaxations are presented, respectively, for even symmetric tensors, odd symmetric tensors, and nonsymmetric tensors. As shown in our numerical experiments, they very often produce best rank-1 approximations, which can be checked mathematically.

In section 2, we only presented the lowest order semidefinite relaxations. When they are not tight, higher order semidefinite relaxations can be applied, and we can get a convergent hierarchy of semidefinite relaxations, as shown by Lasserre [18]. Indeed, this hierarchy almost always converges within finitely many steps, as recently shown in [22]. A question that is closely related to best rank-1 approximations is to compute extreme Z-eigenvalues for symmetric tensors of even orders. Hu, Huang, and Qi [12] proposed a convergent sequence of SOS relaxations for computing maximum or minimum Z-eigenvalues.

Semidefinite relaxations in rank-1 tensor approximations are often large scale. The traditional interior point methods are generally too expensive to be used for solving such SDPs. In our work, the Newton-CG augmented Lagrange method by Zhao, Sun, and Toh [32] is applied. The software `SDPNAL` [31] is based on this method. It is very suitable for solving such large-scale SDPs. In most of our numerical experiments, `SDPNAL` successfully solved the semidefinite relaxations, and we got best rank-1 approximations. For a few cases, `SDPNAL` has trouble to get accurate solutions. This is probably because these semidefinite programs are degenerate. Typically, `SDPNAL` [31] works well when the SDP is nondegenerate. For degenerate SDPs, `SDPNAL` might experience numerical troubles and may have very slow convergence. It is an important future work to design efficient methods for solving large-scale, possibly degenerate, semidefinite programs arising from tensor approximations.

In practice, the optimal matrices of semidefinite relaxations (2.3), (2.7), and (2.14) often have rank one. For such semidefinite relaxations, the low-rank methods by Burer and Monteiro [3, 4] can be applied. They can also be very efficient in applications. On the other hand, these kinds of methods are not always guaranteed to get optimal solutions of semidefinite relaxations, because local nonlinear optimization methods are mainly used.

In Algorithms 2.1, 2.3, and 2.4, if the matrices $M(y^*)$, $M(z^*)$, $K(w^*)$ have rank one, we are guaranteed to get best rank-1 approximations. If their ranks are bigger than one, we can improve the rank-1 approximation by using some nonlinear

optimization methods. In our numerical experiments, we used the MATLAB Optimization Toolbox function `fmincon` to do this. More advanced nonlinear optimization methods can also be applied, e.g., the quasi-Newton method by Savas and Lim [29].

Acknowledgment. The authors would like very much to thank the associate editor and two anonymous referees for fruitful comments on improving the paper.

REFERENCES

[1] J. Brachat, P. Comon, B. Mourrain, and E. Tsigaridas, *Symmetric tensor decomposition*, Linear Algebra Appl., 433 (2010), pp. 1851–1872.
[2] A. Bernardi, J. Brachat, P. Comon, and B. Mourrain, *General tensor decomposition, moment matrices and applications*, J. Symbolic Comput., 52 (2013), pp. 51–71.
[3] S. Burer and R. D. C. Monteiro, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Math. Program. Ser. B, 95 (2003), pp. 329-357.
[4] S. Burer and R. D. C. Monteiro, *Local minima and convergence in low-rank semidefinite programming*, Math. Program. Ser. A, 103 (2005), pp. 427–444.
[5] M. D. Choi, *Positive semidefinite biquadratic forms*, Linear Algebra Appl., 12 (1975), pp. 95–100.
[6] P. Comon, *Tensor Decompositions: State of the Art and Applications*, presented at IMA Conference on Signal Processing, Warwick, UK, 2000.
[7] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, *Symmetric tensors and symmetric tensor rank*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1254–1279.
[8] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.
[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-*1 *and rank-*$(R1, R2, \ldots, RN)$ *approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.
[10] V. De Silva and L.-H. Lim, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1084–1127.
[11] C. Hillar and L.-H. Lim, *Most tensor problems are NP-hard*, J. ACM, 60 (2013).
[12] S. Hu, Z. H. Huang, and L. Qi, *Finding the extreme Z-eigenvalues of tensors via a sequential semidefinite programming method*, Numer. Linear Algebra Appl., 20 (2013), pp. 972–984.
[13] M. Ishteva, P.-A. Absil, and P. Van Dooren, *Jacobi algorithm for the best low multilinear rank approximation of symmetric tensors*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 651–672.
[14] E. Kofidis and P. Regalia, *On the best rank-*1 *approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 863–884.
[15] T. Kolda and B. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.
[16] T. Kolda and J. Mayo, *Shifted power method for computing tensor eigenpairs*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1095–1124.
[17] J. M. Landsberg, *Tensors: Geometry and Applications*, Grad. Stud. Math., 128, AMS, Providence, RI, 2012.
[18] J. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.
[19] J. Lasserre, *Moments, Positive Polynomials and Their Applications*, Imperial College Press, London, UK, 2009.
[20] M. Laurent, *Sums of squares, moment matrices and optimization over polynomials*, in Emerging Applications of Algebraic Geometry, IMA Vol. Math. Appl., 149 Springer, New York, 2009, pp. 157–270.
[21] C. Ling, J. Nie, L. Qi, and Y. Ye, *Biquadratic optimization over unit spheres and semidefinite programming relaxations*, SIAM J. Optim., 20 (2009), pp. 1286–1310.
[22] J. Nie, *Optimality conditions and finite convergence of Lasserre's hierarchy*, Math. Program., Math. Program., 146 (2014), pp. 97–121.
[23] J. Nie, *Sum of squares methods for minimizing polynomial forms over spheres and hypersurfaces*, Frontiers Math. China, 7 (2012), pp. 321–346.
[24] J. Nie and L. Wang, *Regularization methods for SDP relaxations in large scale polynomial optimization*, SIAM J. Optim., 22 (2012), pp. 408–428.
[25] L. Oeding and G. Ottaviani, *Eigenvectors of tensors and algorithms for Waring decomposition*, J. Symbolic Comput., 54, (2013), pp. 9–35.

[26] P. Parrilo and B. Sturmfels, *Minimizing polynomial functions*, in Algorithmic and Quantitative Real Algebraic Geometry, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 60, AMS, Providence, RI, 2003.
[27] L. Qi, *The best rank-one approximation ratio of a tensor space*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 430–442.
[28] B. Reznick, *Some concrete aspects of Hilbert's* 17*th problem*, Contemp. Math., 253 (2000), pp. 251–272.
[29] B. Savas and L.-H. Lim, *Quasi-Newton methods on Grassmannians and multilinear approximations of tensors*, SIAM J. Sci. Comput., 32 (2010), pp. 3352–3393.
[30] M. Todd, *Semidefinite optimization*, Acta Numer., 10 (2001), pp. 515–560.
[31] X. Zhao, D. Sun, and K. Toh, SDPNAL version 0.1, http://www.math.nus.edu.sg/∼mattohkc/SDPNAL.html.
[32] X. Zhao, D. Sun, and K. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optim., 20 (2010), pp. 1737–1765.
[33] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.
[34] X. Zhang, L. Qi, and Y. Ye, *The cubic sperical optimization problems*, Math. Comp., 81 (2012), pp. 1513–1525.
[35] X. Zhang, C. Ling, and L. Qi, *The best rank-*1 *approximation of a symmetric tensor and related spherical optimization problems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 806–821.