SIAM J. MATRIX ANAL. APPL.
Vol. 40, No. 4, pp. 1527–1554

BEST NONNEGATIVE RANK-ONE APPROXIMATIONS OF TENSORS[*]

SHENGLONG HU[†], DEFENG SUN[‡], AND KIM-CHUAN TOH[§]

Abstract. In this paper, we study the polynomial optimization problem of a multiform over the intersection of the multisphere and the nonnegative orthants. This class of problems is NP-hard in general and includes the problem of finding the best nonnegative rank-one approximation of a given tensor. A Positivstellensatz is given for this class of polynomial optimization problems, based on which a globally convergent hierarchy of doubly nonnegative (DNN) relaxations is proposed. A (zeroth order) DNN relaxation method is applied to solve these problems, resulting in linear matrix optimization problems under both the positive semidefinite and nonnegative conic constraints. A worst case approximation bound is given for this relaxation method. The recent solver SDPNAL+ is adopted to solve this class of matrix optimization problems. Typically the DNN relaxations are tight, and hence the best nonnegative rank-one approximation of a tensor can be obtained frequently. Extensive numerical experiments show that this approach is quite promising.

Key words. tensor, nonnegative rank-1 approximation, polynomial, multiforms, doubly nonnegative semidefinite program, doubly nonnegative relaxation method

AMS subject classifications. 15A18, 15A42, 15A69, 90C22

DOI. 10.1137/18M1224064

1. Introduction.

Nonnegative factorizations of data observations, prevalent in data analysis, have been popularized to an unprecedented level since the works of Paatero and Tapper [45] and Lee and Seung [30]. In many applications, data are naturally represented by third order or higher order tensors (also known as hypermatrices). For example, a color image is stored digitally as a third order nonnegative tensor composed of three nonnegative matrices, representing the red, green, and blue pixels, and therefore a set of such images or a video is actually a fourth order nonnegative tensor. In the literature, however, these fourth order tensors are typically flattened into matrices for data analysis [8, 22, 30, 45]. As we can see, intrinsic structures of an image or a video are destroyed after the flattening. Therefore, direct treatments of tensors are necessary, and accordingly nonnegative factorizations of higher order data are needed. As a result, tensor counterparts of the nonnegative matrix factorizations have become a new frontier in this area [1, 11, 13, 26, 31, 44, 54, 55, 64]. As expected, nonnegative tensor factorizations have their own advantages over the tradi-

[*] Received by the editors November 1, 2018; accepted for publication (in revised form) by L. De Lathauwer October 7, 2019; published electronically December 3, 2019.
https://doi.org/10.1137/18M1224064

Funding: The work of the first author was partially supported by the National Natural Science Foundation of China grant 11771328, the Young Elite Scientists Sponsorship Program, and by Tianjin and Innovation Research Foundation of Tianjin University grants 2017XZC-0084 and 2017XRG-0015. The work of the second author was partially supported by a start-up research grant from the Hong Kong Polytechnic University and the Hong Kong Research Grant Council grant PolyU153014/18p. The work of the third author was partially supported by the Ministry of Education, Singapore, Academic Research Fund grant R-146-000-257-112.

[†] Department of Mathematics, School of Science, Hangzhou Dianzi University, Hangzhou 310018, China (shenglonghu@hdu.edu.cn).

[‡] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong (defeng.sun@polyu.edu.hk).

[§] Department of Mathematics and Institute of Operations Research and Analytics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore (mattohkc@nus.edu.sg).

tional nonnegative matrix factorizations; see, for example, [22, 56] and the references therein.

Nonnegative tensor factorizations have found diverse applications, such as latent class models in statistics, spectroscopy, sparse image coding in computer vision, sound source separation, and pattern recognition, etc.; see [8, 15, 22, 55, 58] and the references therein. Several methods have been proposed for computing nonnegative tensor factorizations; see, for example, [7, 9, 16, 29, 33, 44, 57, 64, 67] and the references therein. Due to errors in measurements of the data collected or simply because of inattainability, the problem of approximating a given tensor by a nonnegative tensor factorization of low rank occurs more often in practice than the problem of finding the exact factorization of a given tensor. We note that the existence and uniqueness of nonnegative tensor factorizations are well studied in [33, 52].

As is well known, mathematical models and numerical methods are both necessary ingredients for tackling an application problem. An accurate method for the mathematical model is a necessary tool to certify whether the model built is appropriate and whether we are on the correct path for solving the application problem. Thus, it is of both theoretical importance and application necessity to study global solution methods for computing the best low rank nonnegative tensor approximation. As an initial attempt, in this paper, we will investigate the specific case when the approximating tensor is of rank one. There is also another motivation from the computational perspective. For a given tensor, a classical method to compute a nonnegative tensor factorization/approximation is by *multiple best nonnegative rank-one approximations*, proposed by Shashua and Hazan [55]. The principle is alternatively splitting/approximating the given tensor by several (nonnegative) ones and approximating each (nonnegative) tensor by a best nonnegative rank-one tensor. Therefore, in this framework, *finding the best nonnegative rank-one approximation of a given tensor* is of crucial importance in nonnegative tensor factorizations/approximations. This problem is also the foundation for the heuristic methods based on greedy rank-one downdating for nonnegative factorizations [2, 4, 5, 19]. It plays an analogous role as *the best rank-one approximations of tensors* to approximation and decomposition problems of tensors, studied by De Lathauwer, De Moor, and Vandewalle [11, 12].

This article will focus on the problem of computing the best nonnegative rank-one approximation of a given tensor from the perspective of mathematical optimization. The problem will be formulated as a polynomial minimization problem over the intersection of the multisphere and the nonnegative orthants. With this formulation, the study can also be applied to the problem of testing the copositivity for a homogeneous polynomial, which is important in completely positive programming [47].

A negative aspect from the computational complexity point of view is that the problem under consideration is NP-hard in general (cf. Proposition 3.1); see also [14, 21, 36, 63]. Thus, no algorithm with polynomial complexity exists unless P=NP. Consequently, in practical applications, approximation or relaxation methods are employed to solve this problem. In this article, instead of adopting the traditional sums of squares (SOS) relaxation methods for a polynomial optimization problem (cf. [28, 37, 38, 39, 46]), we will introduce a *doubly nonnegative (DNN) relaxation method* to solve this problem. DNN relaxation methods will provide tighter approximation results, since the cone of SOS polynomials is strictly contained in the cone of polynomials that can be written as sums of SOS polynomials and polynomials with nonnegative coefficients. While the standard SOS relaxations of a polynomial optimization problem will give rise to standard semidefinite programming (SDP) problems with variables in the SDP cones, the DNN relaxation method will give rise to DNNSDP problems whose variables are constrained to be in the SDP cones and the cones of nonnegative matrices, in addition to linear equality constraints. It has been well recognized that solving the DNNSDP problems by primal-dual interior-point methods as implemented in popular solvers such as Mosek, SDPT3 [62], or SeDuMi [59] is computationally much more challenging than solving the standard SDP counterparts. Fortunately, with the recent advances on augmented Lagrangian based methods for solving SDP problems with bound constraints [60, 61, 65], we have reached a stage where solving the DNNSDP problems is computationally not much more expensive than the standard SDP counterparts. In this paper, we will employ the Newton-conjugate gradient augmented Lagrangian method implemented in the solver SDPNAL+ [65] to solve the DNNSDP problems arising from best nonnegative rank-one tensor approximation problems. Extensive numerical computations will show that our new approach is quite promising.

The main message we want to convey in this paper is that the proposed DNN approach can serve as a global optimal solution method for the best nonnegative rank-one tensor approximation problem. More importantly, it can certify the global optimality of the solution found in many cases. Putting aside the requirement of accurate global solution in several applications, this will also provide a benchmark for evaluating the quality of approximate solutions obtained by faster local solution methods, such as alternating methods [24, 58]. It can be seen from section 4.1.2 that the gap (in terms of attained approximation errors) between the approximate solutions obtained by the global and local method is not negligibly small. Although the scalability of the DNN approach is limited by that of the SDP solvers at present, much more efficient numerical computation for specific problems by exploiting data structure within this approach is conceivable.

The remaining parts of this article are organized as follows. Some preliminaries will be given in section 2, in which nonnegative tensor approximations and in particular the best nonnegative rank-one approximation problems will be presented in section 2.1. The problem of testing the copositivity of a tensor will be given in section 2.2. Both the problems in section 2 will be formulated as minimizing a multiform over the intersection of the multisphere and the nonnegative orthants in section 3. In the ensuing section, basic properties of this polynomial optimization problem will be investigated, including a Positivstellensatz for this problem (cf. section 3.3), the DNN relaxation (cf. section 3.4), a worst case approximation bound (cf. section 3.6), and the extraction of a nonnegative rank-one tensor from a solution of the DNN problem (cf. sections 3.7 and 3.8). Numerical experiments will be presented in section 4, in which extensive examples on best nonnegative rank-one approximations and examples on testing the copositivity of a tensor will be given. Some conclusions will be given in the last section.

2. Preliminaries.

In this article, tensors will be considered in the most general setting. Given positive integers $n_1,\dots,n_r$, a tensor $\mathcal{A}\in\mathbb{R}^{n_1}\otimes\cdots\otimes\mathbb{R}^{n_r}$ is a collection of $n_1\cdots n_r$ scalars $a_{i_1\dots i_r}$, termed as the entries of $\mathcal{A}$, for all $i_j\in\{1,\dots,n_j\}$ and $j\in\{1,\dots,r\}$. If $n_1=\cdots=n_r=n$, $\mathbb{R}^{n_1}\otimes\cdots\otimes\mathbb{R}^{n_r}$ is abbreviated as $\otimes^r\mathbb{R}^n$. Given positive integers p, $\alpha_1,\dots,\alpha_p$, $n_1,\dots,n_p$, we denote by $\operatorname{Sym}(\otimes^{\alpha_i}\mathbb{R}^{n_i})$ the symmetric subspace of the tensor space $\otimes^{\alpha_i}\mathbb{R}^{n_i}$, consisting of real symmetric tensors with order α_i and dimension n_i, and $\operatorname{Sym}(\otimes^{\alpha_1}\mathbb{R}^{n_1})\otimes\cdots\otimes\operatorname{Sym}(\otimes^{\alpha_p}\mathbb{R}^{n_p})$ the tensor space with p symmetric factors. Note that when $p=1$, the tensor space is the usual space of symmetric tensors, and when $\alpha_1=\cdots=\alpha_p=1$, the tensor space is the usual space of nonsymmetric tensors. A tensor $\mathcal{A}\in\operatorname{Sym}(\otimes^{\alpha_1}\mathbb{R}^{n_1})\otimes\cdots\otimes\operatorname{Sym}(\otimes^{\alpha_p}\mathbb{R}^{n_p})$ is usually referred to as a *partially symmetric* tensor, which appears in many applications. A symmetric rank-one tensor in $\mathrm{Sym}(\otimes^{\alpha_i}\mathbb{R}^{n_i})$ is an element $\left(\mathbf{x}^{(i)}\right)^{\otimes\alpha_i}$ for some nonzero vector $\mathbf{x}^{(i)}\in\mathbb{R}^{n_i}$, where $\left(\mathbf{x}^{(i)}\right)^{\otimes\alpha_i}$ is a short hand for

$$\underbrace{\mathbf{x}^{(i)}\otimes\cdots\otimes\mathbf{x}^{(i)}}_{\alpha_i\text{ copies}}.$$

Therefore, a rank-one tensor in $\mathrm{Sym}(\otimes^{\alpha_1}\mathbb{R}^{n_1})\otimes\cdots\otimes\mathrm{Sym}(\otimes^{\alpha_p}\mathbb{R}^{n_p})$ is of the form $\mathbf{x}^{\alpha}:=\left(\mathbf{x}^{(1)}\right)^{\otimes\alpha_1}\otimes\cdots\otimes\left(\mathbf{x}^{(p)}\right)^{\otimes\alpha_p}$ for some nonzero vectors $\mathbf{x}^{(i)}\in\mathbb{R}^{n_i}$, $i=1,\ldots,p$.

As an Euclidean space, the inner product $\langle\mathcal{A},\mathcal{B}\rangle$ of two tensors $\mathcal{A},\mathcal{B}\in\mathbb{R}^{n_1}\otimes\cdots\otimes\mathbb{R}^{n_r}$ is defined as

$$\langle\mathcal{A},\mathcal{B}\rangle:=\sum_{i_1=1}^{n_1}\cdots\sum_{i_r=1}^{n_r}a_{i_1\ldots i_r}b_{i_1\ldots i_r}.$$

The Hilbert–Schmidt norm $\|\mathcal{A}\|$ is then defined as

$$\|\mathcal{A}\|:=\sqrt{\langle\mathcal{A},\mathcal{A}\rangle}.$$

We refer the readers to [32] and the references herein for basic notions on tensors.

2.1. Nonnegative tensor approximation.

In the context of computer vision, chemometrics, statistics, and spectral intensity, the multiway (tensor) data often cannot take negative values. Therefore, one expects to approximate as much as possible the observed data (which may have negative components) $\mathcal{A}\in\mathrm{Sym}(\otimes^{\alpha_1}\mathbb{R}^{n_1})\otimes\cdots\otimes\mathrm{Sym}(\otimes^{\alpha_p}\mathbb{R}^{n_p})$ with a sum of rank-one nonnegative tensors

$$\mathcal{A}\approx\sum_{i=1}^{r}\lambda_i\mathbf{x}_i^{\otimes\alpha},\ \lambda_i\geq 0,\ \mathbf{x}_i\geq\mathbf{0},\ i=1,\ldots,r \tag{2.1}$$

for some nonnegative integer r, with

$$\mathbf{x}_i:=(\mathbf{x}_i^{(1)},\ldots,\mathbf{x}_i^{(p)})\in\mathbb{R}^{n_1}\times\cdots\times\mathbb{R}^{n_p}$$

and

$$\mathbf{x}_i^{\otimes\alpha}:=\left(\mathbf{x}_i^{(1)}\right)^{\otimes\alpha_1}\otimes\cdots\otimes\left(\mathbf{x}_i^{(p)}\right)^{\otimes\alpha_p}.$$

For a given continuous distance measure ϕ over the tensor space, we can formulate problem (2.1) as

$$\min\left\{\phi(\mathcal{A},\sum_{i=1}^{r}\lambda_i\mathbf{x}_i^{\otimes\alpha}):\lambda_i\geq 0,\ \mathbf{x}_i\geq\mathbf{0},\ i=1,\ldots,r\right\}. \tag{2.2}$$

In most cases, ϕ is chosen as the Hilbert–Schmidt norm distance, i.e., $\phi(\mathcal{A},\mathcal{B}):=\|\mathcal{A}-\mathcal{B}\|$. Problem (2.2) is well defined for each $r\in\mathbb{N}$, while it is NP-hard in most cases. Moreover, it is extremely difficult to solve problem (2.1) when we do not know a priori the value of r, and even if we are lucky enough to know the exact r, it is still very difficult to solve (2.2). Thus, one procedure to solve (2.1) is by multiple best nonnegative rank-one approximations, and another is by successive best nonnegative rank-one approximations.

In this article, we focus on the problem (2.2) with fixed $r=1$, i.e., *the best nonnegative rank-one approximation* of the tensor $\mathcal{A}$. We will see that this problem is already hard, both theoretically and numerically. The computational complexity is NP-hard in general.

With the common choice of ϕ as the Hilbert–Schmidt norm distance, problem (2.2) becomes

$$\begin{array}{rl} \min_{\lambda,\mathbf{x}} & \|\mathcal{A}-\lambda\mathbf{x}^{\otimes\alpha}\|^2 \\ \text{such that (s.t.)} & \lambda\geq 0,\ \langle\mathbf{x}^{(i)},\mathbf{x}^{(i)}\rangle=1,\ \mathbf{x}^{(i)}\geq\mathbf{0}\text{ for all } i=1,\dots,p, \end{array}\tag{2.3}$$

where $\mathbf{x}:=(\mathbf{x}^{(1)},\dots,\mathbf{x}^{(p)})$. It is easy to see that (2.3) always has an optimal solution $(\lambda,\mathbf{x})$ with

$$\lambda:=\begin{cases}\langle\mathcal{A},\mathbf{x}^{\otimes\alpha}\rangle & \text{whenever } \langle\mathcal{A},\mathbf{x}^{\otimes\alpha}\rangle>0,\\ 0 & \text{otherwise,}\end{cases}\tag{2.4}$$

and in both cases

$$\|\mathcal{A}-\lambda\mathbf{x}^{\otimes\alpha}\|^2=\|\mathcal{A}\|^2-\lambda^2.$$

Therefore, (2.3) is equivalent to

$$\begin{array}{rl} \min & \langle-\mathcal{A},\mathbf{x}^{\otimes\alpha}\rangle \\ \text{s.t.} & \langle\mathbf{x}^{(i)},\mathbf{x}^{(i)}\rangle=1,\ \mathbf{x}^{(i)}\geq\mathbf{0}\text{ for all } i=1,\dots,p \end{array}\tag{2.5}$$

in the sense that

1. if the optimal value of (2.5) is nonnegative, then the zero tensor is the best nonnegative rank-one approximation of $\mathcal{A}$,
2. if the optimal value λ of (2.5) is negative with an optimal solution $\mathbf{x}_*$, then $-\lambda\mathbf{x}_*^{\otimes\alpha}$ is the best nonnegative rank-one approximation of $\mathcal{A}$.

2.2. Copositivitiy of tensors.

A given tensor $\mathcal{A}\in\operatorname{Sym}(\otimes^{\alpha_1}\mathbb{R}^{n_1})\otimes\cdots\otimes\operatorname{Sym}(\otimes^{\alpha_p}\mathbb{R}^{n_p})$ is said to be *copositive* if

$$\langle\mathcal{A},\mathbf{x}^{\otimes\alpha}\rangle\geq 0\text{ for all }\mathbf{x}\in\mathbb{R}^{n_1}_+\times\cdots\times\mathbb{R}^{n_p}_+.$$

The copositivity of a tensor is a generalized notion of both the nonnegativity of a matrix and the copositivity of a symmetric matrix. When $p=1$ and $\alpha_1=2$, it reduces to the copositivity of a symmetric matrix, and when $\alpha_1=\cdots=\alpha_p=1$, it reduces to the nonnegativity of a tensor. The problem of deciding the copositivity of a tensor is therefore co-NP-hard [14, 36], i.e., testing whether a given tensor is not copositive is an NP-hard problem. When $p=1$, discussions on copositive tensors can be found in [42, 51] and the references therein.

Testing the copositivity of a tensor can also be formulated as a polynomial optimization problem as in (2.5). Indeed, a tensor $\mathcal{A}$ is copositive if and only if the optimal value of

$$\begin{array}{rl} \min & \langle\mathcal{A},\mathbf{x}^{\otimes\alpha}\rangle \\ \text{s.t.} & \langle\mathbf{x}^{(i)},\mathbf{x}^{(i)}\rangle=1,\ \mathbf{x}^{(i)}\geq\mathbf{0}\text{ for all } i=1,\dots,p \end{array}\tag{2.6}$$

is nonnegative.

3. Homogeneous polynomials.

Since both the problem of finding the best nonnegative rank-one approximation of a tensor and the copositivity certification of a tensor can be equivalently reformulated as (2.5) (or (2.6)), we focus on this polynomial optimization problem in this section.

Let $\mathbf{x} := (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(p)}) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_p}$ be partitioned into p groups. A polynomial $f(\mathbf{x})$ is *multihomogeneous* or a *multiform* if each monomial of f has the same degree with respect to each group variables $\mathbf{x}^{(i)}$ for all $i \in \{1, \dots, p\}$. We consider the following optimization problem:

$$(3.1) \qquad \begin{array}{rll} f_{\min} := & \min & f(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(p)}) \\ & \text{s.t.} & \|\mathbf{x}^{(i)}\| = 1,\ \mathbf{x}^{(i)} \geq \mathbf{0},\ \mathbf{x}^{(i)} \in \mathbb{R}^{n_i},\ i = 1, \dots, p, \end{array}$$

where $f(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(p)}) \in \mathbb{R}[\mathbf{x}]$ is a multiform of even degree $d_i = 2\tau_i$ for some $\tau_i \geq 0$ with respect to each $\mathbf{x}^{(i)}$ for all $i \in \{1, \dots, p\}$. Problem (3.1) covers all instances of minimizing a multiform over the intersection of the multisphere and the nonnegative orthants, since the cases with odd d_i's can be equivalently formulated into (3.1) as in section 3.1. Polynomial optimization over the multisphere is one research direction in recent years; see [34, 37, 41, 43] and the references therein. Moreover, in [34] a biquadratic optimization over the joint sphere (multisphere with $p = 2$) with one group variables being nonnegative is discussed as well.

For easy references, in the following, we will denote the $(n-1)$-dimensional sphere in $\mathbb{R}^n$ as $\mathbb{S}^{n-1}$, i.e., $\mathbb{S}^{n-1} := \{\mathbf{x} \in \mathbb{R}^n \colon \mathbf{x}^\mathsf{T}\mathbf{x} = 1\}$. The nonnegative part of the $(n-1)$-dimensional sphere is denoted by $\mathbb{S}^{n-1}_+$, i.e., $\mathbb{S}^{n-1}_+ := \{\mathbf{x} \in \mathbb{R}^n_+ \colon \mathbf{x}^\mathsf{T}\mathbf{x} = 1\}$. Thus, the feasible set of (3.1) can be called the *nonnegative multisphere*.

3.1. Odd order case. If $f(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(p)})$ is of odd degree $d > 0$ for $\mathbf{x}^{(1)}$ (without loss of generality), then we introduce a variable t and let

$$\tilde{f}(\tilde{\mathbf{x}}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(p)}) := t f(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(p)})$$

with $\tilde{\mathbf{x}}^{(1)} = ((\mathbf{x}^{(1)})^\mathsf{T}, t)^\mathsf{T}$. It can be shown that

$$f_{\min} = \sqrt{\frac{(d+1)^{d+1}}{d^d}}\tilde{f}_{\min},$$

since

$$\max\{t\alpha^d : \alpha^2 + t^2 = 1\} = \sqrt{\frac{d^d}{(d+1)^{d+1}}}$$

with a positive optimal t.

If the degree of f for $\mathbf{x}^{(1)}$ is one, we can construct

$$g(\mathbf{x}^{(2)}, \dots, \mathbf{x}^{(p)}) := \sum_{j=1}^{n_1} \left(f(\mathbf{e}^{(1)}_j, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(p)})\right)^2,$$

where $\mathbf{e}^{(1)}_j \in \mathbb{R}^{n_1}$ is the jth standard basis vector. In some cases, (3.1) can be solved via maximizing g over the nonnegative multisphere, i.e.,

$$(3.2) \qquad \max\{g(\mathbf{x}^{(2)}, \dots, \mathbf{x}^{(p)}) \colon \|\mathbf{x}^{(i)}\| = 1,\ \mathbf{x}^{(i)} \geq \mathbf{0},\ \mathbf{x}^{(i)} \in \mathbb{R}^{n_i},\ i = 2, \dots, p\}.$$

Actually, if $(\mathbf{x}^{(2)}, \dots, \mathbf{x}^{(p)})$ is an optimal solution of (3.2) with positive optimal value and $f(\mathbf{e}^{(1)}_j, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(p)})$'s are all nonpositive, then we can construct a solution for (3.1) from a solution for (3.2). Indeed, one optimal solution of (3.1) is given by

$$\left(\mathbf{x}^{(1)} := -\frac{\mathbf{u}_-}{\|\mathbf{u}\|}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(p)}\right)$$

with

$$\mathbf{u} := (f(\mathbf{e}_1^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(p)}), \dots, f(\mathbf{e}_{n_1}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(p)}))^{\mathsf{T}}$$

and $(\mathbf{u}_-)_i := \min\{0, u_i\}$. This is based on the fact that

$$\min\{\mathbf{x}^{\mathsf{T}}\mathbf{y} : \|\mathbf{y}\| = 1,\ \mathbf{y} \geq \mathbf{0}\} = -\|\mathbf{x}_-\|$$

with the optimizer $\mathbf{y}^* := -\frac{\mathbf{x}_-}{\|\mathbf{x}_-\|}$ when $\mathbf{x}_- \neq \mathbf{0}$. Note that the number of variables is reduced from (3.1) to (3.2).

Before proceeding to the computation of (3.1), we state the computational complexity of it.

3.2. NP-hardness.

PROPOSITION 3.1. *Let $d_i \geq 2$ for all $i \in \{1, \dots, p\}$. Problem* (3.1) *is NP-hard in general.*

Proof. We will construct a subclass of (3.1), which is NP-hard. Let $G = (V, E)$ be a simple graph with the set of vertices being $V = \{1, \dots, n\}$ and the set of edges being E. Let $\Delta_n \subset \mathbb{R}^n_+$ be the standard simplex. Then

$$1 - \frac{1}{\alpha(G)} = 2 \max_{\mathbf{x} \in \Delta_n} \sum_{(i,j) \in E} x_i x_j$$

by the famous Motzkin–Straus theorem [35], where $\alpha(G)$ is the stability number of G. It is well known that computing $\alpha(G)$ is an NP-hard problem [18, 36]. On the other hand, we have that

$$\max_{\mathbf{x} \in \Delta_n} \sum_{(i,j) \in E} x_i x_j = \max_{\|\mathbf{y}\|=1} \sum_{(i,j) \in E} y_i^2 y_j^2 = \max_{\|\mathbf{y}\|=1,\ \mathbf{y} \geq \mathbf{0}} \sum_{(i,j) \in E} y_i^2 y_j^2,$$

where the second equality follows from the fact that in the objective function only squared y_i^2's are involved. Immediately, the last optimization problem is of the form given in (3.1). The required result then follows. □

A standard SOS relaxation can be applied to the polynomial optimization problem (3.1); see [28]. However, in order to reduce the size of the resulting SDP, we would like to compress the spherical constraints as follows.

The homogeneity property implies that (3.1) is equivalent to

$$(3.3)\qquad \begin{array}{rll} f_{\min} := & \min & f(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(p)}) \\ & \text{s.t.} & \prod_{i=1}^{p} \|\mathbf{x}^{(i)}\|^{d_i} = 1, \\ & & \mathbf{x}^{(i)} \geq \mathbf{0},\ \mathbf{x}^{(i)} \in \mathbb{R}^{n_i},\ i = 1, \dots, p \end{array}$$

in the sense that they have the same optimal objective value and we can get an optimal solution for one from the other.

3.3. A Positivstellensatz. Testing the nonnegativity of a polynomial over a (compact) semialgebraic set is a very difficult problem [6]. Thus, certifications of nonnegativity of a polynomial are foundations for polynomial optimization [28]. In the literature, such certifications are called *Positivstellensatz*. Of crucial importance are Putinar's Positivstellensatz [50], Pólya's theorem [48] and Reznick's theorem [53].

While Putinar's result is more general, and the theorems of Pólya and Reznick are applicable only to homogeneous polynomials over the simplices and spheres, respectively, the resulting SDP problems obtained from the latter two theorems have sizes that are about half of those obtained by using Putinar's Positivstellensatz directly. Since the cost of solving SDP problems grow rapidly with the sizes of problems, Pólya's theorem and Reznick's theorem are more important for homogeneous problems.

In this section, we will derive a Positivstellensatz for the optimization problem (3.3) by taking into account both the homogeneity structures of the objective function and constraints, as well as the nonnegativity constraints.

Let $g(\mathbf{x}) := \prod_{i=1}^{p} \|\mathbf{x}^{(i)}\|^{d_i}$ and $\mathcal{F}$ be the feasible set of problem (3.3). Suppose that $\gamma := f_{\min}$ is the optimal value of (3.3). It follows that

$$f(\mathbf{x}) - \gamma g(\mathbf{x}) \geq 0 \text{ for all } \mathbf{x} \in \mathcal{F}.$$

Since $f(\mathbf{x}) - \gamma g(\mathbf{x})$ is a multihomogeneous polynomial, we then have

$$f(\mathbf{x}) - \gamma g(\mathbf{x}) \geq 0 \text{ for all } \mathbf{x} \in \mathbb{S}_+^{n_1-1} \times \cdots \times \mathbb{S}_+^{n_p-1},$$

which is equivalent to

$$f(\mathbf{x}) - \gamma g(\mathbf{x}) \geq 0 \text{ for all } \mathbf{x} \in \Delta_{n_1} \times \cdots \times \Delta_{n_p},$$

where Δ_{n_i} is the standard simplex in $\mathbb{R}^{n_i}$, i.e., $\Delta_{n_i} := \{\mathbf{x} \in \mathbb{R}_+^{n_i} \colon \mathbf{e}^\mathsf{T}\mathbf{x} = 1\}$ with $\mathbf{e}$ being the vector of all ones with matching dimension. In the following, we will discuss the positivity of a multiform over the joint simplex. To that end, we first recall the well-known Pólya theorem on positive polynomials over the simplex [20, 48]. A quantitative version (cf. [49, Theorem 1]) is needed for our analysis.

Let h be a homogeneous polynomial of degree d in n variables with the monomial expansion

$$h(\mathbf{x}) := \sum_{|\alpha|=d} \alpha! h_\alpha \mathbf{x}^\alpha,$$

where $\alpha \in \mathbb{N}^n$ and $\alpha! := \frac{|\alpha|!}{\prod_{i=1}^{n} \alpha_i!}$. Define

$$L(h) := \max\{|h_\alpha| \colon |\alpha| = d\} \quad \text{and} \quad \lambda(h) := \min\{h(\mathbf{x}) \colon \mathbf{x} \in \Delta_n\}.$$

The following result can be found in [49, Theorem 1].

PROPOSITION 3.2. *Let h be a homogeneous polynomial of degree d in n variables and positive on the simplex Δ_n. Then, for any positive integer*

$$r > \frac{d(d-1)}{2} \frac{L(h)}{\lambda(h)} - d,$$

the polynomial

$$(\mathbf{e}^\mathsf{T}\mathbf{x})^r h(\mathbf{x})$$

has positive coefficients.

Next, we will generalize Proposition 3.2 to multiforms over the joint simplex. It will serve as a theoretical foundation for the DNN relaxation methods to be introduced later for (3.3).

PROPOSITION 3.3. *Let f be a multiform of degree d_i with respect to each $\mathbf{x}^{(i)}$ for $i = 1, \ldots, p$. If f is positive on $\Delta_{n_1} \times \cdots \times \Delta_{n_p}$, then*

$$\left[\prod_{i=1}^{p} (\mathbf{e}^\mathsf{T}\mathbf{x}^{(i)})^{r_i}\right] f(\mathbf{x})$$

is a polynomial with positive coefficients for all sufficiently large r_i with $i \in \{1, \ldots, p\}$.

Proof. For any $\gamma = (\gamma^{(1)}, \ldots, \gamma^{(p)}) \in \mathbb{N}^{n_1} \times \cdots \times \mathbb{N}^{n_p}$, let

$$\gamma^{(i)}! := \frac{|\gamma^{(i)}|!}{\prod_{j=1}^{n_i} \gamma_j^{(i)}!} \text{ for all } i \in \{1, \ldots, p\}$$

and

$$\gamma! := \gamma^{(1)}! \ldots \gamma^{(p)}!.$$

Let the monomial expansion of f be

$$f(\mathbf{x}) = \sum_{\alpha \in \Lambda(d_1,\ldots,d_p)} \alpha! a_\alpha \prod_{i=1}^{p} \left(\mathbf{x}^{(i)}\right)^{\alpha^{(i)}},$$

where

$$\Lambda(d_1, \ldots, d_p) := \{\alpha \in \mathbb{N}^{n_1} \times \cdots \times \mathbb{N}^{n_p} \colon |\alpha^{(i)}| = d_i \text{ for all } i = 1, \ldots, p\}.$$

Define $M(f)$ as

$$M(f) := \sum_{\alpha \in \Lambda(d_1,\ldots,d_p)} \alpha! |a_\alpha|,$$

i.e., the absolute sum of the coefficients of the multiform f.

We will prove this result by induction on p, the number of group variables. The case $p = 1$ is the classical result by Proposition 3.2. Suppose that the result is true for the case $p = s - 1$ for some $s \geq 2$. In the following, we show that it is true for the case $p = s$.

We first rewrite the multiform f as

$$f(\mathbf{x}) = \sum_{|\gamma^{(s)}| = d_s} \gamma^{(s)}! f_{\gamma^{(s)}}(\hat{\mathbf{x}}_s) \left(\mathbf{x}^{(s)}\right)^{\gamma^{(s)}},$$

where $\gamma^{(s)} \in \mathbb{N}^{n_s}$, $\hat{\mathbf{x}}_s := (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(s-1)})$, and $f_{\gamma^{(s)}}(\hat{\mathbf{x}}_s)$ is a multiform in $\hat{\mathbf{x}}_s$ of degree d_i with respect to each $\mathbf{x}^{(i)}$ for $i = 1, \ldots, s-1$. Thus, $f(\mathbf{x})$ can be viewed as a form $h_{\hat{\mathbf{x}}_s}(\mathbf{x}^{(s)})$ in $\mathbf{x}^{(s)}$ with coefficients being multiforms in $\hat{\mathbf{x}}_s$. We have that

$$\lambda(h_{\hat{\mathbf{x}}_s}) \geq \lambda(f) := \min\{f(\mathbf{x}) \colon \mathbf{x} \in \Delta_{n_1} \times \cdots \times \Delta_{n_s}\} \text{ for all } \hat{\mathbf{x}}_s \in \Delta_{n_1} \times \cdots \times \Delta_{n_{s-1}}.$$

In addition, for each $\hat{\mathbf{x}}_s \in \Delta_{n_1} \times \cdots \times \Delta_{n_{s-1}}$, viewed as a form in $\mathbf{x}^{(s)}$,

$$\begin{aligned} L(h_{\hat{\mathbf{x}}_s}) &= \max\left\{\frac{1}{\gamma^{(s)}!}\left|\sum_{\alpha \in \Lambda(d_1,\ldots,d_s),\ \alpha^{(s)} = \gamma^{(s)}} \alpha! a_\alpha \prod_{i=1}^{s-1} \left(\mathbf{x}^{(i)}\right)^{\alpha^{(i)}}\right| \colon |\gamma^{(s)}| = d_s\right\} \\ &\leq \max\left\{\left|\sum_{\alpha \in \Lambda(d_1,\ldots,d_s),\ \alpha^{(s)} = \gamma^{(s)}} \alpha! a_\alpha \prod_{i=1}^{s-1} \left(\mathbf{x}^{(i)}\right)^{\alpha^{(i)}}\right| \colon |\gamma^{(s)}| = d_s\right\} \\ &\leq \max\left\{\sum_{\alpha \in \Lambda(d_1,\ldots,d_s),\ \alpha^{(s)} = \gamma^{(s)}} \alpha! |a_\alpha| \colon |\gamma^{(s)}| = d_s\right\} \\ &\leq M(f), \end{aligned}$$

where the first inequality follows from the fact $\gamma^{(s)}! \geq 1$ and the second from the fact $\hat{\mathbf{x}}_s \in \Delta_{n_1} \times \cdots \times \Delta_{n_{s-1}}$. Thus, by Proposition 3.2, for any positive integer

$$r_s > \frac{d_s(d_s-1)}{2}\frac{M(f)}{\lambda(f)} - d_s, \tag{3.4}$$

the polynomial

$$(\mathbf{e}^{\mathsf{T}}\mathbf{x}^{(s)})^{r_s} h_{\hat{\mathbf{x}}_s}(\mathbf{x}^{(s)})$$

is a polynomial with positive coefficients. Since the lower bound in (3.4) is independent of $\hat{\mathbf{x}}_s$, it is uniformly true for all $\hat{\mathbf{x}}_s \in \Delta_{n_1} \times \cdots \times \Delta_{n_{s-1}}$.

Now fix a positive integer r_s satisfying (3.4). Let

$$g_{\hat{\mathbf{x}}_s}(\mathbf{x}^{(s)}) = (\mathbf{e}^{\mathsf{T}}\mathbf{x}^{(s)})^{r_s} h_{\hat{\mathbf{x}}_s}(\mathbf{x}^{(s)}).$$

Then each coefficient of $g_{\hat{\mathbf{x}}_s}(\mathbf{x}^{(s)})$ is a multiform in $\hat{\mathbf{x}}_s$ of degree d_i with respect to each $\mathbf{x}^{(i)}$ for $i = 1, \ldots, s-1$, which is positive on the joint simplex $\Delta_{n_1} \times \cdots \times \Delta_{n_{s-1}}$. For each coefficient multiform, by the induction hypothesis, there exists $r_1, \ldots, r_{s-1}$ such that it has positive coefficients after multiplying $\prod_{i=1}^{s-1}(\mathbf{e}^{\mathsf{T}}\mathbf{x}^{(i)})^{r_i}$. Since there are only finitely many such coefficient multiforms and $\mathbf{e}^{\mathsf{T}}\mathbf{x}^{(i)}$'s are all polynomials with positive coefficients, a valid tuple $(r_1, \ldots, r_{s-1})$ for all the coefficients can be chosen. Therefore, a tuple $(r_1, \ldots, r_s)$ can be found such that all the coefficients of

$$\left[\prod_{i=1}^{s}(\mathbf{e}^{\mathsf{T}}\mathbf{x}^{(i)})^{r_i}\right] f(\mathbf{x})$$

are positive.

Finally, if a tuple $(r_1, \ldots, r_s)$ is valid for the conclusion, then it is easy to see that any other tuples with larger exponents are definitely valid as well. The conclusion then follows. □

The complexity of this Positivstellensatz can be investigated, as in [40, 49]. But we will leave it for future research since this article is focused on the zeroth order relaxation.

3.4. DNN relaxation. In this section, we will introduce a DNN relaxation method for solving problem (3.3).

Let $\mathbf{z} = (z_1, \ldots, z_n)^{\mathsf{T}}$ and

$$\mathbf{z}^{[s]} := \left(z_1^s, z_1^{s-1}z_2, z_1^{s-1}z_3, \ldots, z_1^{s-2}z_2^2, z_1^{s-2}z_2z_3, \ldots, z_2^s, \ldots, z_n^s\right)^{\mathsf{T}}$$

be the monomial basis of degree s in n variables. The order is the lexicographic order and $z_1 \succ z_2 \succ \cdots \succ z_n$. Note that the length of $\mathbf{z}^{[s]}$ is

$$\nu(s,n) := \binom{n+s-1}{s}.$$

Let $\tau = (\tau_1, \ldots, \tau_p) \in \mathbb{Z}_+^p$, $\mathbf{x} \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_p}$, and

$$\mathbf{x}^{[\tau]} := \left(\mathbf{x}^{(1)}\right)^{[\tau_1]} \otimes \cdots \otimes \left(\mathbf{x}^{(p)}\right)^{[\tau_p]}.$$

The monomials are ordered in the lexicographic order with $\mathbf{x}^{(1)} \succ \cdots \succ \mathbf{x}^{(p)}$ for the groups of variables. Let

$$\nu(\tau, n_1, \ldots, n_p) := \prod_{j=1}^{p} \nu(\tau_j, n_j)$$

and $A_\alpha \in \mathbb{R}^{\nu(\tau,n_1,\dots,n_p)\times\nu(\tau,n_1,\dots,n_p)}$ be the coefficient matrix of $\mathbf{x}^{[\tau]}\big(\mathbf{x}^{[\tau]}\big)^{\mathsf{T}}$ in the standard basis $\mathbf{x}^{[2\tau]}$, i.e.,

$$\mathbf{x}^{[\tau]}\big(\mathbf{x}^{[\tau]}\big)^{\mathsf{T}} = \sum_{\alpha\in\mathbb{N}^{n_1}_{2\tau_1}\times\cdots\times\mathbb{N}^{n_p}_{2\tau_p}} A_\alpha \mathbf{x}^\alpha, \tag{3.5}$$

where $\mathbb{N}^n_m := \{\gamma\in\mathbb{N}^n : |\gamma| := \gamma_1+\cdots+\gamma_n = m\}$.

Before stating the DNN relaxation problem, we first give a simple observation on the nonnegativity of moment sequences. Denote $\mathbf{d} := 2\tau = (2\tau_1,\dots,2\tau_p)$.

Proposition 3.4 (nonnegativity equivalence). *Let all notation be as above. Then, the coefficient matrices in the set $\{A_\alpha\}$ are nonnegative and orthogonal to each other, and thus*

$$\mathbf{y}\in\mathbb{R}^{\nu(\mathbf{d},n_1,\dots,n_p)}_{+} \textit{ if and only if } M(\mathbf{y}) := \sum_{\alpha\in\mathbb{N}^{n_1}_{d_1}\times\cdots\times\mathbb{N}^{n_p}_{d_p}} A_\alpha y_\alpha \geq \mathbf{0}. \tag{3.6}$$

Proof. According to the definition, each A_α is a nonnegative matrix. Therefore, the necessity is obvious. The sufficiency follows from the fact that $\langle A_\alpha, A_\gamma\rangle = 0$ for all $\alpha\neq\gamma$ and

$$\sum_{\alpha\in\mathbb{N}^{n_1}_{d_1}\times\cdots\times\mathbb{N}^{n_p}_{d_p}} A_\alpha = E,$$

where E is the matrix of all ones. □

Let $\mathbf{f}\in\mathbb{R}^{\nu(\mathbf{d},n_1,\dots,n_p)}$ be the coefficient vector of the polynomial $f(\mathbf{x}^{(1)},\dots,\mathbf{x}^{(p)})$ in the standard basis $\mathbf{x}^{[\mathbf{d}]}$, and let $\mathbf{g}\in\mathbb{R}^{\nu(\mathbf{d},n_1,\dots,n_p)}$ be that for the polynomial $g(\mathbf{x}) := \prod_{j=1}^p \big[\big(\mathbf{x}^{(j)}\big)^{\mathsf{T}}\mathbf{x}^{(j)}\big]^{\tau_j}$.

The basic idea of the SOS relaxation in [28] is by relaxing the rank characterization of a moment vector $\mathbf{y}\in\mathbb{R}^{\nu(\mathbf{d},n_1,\dots,n_p)}$. Without the nonnegativity constraint, it is classically relaxed as $M(\mathbf{y})\succeq\mathbf{0}$, i.e., the positive semidefiniteness of the moment matrix; see [28, 37, 43]. It can be shown that the dual problem under this method is an SDP problem obtained by representing a polynomial as an SOS. Therefore, this relaxation method is usually referred to as the *SOS relaxation.* With Proposition 3.4, a moment vector generated by a nonnegative vector is then naturally relaxed as $M(\mathbf{y})\succeq\mathbf{0}$ and $M(\mathbf{y})\geq\mathbf{0}$, i.e., the moment matrix is both positive semidefinite and componentwise nonnegative. A matrix that is both positive semidefinite and componentwise nonnegative is said to be DNN.

Naturally, a standard *DNN relaxation* of problem (3.3) is

$$\begin{aligned} f_{\text{dnn}} := \quad \min \quad & \langle \mathbf{f},\mathbf{y}\rangle \\ \text{s.t.} \quad & M(\mathbf{y})\succeq\mathbf{0}, \\ & M(\mathbf{y})\geq\mathbf{0}, \\ & \langle\mathbf{g},\mathbf{y}\rangle = 1,\ \mathbf{y}\in\mathbb{R}^{\nu(\mathbf{d},n_1,\dots,n_p)}, \end{aligned} \tag{3.7}$$

the dual of which is

$$\begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & \mathbf{f}-\gamma\mathbf{g}\in\Sigma^{+}_{\mathbf{d},n_1,\dots,n_p}, \end{aligned} \tag{3.8}$$

where

$$\Sigma^+_{\mathbf{d},n_1,\dots,n_p} := \left\{\mathbf{h}\colon \begin{array}{c} h(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]_{\mathbf{d}}, \\ h(\mathbf{x}) = \left(\mathbf{x}^{[\tau]}\right)^{\mathsf{T}}(S+T)(\mathbf{x}^{[\tau]}) \text{ for some } S \succeq \mathbf{0} \text{ and } T \geq \mathbf{0} \end{array}\right\}.$$

Here $\mathbb{R}[\mathbf{x}]_{\mathbf{d}} \subset \mathbb{R}[\mathbf{x}]$ is the set of multiforms being homogeneous of degree d_i with respect to $\mathbf{x}^{(i)}$ for all $i \in \{1,\dots,p\}$. Note that the cone of SOS

$$\Sigma_{\mathbf{d},n_1,\dots,n_p} = \left\{\mathbf{h}\colon h(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]_{\mathbf{d}},\ h(\mathbf{x}) = \left(\mathbf{x}^{[\tau]}\right)^{\mathsf{T}} S(\mathbf{x}^{[\tau]}) \text{ for some } S \succeq \mathbf{0}\right\}$$

is strictly contained in $\Sigma^+_{\mathbf{d},n_1,\dots,n_p}$. If there is no confusion, we sometimes will write $h(\mathbf{x}) \in \Sigma^+_{\mathbf{d},n_1,\dots,n_p}$ for a multiform $h(\mathbf{x})$, meaning its coefficient vector $\mathbf{h} \in \Sigma^+_{\mathbf{d},n_1,\dots,n_p}$.

The above DNN relaxation, together with Proposition 3.3, motivates a hierarchy of DNN relaxations for the optimization problem (3.3).

PROPOSITION 3.5. *Let $\eta \in \mathbb{N}^p$ and γ_η be the optimal value of the following problem:*

$$\gamma_\eta := \max\left\{\gamma\colon \prod_{i=1}^{p}(\mathbf{e}^{\mathsf{T}}\mathbf{x}^{(i)})^{2\eta_i}(f(\mathbf{x}) - \gamma g(\mathbf{x})) \in \Sigma^+_{\mathbf{d}+2\eta,n_1,\dots,n_p}\right\}. \tag{3.9}$$

Then

$$f_{\mathrm{dnn}} \leq \gamma_\eta \leq f_{\min} \text{ and } \gamma_\eta \leq \gamma_{\overline{\eta}} \text{ whenever } \eta \leq \overline{\eta}. \tag{3.10}$$

Moreover,

$$\gamma_\eta \to f_{\min} \text{ as } \min\{\eta_i\colon i = 1,\dots,p\} \to \infty.$$

Proof. The relations in (3.10) follows directly from the fact that each $\mathbf{e}^{\mathsf{T}}\mathbf{x}^{(i)}$ is a polynomial with positive coefficients.

Given an arbitrary $\epsilon > 0$, we know that the multiform $f(\mathbf{x}) - (f_{\min} - \epsilon)g(\mathbf{x})$ is positive on the nonnegative multisphere. Since $f(\mathbf{x}) - (f_{\min} - \epsilon)g(\mathbf{x})$ is a multiform, it is still positive on the joint simplex. Thus, it follows from Proposition 3.3 that there are positive integers r_i's such that

$$\prod_{i=1}^{p}(\mathbf{e}^{\mathsf{T}}\mathbf{x}^{(i)})^{2\eta_i}(f(\mathbf{x}) - (f_{\min} - \epsilon)g(\mathbf{x})) \in \Sigma^+_{\mathbf{d}+2\eta,n_1,\dots,n_p}$$

for all $\eta \geq \mathbf{r}$. Therefore, for all $\eta \geq \mathbf{r}$,

$$f_{\min} - \epsilon \leq \gamma_\eta \leq f_{\min}.$$

The conclusion thus follows. □

Proposition 3.5 gives the global convergence of the hierarchy of DNN relaxations (cf. (3.9)) for the problem (3.3), parallel to that of SOS relaxations (cf. [28, Theorem 3.4]). Problem (3.8) is the *zeroth order* DNN relaxation, i.e., $\eta = \mathbf{0}$ in (3.9).

In the following, some properties on the two matrix optimization problems (3.7) and (3.8) will be investigated.

LEMMA 3.6. *There exists a $\mathbf{y} \in \mathbb{R}^{\nu(\mathbf{d},n_1,\dots,n_p)}$ such that $M(\mathbf{y}) \succ \mathbf{0}$ and $M(\mathbf{y}) > \mathbf{0}$, i.e., the linear conic problem (3.7) is strictly feasible.*

Proof. Let λ be the Lebesgue measure on $\mathbb{S}^{n_1-1}\times\cdots\times\mathbb{S}^{n_p-1}$. Let μ be the normalized standard measure over the nonnegative multisphere $S := (\mathbb{R}^{n_1}_+\cap\mathbb{S}^{n_1-1})\times\cdots\times(\mathbb{R}^{n_p}_+\cap\mathbb{S}^{n_p-1})$, also known as the uniform probability measure on $\mathbb{S}^{n_1-1}_+\times\cdots\times\mathbb{S}^{n_p-1}_+$, defined as

$$\mu(A) := \frac{1}{\lambda(S)}\lambda(A\cap S) \text{ for any Borel set } A.$$

Define

$$y_\alpha := \int \mathbf{x}^\alpha \,\mathrm{d}\,\mu \quad \text{for all } \alpha\in\mathbb{N}^{n_1}_{d_1}\times\cdots\times\mathbb{N}^{n_p}_{d_p}$$

to be the truncated moment sequence of μ. It is obvious that $\mathbf{y} > \mathbf{0}$, and

$$\langle \mathbf{g},\mathbf{y}\rangle = \int g(\mathbf{x})\,\mathrm{d}\,\mu = 1,$$

since $g(\mathbf{x})\equiv 1$ over the support S of μ.

For any $f(\mathbf{x})\in\mathbb{R}[\mathbf{x}]_\tau$, we have

$$\mathbf{f}^\mathsf{T} M(\mathbf{y})\mathbf{f} = \int f(\mathbf{x})^2\,\mathrm{d}\,\mu.$$

Since the support of μ is the nonnegative orthant part of the multisphere, we can then conclude from the fact $\mathbf{f}^\mathsf{T}M(\mathbf{y})\mathbf{f} = 0$ that

$$f(\mathbf{x}) = 0 \text{ for all } \mathbf{x}\in S := (\mathbb{R}^{n_1}_+\cap\mathbb{S}^{n_1-1})\times\cdots\times(\mathbb{R}^{n_p}_+\cap\mathbb{S}^{n_p-1}).$$

Since f is multihomogeneous, we immediately have that

$$f(\mathbf{x}) = 0 \text{ for all } \mathbf{x}\in\mathbb{R}^{n_1}_+\times\cdots\times\mathbb{R}^{n_p}_+.$$

Note that $\mathbb{R}^{n_1}_+\times\cdots\times\mathbb{R}^{n_p}_+$ is a set with the Zariski closure being the whole space $\mathbb{R}^{n_1}\times\cdots\times\mathbb{R}^{n_p}$. We conclude that $f\equiv 0$. Thus, the matrix $M(\mathbf{y})$ is positive definite. ◻

LEMMA 3.7. *There exist a scalar γ, a matrix $S\succ\mathbf{0}$, and a matrix $T>\mathbf{0}$ such that $f(\mathbf{x})-\gamma g(\mathbf{x}) = \big(\mathbf{x}^{[\tau]}\big)^\mathsf{T}(S+T)(\mathbf{x}^{[\tau]})$, i.e., the linear conic problem* (3.8) *is strictly feasible.*

Proof. Note that there exists a nonnegative diagonal matrix D such that

$$g(\mathbf{x}) = \big(\mathbf{x}^{[\tau]}\big)^\mathsf{T} D(\mathbf{x}^{[\tau]})$$

with the minimum diagonal element being one. Thus, $D\succ\mathbf{0}$. The result follows immediately if a sufficiently small $\gamma<0$ is chosen. ◻

PROPOSITION 3.8. *Both* (3.7) *and* (3.8) *are solvable, and there is no duality gap.*

Proof. Both (3.7) and (3.8) have strictly feasible solutions by Lemmas 3.6 and 3.7, respectively. The conclusion then follows from standard duality theory for linear conic optimization problems (cf. [3]). ◻

PROPOSITION 3.9 (exact relaxation). *Let $d_i = 2\tau_i$ for all $i = 1,\ldots,p$. If* (3.7) *has an optimal solution $\mathbf{y}^*$ such that*

$$\operatorname{rank}(M(\mathbf{y}^*)) = 1, \tag{3.11}$$

then the relaxation is tight, i.e., $f_{\min} = f_{\mathrm{dnn}}$, and an optimal solution for (3.1) *can be extracted from $\mathbf{y}^*$.*

Proof. It follows from [10, 46] that $\mathbf{y}^*$ is a monomial vector in this situation. Let

$$M(\mathbf{y}^*) = \mathbf{x}_*^{[\tau]}\big(\mathbf{x}_*^{[\tau]}\big)^{\mathsf{T}}$$

with $\mathbf{x}_* = (\mathbf{x}_*^{(1)}, \dots, \mathbf{x}_*^{(p)})$. Then, we have from $M(\mathbf{y}^*) \geq \mathbf{0}$ that

$$\mathbf{x}_*^{(i)} \geq \mathbf{0} \ \text{ or } \ \ \mathbf{x}_*^{(i)} \leq \mathbf{0}$$

for each $i = 1, \dots, p$. Since each d_i is even, the monomial vector $\mathbf{z}^*$ with

$$M(\mathbf{z}^*) = \mathbf{w}_*^{[\tau]}\big(\mathbf{w}_*^{[\tau]}\big)^{\mathsf{T}} \text{ and } \mathbf{w}_* = (|\mathbf{x}_*^{(1)}|, \dots, |\mathbf{x}_*^{(p)}|)$$

satisfies $\mathbf{z}^* = \mathbf{y}^*$. Therefore, the results follow. □

We will see from later numerical experiments that (3.11) is a *typical property*, i.e., it holds with high probability if we randomly generate f from a continuous probability distribution.

3.5. DNN reformulation. In this section, we formulate (3.7) as a linear optimization problem over the cone of DNN matrices more explicitly. We shall eliminate the variable vector $\mathbf{y}$ by exploiting the hidden constraints on the matrix $M(\mathbf{y})$. We have already shown that the $\nu(\mathbf{d}, n_1, \dots, n_p)$ matrices (cf. (3.5))

$$A_\alpha : \alpha \in \mathbb{N}_{2\tau_1}^{n_1} \times \cdots \times \mathbb{N}_{2\tau_p}^{n_p}$$

are orthogonal to each other. Let

$$\big\{B_i : 1 \leq i \leq \mu(\mathbf{d}, n_1, \dots, n_p)\big\}$$

with $\mu(\mathbf{d}, n_1, \dots, n_p) := \nu(\tau, n_1, \dots, n_p)(\nu(\tau, n_1, \dots, n_p) + 1)/2 - \nu(\mathbf{d}, n_1, \dots, n_p)$ be the set of matrices that are orthogonal to each other such that

$$\big\{A_\alpha : \alpha \in \mathbb{N}_{2\tau_1}^{n_1} \times \cdots \times \mathbb{N}_{2\tau_p}^{n_p}\big\} \cup \big\{B_i : i = 1, \dots, \mu(\mathbf{d}, n_1, \dots, n_p)\big\}$$

forms an orthogonal basis of the space of $\nu(\tau, n_1, \dots, n_p) \times \nu(\tau, n_1, \dots, n_p)$ real symmetric matrices. Let

$$\mathbf{w} \in \mathbb{R}^{\nu(\mathbf{d}, n_1, \dots, n_p)} \text{ with } w_\alpha = \langle A_\alpha, A_\alpha\rangle \text{ for all } \alpha.$$

Then the problem (3.7) can be equivalently reformulated as

$$\begin{aligned} f_{\mathrm{dnn}} := \quad \min \quad & \Big\langle \textstyle\sum_{\alpha \in \mathbb{N}_{2\tau_1}^{n_1} \times \cdots \times \mathbb{N}_{2\tau_p}^{n_p}} \frac{f_\alpha}{w_\alpha} A_\alpha, X \Big\rangle \\ \text{s.t.} \quad & \langle B_i, X\rangle = 0, \ i = 1, \dots, \mu(\mathbf{d}, n_1, \dots, n_p), \\ & \Big\langle \textstyle\sum_{\alpha \in \mathbb{N}_{2\tau_1}^{n_1} \times \cdots \times \mathbb{N}_{2\tau_p}^{n_p}} \frac{g_\alpha}{w_\alpha} A_\alpha, X \Big\rangle = 1, \\ & X \succeq \mathbf{0}, \ X \geq \mathbf{0}. \end{aligned} \tag{3.12}$$

The optimization problem (3.12) is classified as a DNN problem, since it requires the matrix variable X to be both positive semidefinite and component-wisely nonnegative. As a linear conic problem, it can be reformulated as a standard SDP problem by introducing a new variable Y and adding the constraints that $X - Y = \mathbf{0}$ so that the original DNN conic constraint can be replaced by $X \succeq \mathbf{0}$ and $Y \geq \mathbf{0}$. However, this

TABLE 1

$(\mathbf{d}, n_1, \ldots, n_p)$: (number of equations; dimension of the matrix variable) of (3.12) for several $\mathbf{d}$'s and $(n_1, \ldots, n_p)$'s.

$(\mathbf{d}, n_1, \ldots, n_p)$:(# eq.; dim)	$(\mathbf{d}, n_1, \ldots, n_p)$:(# eq.; dim)
(4, 100):(8,332,501; 5,050)	((2,2,2), 20,20,20):(22,743,001; 8,000)
(4,150):(42,185,626; 11,325)	((2,2,3),15,10,10):(42,403,351; 9,900)
((2,2), 100,100):(24,502,501; 10,000)	((2,2,2,2), 10,10,10,10):(40,854,376; 10,000)
((2,3),50,20):(53,158,876; 11,550)	((2,2,2,3),6,6,6,8):(42,659,866; 9,720)

reformulation introduces too many new equality constraints which not only make the resulting standard SDP problem computationally much more expensive to solve but also make it likely to encounter numerical difficulties when solving the standard SDP reformulation since it is likely to be constraint degenerate (cf. [66]).

Table 1 gives some information on the sizes of the DNN relaxation problem (3.12) for different $\mathbf{d}$ and $n_1, \ldots, n_p$. When d_i is odd, we use the technique in section 3.1 to transform it into the standard formulation involving only even orders. In this table, **# eq.** means the number of equality constraints, and **dim** means the dimension of the matrix variable. For a general DNN problem, on a laptop, the current state-of-the-art solver can solve problems with the matrix dimension around 5000 and several millions of equality constraints. Therefore, except the first case of a quartic polynomial in 100 variables, all the other cases are almost hopeless to solve at present [65]. On the other hand, all the cases are tensors with small to moderate dimensions, showing the difficulty of solving the problem (3.1) globally from another perspective.

3.6. Worst case approximation bound. In this section, we present a worst case approximation bound for f_{dnn}.

Given a positive integer n, define the matrix Θ_n by

$$\Theta_n := \int_{\mathbb{S}_+^{n-1}} \mathbf{x}^{[n]} \left(\mathbf{x}^{[n]}\right)^{\mathsf{T}} \mathrm{d}\,\mu(\mathbf{x}),$$

where $\mu(\mathbf{x})$ is the uniform probability measure on $\mathbb{S}_+^{n-1}$. It is easy to see that Θ_n is positive definite, since the set $\mathbb{S}_+^{n-1}$ is of dimension $n-1$ and the monomial vector $\mathbf{x}^{[n]}$ consists of homogeneous monomials. Let

$$\delta_{n_1,\ldots,n_p} := \prod_{i=1}^{p} \sqrt{\lambda_{\min}(\Theta_{n_i})},$$

where $\lambda_{\min}(\Theta_{n_i})$ is the smallest eigenvalue of the matrix Θ_{n_i}. Since each Θ_{n_i} is positive definite, we have that $\delta_{n_1,\ldots,n_p} > 0$.

Since the set $\mathbb{S}_+^{n-1}$ is involved in this article instead of $\mathbb{S}^{n-1}$, $\lambda_{\min}(\Theta_{n_i})$ is different from those given in [37, Table 1]. For example,

$$\Theta_2 = \frac{1}{8\pi} \begin{bmatrix} 3\pi & 4 & \pi \\ 4 & 2\pi & 4 \\ \pi & 4 & 3\pi \end{bmatrix}.$$

Consequently, $\delta_2 = \sqrt{\lambda_{\min}(\Theta_2)} = 0.4849$, which is different from 0.5 in [37] with respect to $\mathbb{S}^{n-1}$.

Let $f_{\max}$ and $f_{\min}$ be the maximum and minimum values of the objective function f over the feasible set of problem (3.1), and $f_{\max} > f_{\min}$. We then have the next proposition, whose proof is almost the same as that in [37, Theorem 3.4].

PROPOSITION 3.10. *Suppose that $n_i \geq d_i$ for all $i \in \{1, \dots, p\}$ and all notation are as above. Then we have that*

$$1 \ \leq \ \frac{f_{\max} - f_{dnn}}{f_{\max} - f_{\min}} \ \leq \ \frac{1}{\delta_{d_1,\dots,d_p}} \sqrt{\binom{n_1}{d_1} \cdots \binom{n_p}{d_p}}. \tag{3.13}$$

With δ_2 computed as above, we have that for a biquadratic form over the intersection of the multisphere and the nonnegative orthant

$$1 \ \leq \ \frac{f_{\max} - f_{dnn}}{f_{\max} - f_{\min}} \ \leq \ 4.2535 \sqrt{\binom{n_1}{2}\binom{n_2}{2}}.$$

The upper bound is slightly different from that with respect to the multisphere; see [37, Corollary 3.5].

If the polynomial is sparse, i.e., with fewer terms in its polynomial expansion, then an improved worst case approximation bound in terms of the number of monomials $\Omega(f)$ can be derived as in [37, section 4]. In particular, if the polynomial is a monomial or the number of monomials is bounded by a constant, then a constant worst case approximation bound, independent of the problem dimensions, can be given.

3.7. Solution extraction for even order tensors. Let $\mathbf{y}^*$ be an optimal solution for (3.7). By Proposition 3.4, $\mathbf{y}^* \geq \mathbf{0}$. Let

$$y^*_{2\gamma} := \max\{y^*_{2\mu} : \mathbf{x}^{\mu} \in \mathbf{x}^{[\tau]}\}.$$

Since the set $\{y^*_{2\mu} : \mathbf{x}^{\mu} \in \mathbf{x}^{[\tau]}\}$ forms the diagonal elements of the positive semidefinite matrix $M(\mathbf{y}^*)$ and $\mathbf{y}^* \neq 0$, we have that

$$y^*_{2\gamma} > 0.$$

Denote

$$\gamma := (\gamma^1, \dots, \gamma^p)$$

with

$$\gamma^i := (\gamma^i_1, \dots, \gamma^i_{n_i})$$

for all $i = 1, \dots, p$. Then $\gamma^i \neq \mathbf{0}$ for all $i = 1, \dots, p$. Let

$$\gamma^i_{k_i} := \max\{\gamma^i_1, \dots, \gamma^i_{n_i}\}.$$

Define

$$\mathbf{z}^{(i)}_* := \left(y^*_{\gamma+(\gamma^1,\dots,\gamma^{i-1},\gamma^i-\mathbf{e}^{(i)}_{k_i}+\mathbf{e}^{(i)}_1,\gamma^{i+1},\dots,\gamma^p)}, \dots, y^*_{\gamma+(\gamma^1,\dots,\gamma^{i-1},\gamma^i-\mathbf{e}^{(i)}_{k_i}+\mathbf{e}^{(i)}_{n_i},\gamma^{i+1},\dots,\gamma^p)}\right)^{\mathsf{T}},$$

$$\mathbf{x}^{(i)}_* := |\mathbf{z}^{(i)}_*| / \|\mathbf{z}^{(i)}_*\| \quad \text{for all } i = 1, \dots, p.$$

The approximation solution is then

$$\mathbf{x}_* = (\mathbf{x}^{(1)}_*, \dots, \mathbf{x}^{(p)}_*),$$

and the approximation value is

$$f_{\text{app}} := f(\mathbf{x}_*^{(1)}, \dots, \mathbf{x}_*^{(p)}).$$

If $\operatorname{rank}(M(\mathbf{y}^*)) = 1$, then it holds that (cf. Proposition 3.9)

$$M(\mathbf{y}^*) = \mathbf{x}_*^{[\tau]}\big(\mathbf{x}_*^{[\tau]}\big)^{\mathsf{T}}.$$

3.8. Solution extraction for odd order tensors. Let $\mathbf{y}^*$ be an optimal solution for (3.7). Suppose that the tensor space is $\operatorname{Sym}(\otimes^{\alpha_1}\mathbb{R}^{n_1}) \otimes \cdots \otimes \operatorname{Sym}(\otimes^{\alpha_p}\mathbb{R}^{n_p})$, and without loss of generality that $\alpha_1, \dots, \alpha_q$ are odd for some $q \le p$. Let $\mathbf{d} = (\alpha_1 + 1, \dots, \alpha_q + 1, \alpha_{q+1}, \dots, \alpha_p)$. By the scheme in section 3.1, we have that

$$\mathbf{y}^* \in \mathbb{R}^{\nu(\mathbf{d}, n_1+1, \dots, n_q+1, n_{q+1}, \dots, n_p)}.$$

Let

$$y_\gamma^* := \max\big\{y_\mu^* : \mu = \big((\mu^1, 1), \dots, (\mu^q, 1), \mu^{q+1}, \dots, \mu^p\big) \text{ with } \mu^i \in \mathbb{N}_{\alpha_i}^{n_i}\big\}.$$

If $y_\gamma^* = 0$, it follows from section 2.1 that zero is the best approximation solution, since in this case the optimal value of (3.7) is zero. In the following, we assume that

$$y_\gamma^* > 0.$$

Denote

$$\gamma := (\gamma^1, \dots, \gamma^p)$$

with

$$\gamma^i := (\gamma_1^i, \dots, \gamma_{n_i}^i, 1)$$

for all $i = 1, \dots, q$, and

$$\gamma^i := (\gamma_1^i, \dots, \gamma_{n_i}^i)$$

for all $i = q+1, \dots, p$. Let

$$\gamma_{k_i}^i := \max\{\gamma_1^i, \dots, \gamma_{n_i}^i\}.$$

Define

$$\mathbf{z}_*^{(i)} := (y^*_{(\gamma^1, \dots, \gamma^{i-1}, \gamma^i - \mathbf{e}_{k_i}^{(i)} + \mathbf{e}_1^{(i)}, \gamma^{i+1}, \dots, \gamma^p)}, \dots, y^*_{(\gamma^1, \dots, \gamma^{i-1}, \gamma^i - \mathbf{e}_{k_i}^{(i)} + \mathbf{e}_{n_i+1}^{(i)}, \gamma^{i+1}, \dots, \gamma^p)})^{\mathsf{T}},$$

$$\tilde{\mathbf{x}}_*^{(i)} := |\mathbf{z}_*^{(i)}| / \|\mathbf{z}_*^{(i)}\| \quad \text{for all } i = 1, \dots, q,$$

and

$$\mathbf{z}_*^{(i)} := (y^*_{(\gamma^1, \dots, \gamma^{i-1}, \gamma^i - \mathbf{e}_{k_i}^{(i)} + \mathbf{e}_1^{(i)}, \gamma^{i+1}, \dots, \gamma^p)}, \dots, y^*_{(\gamma^1, \dots, \gamma^{i-1}, \gamma^i - \mathbf{e}_{k_i}^{(i)} + \mathbf{e}_{n_i}^{(i)}, \gamma^{i+1}, \dots, \gamma^p)})^{\mathsf{T}},$$

$$\mathbf{x}_*^{(i)} := |\mathbf{z}_*^{(i)}| / \|\mathbf{z}_*^{(i)}\| \quad \text{for all } i = q+1, \dots, p.$$

The approximation solution for the extended problem is then

$$\tilde{\mathbf{x}}_* = (\tilde{\mathbf{x}}_*^{(1)}, \dots, \tilde{\mathbf{x}}_*^{(q)}, \mathbf{x}_*^{(q+1)}, \dots, \mathbf{x}_*^{(p)}).$$

Let

$$\tilde{\mathbf{x}}_*^{(i)} = (\mathbf{x}^{(i)}, t_i) \text{ for } i = 1, \dots, q.$$

For $i = 1, \dots, q$, if $t_i \neq 1$, then we take

$$\mathbf{x}_*^{(i)} := \mathbf{x}^{(i)} / \|\mathbf{x}^{(i)}\|;$$

otherwise, we conclude that the best approximating nonnegative rank-one tensor is the zero tensor.

4. Numerical experiments.

In this section, we present some preliminary numerical experiments for solving problem (3.1) via the DNN relaxation method developed in section 3. All the tests were conducted on a Lenovo laptop with 32GB RAM and 2.7GHz CPU running 64bit Windows operation system. All codes were written in MATLAB with some subroutines in C++. All the linear matrix conic problems were solved by the solver SDPNAL+ [61, 65].

The numerical results will be divided into four subsections, which consist of instances of the best nonnegative rank-one approximations of tensors (cf. sections 4.1, 4.2, and 4.3) and the copositivities of tensors (cf. section 4.4).

Given a tensor $\mathcal{A}$, we use f_{dnn} to denote the optimal value of the corresponding DNN relaxation problem. The approximation solution $\mathbf{x}$ of problem (3.1) is extracted according to sections 3.7 and 3.8. Then $\lambda\mathbf{x}^{\otimes\mathbf{d}}$ with λ giving by (2.4) is the best nonnegative rank-one approximation found. Therefore, $f_{\text{app}} := \lambda$ is the approximate optimal value of (3.1) found by the method. We use the relative approximation error

$$\mathbf{apperr}(\mathcal{A}) := \begin{cases} \frac{|f_{\text{dnn}} - f_{\text{app}}|}{|f_{\text{dnn}}|} & \text{if } f_{\text{dnn}} \neq 0, \\ |f_{\text{app}}| & \text{otherwise} \end{cases}$$

and the relative approximation error with respect to the problem data size

$$\mathbf{apperrnm}(\mathcal{A}) := \frac{|f_{\text{dnn}} - f_{\text{app}}|}{\|\mathcal{A}\|}$$

to measure the approximation quality. Note that due to the accuracy tolerance (the default is 10^{-6}) setting in solving the DNN relaxation problem of (3.1), even if the matrix $M(\mathbf{y}^*)$ for the optimal $\mathbf{y}^*$ of (3.7) has rank one (thus the approximation is tight), we may still have $f_{\text{dnn}} \neq f_{\text{app}}$. But their difference should have the same magnitude as the accuracy tolerance used.

Numerically, we regard the relaxation to be tight (e.g., when $\text{rank}(M(\mathbf{y}^*)) = 1$) whenever the second largest singular value of $M(\mathbf{y}^*)$ is smaller than 1.0×10^{-6}.

4.1. Comparisons.

This section presents the comparisons of the performance among different formulations of DNN problems by different solvers for efficiency, and between AO-ADMM proposed in [24] and the DNN relaxation for global optimality.

4.1.1. Formulations and SDP solvers.

It is already mentioned that the DNN problems can be formulated as standard SDP problems by introducing extra variables and constraints, and the resulting standard SDP problems can also be solved directly by the solver SDPNAL [66] (cf. section 3.5). We will call this approach the *naïve SDPNAL*. On the other hand, we can solve the DNN relaxation via the dual formulation (3.8). Moreover, during the implementation, we can take a nonnegative vector of dimension $\nu(\mathbf{d}, n_1, \dots, n_p)$ to represent the linear equality constraint, instead of the nonnegative matrix of size $\nu(\tau, n_1, \dots, n_p) \times \nu(\tau, n_1, \dots, n_p)$ (cf. see the definition of $\Sigma^+_{\mathbf{d},n_1,\dots,n_p}$ in section 3.4). By doing so, the number of nonnegative variables can be reduced dramatically; see Table 2. This approach is termed *SDPNAL*.

On the other hand, the DNN relaxation can be solved more efficiently by the DNN-focused solver SDPNAL+ [61, 65]. Thus, we apply SDPNAL+ for solving our DNN problems (3.12). In the following, we have a comparison among the performance of *naïve SDPNAL*, *SDPNAL*, and *SDPNAL+* in solving Example 4.1. The primary purpose is to give recommendation on the approaches for solving the DNN relaxations. The results are reported in Table 2. We can see the superiority of *SDPNAL* and *SDPNAL+* over *naïve SDPNAL*. Moreover *SDPNAL+* performs better than

TABLE 2
Comparisons on Example 4.1.

m	n	# sdp	Naïve SDPNAL		SDPNAL		SDPNAL+	
			#non; # con	Time	#non; # con	Time	# con	Time
3	5	216	23436; 37612	29.9	9261; 9261	6.4	14176	7.9
3	8	729	266085; 441046	27:1.5	91125; 91125	2:21.9	174961	2:23.2
3	10	1331	886446; 1485397	2:55:5.8	287496; 287496	F	598951	20:16.2
3	12	2197	2414503; 4075436	3:51:46.8	753571; 753571	43:16.4	1660933	49:53.5
4	4	625	195625; 340626	7:43.7	50625; 50625	24.1	145001	28.8
4	5	1296	840456; 1486432	1:18:10.8	194481; 194481	8:44.4	645976	4:39.3
4	6	2401	2883601; 5152547	4:24:32.7	614656; 614656	30:33.5	2268946	18:10.7
5	3	1024	524800; 949601	1:4:52.0	100000; 100000	1:56.2	424801	1:46.9
6	2	729	266085; 485515	1:54.7	46656; 46656	41.9	219430	32.9
7	2	2187	2392578; 4505221	3:59:18.0	279936; 279936	6:17.9	2112643	12:4.6

SDPNAL for most cases with large problem sizes. Note that *SDPNAL* fails in the case $(3, 10)$. In the tables, *Time* denotes the computation time consumed in the format of *hours:minutes:seconds*, *# sdp* the dimension of the positive semidefinite matrix variable, *# non* the number of nonnegative constraints, and *# con* the number of equality constraints.

Example 4.1. This example comes from [41, Example 3.16]. The tensor $\mathcal{A} \in \otimes^m \mathbb{R}^n$ has the entries

$$a_{i_1 \dots i_m} = \sum_{j=1}^{m} (-1)^{j+1} \cdot j \cdot \exp(-i_j).$$

4.1.2. AO-ADMM. In this subsection, we compare the method proposed by Huang, Sidiropoulos, and Liavas in [24] (see also the excellent survey [58]) with the DNN relaxation approach proposed for a set of examples. The method in [24] is called *alternating optimization (AO)* with subproblems being solved by *alternating direction method of multipliers (ADMM)*. This method is designed for the case of nonnegative tensor approximation with $\alpha_1 = \cdots = \alpha_p = 1$ (cf. section 2). For the sake of simplicity, we take $p = 3$ for an illustration. It can be extended to the more general case $p > 3$ directly. The method is terminated when the improvements of both the iteration and the objective function value are within 10^{-6}.

Example 4.2. This example comes from [41, Example 3.14]. The tensor $\mathcal{A} \in \otimes^3 \mathbb{R}^n$ has the entries

$$a_{ijk} = \cos\big(i + 2j + 3k\big).$$

The numerical computations are recorded in Table 3, in which *T(DNN)* denotes the computation time in the format of *hours:minutes:seconds* consumed for solving the corresponding problem by the DNN approach and λ(DNN) the norm of the best rank-one approximation tensor found, i.e., f_{app}. *T(AO)* and λ(AO) are for the AO-ADMM method. f_{dnn} is the optimal value of the DNN relaxation found.[1] We see that the DNN relaxation method finds a nonnegative rank-one approximation with much larger norm (cf. λ) in most cases.

[1] By the formulation (2.5), there is a sign change of the objective function in the case of best nonnegative rank-one approximation. Thus, the columns of f_{dnn} in this case are actually $-f_{\text{dnn}}$ of that given by (3.12).

Table 3
Computational results for Example 4.2.

n	T(DNN)	λ(DNN)	T(AO)	λ(AO)	f_{dnn}	**apperr**	**apperrnm**
2	0.12	1.2208	0.005	0.8439	1.2208	6.75×10^{-6}	3.85×10^{-6}
3	0.29	1.7342	0.002	1.0715	1.7342	1.90×10^{-5}	8.96×10^{-6}
4	1.67	2.4413	0.038	2.4417	2.4508	3.88×10^{-3}	1.67×10^{-3}
5	3.33	2.9581	0.119	2.9309	3.0911	4.30×10^{-2}	1.69×10^{-2}
6	5.95	2.8465	0.011	2.7676	3.7989	0.251	9.16×10^{-2}
7	26.41	4.0467	0.017	3.5431	4.6857	0.136	4.90×10^{-2}
8	46.12	5.0071	0.186	4.3912	5.7113	0.123	4.41×10^{-2}
9	3:33.83	6.0120	0.211	4.8147	6.7987	0.116	4.11×10^{-2}
10	11:52.01	6.9082	0.504	5.1320	7.4669	0.748	2.50×10^{-2}

Table 4
Computational results for Example 4.3.

n	T(DNN)	λ(DNN)	T(AO)	λ(AO)	f_{dnn}	**apperr**	**apperrnm**
2	0.207	36.90	0.002	36.62	36.90	9.06×10^{-6}	8.85×10^{-6}
3	0.606	166.6	0.005	158.0	166.6	5.54×10^{-8}	5.22×10^{-8}
4	3.578	636.9	0.006	521.0	636.9	1.44×10^{-8}	1.32×10^{-8}
5	7.164	2229.9	0.007	1580.2	2229.9	3.11×10^{-8}	2.79×10^{-6}
6	5.874	7408.07	0.006	4817.8	7408.02	7.23×10^{-6}	6.38×10^{-6}
7	51.01	23785.9	0.007	14706.2	23785.9	2.10×10^{-7}	1.83×10^{-7}
8	2:26.62	74495.0	0.007	44287.5	74495.0	1.03×10^{-6}	8.87×10^{-7}
9	2:50.34	229151.2	0.006	131688.4	229150.8	1.43×10^{-6}	1.22×10^{-6}

Example 4.3. This example is a modification of [41, Example 3.16]. The tensor $\mathcal{A} \in \otimes^3\mathbb{R}^n$ has the entries

$$a_{ijk} = \exp(i) - 2\exp(j) + 3\exp(k).$$

The numerical computations are recorded in Table 4. In each case, the DNN relaxation finds a global optimal solution. We can see from Table 4 that the gap between the solution found by AO-ADMM and the global optimal solution is large in most cases.

Example 4.4. This example comes from [41, Example 3.18]. The tensor $\mathcal{A} \in \otimes^3\mathbb{R}^n$ has the entries

$$a_{ijk} = \tan\left(i - \frac{j}{2} + \frac{k}{3}\right).$$

The numerical computations are recorded in Table 5. Similar conclusions as in Example 4.3 can be drawn from Table 5. We note that for the case $n = 4$, AO-ADMM returns the zero tensor and hence fails to find a valid nonnegative approximation solution.

Example 4.5. This example tests a tensor $\mathcal{A} \in \mathbb{R}^m \otimes \mathbb{R}^n \otimes \mathbb{R}^k$ with the entries being randomly chosen from either $[0, 1]$ (the case in the upper six rows of Table 6) or $[-1, 1]$ (the last six rows of Table 6). For each case, ten simulations are drawn with T(DNN) being the mean computation time consumed for DNN. We can observe that the DNN approach always finds a better objective function value f_{dnn} than f_{ao}

Table 5
Computational results for Example 4.4.

n	T(DNN)	λ(DNN)	T(AO)	λ(AO)	f_{dnn}	**apperr**	**apperrnm**
2	0.114	4.1462	0.023	2.7232	4.1462	9.79×10^{-6}	3.33×10^{-6}
3	0.696	14.4480	0.002	4.2858	14.4480	9.19×10^{-8}	5.52×10^{-8}
4	1.205	15.3004	2.912	0.0000	15.3004	6.14×10^{-7}	3.16×10^{-7}
5	17.261	22.1107	0.041	3.8793	25.3944	0.129	5.99×10^{-2}
6	6.681	22.0960	0.093	22.0068	27.9674	0.210	8.23×10^{-2}
7	1:05.47	56.0168	4.373	24.9464	56.0168	1.80×10^{-7}	8.50×10^{-8}
8	42.086	51.9399	0.112	15.0330	64.5472	0.195	7.93×10^{-2}

Table 6
Computational results for random examples. Ten instances are considered for each case.

m	n	k	T(DNN)	T(AO)	mean(rt)	max(rt)	min(rt)	**bet**	**tight**
3	3	3	0.5803	0.9944	0.0170	0.0361	0.0020	10	10
3	4	5	1.1348	0.8587	0.0120	0.0245	0.0020	10	10
5	5	5	2.3505	0.0411	0.0110	0.0209	0.0055	10	10
5	6	7	5.1839	0.0059	0.0064	0.0103	0.0035	10	10
7	7	7	21.3171	0.0040	0.0083	0.0152	0.0027	10	10
6	7	8	16.6694	0.0036	0.0085	0.0140	0.0050	10	10
2	2	2	0.2609	0.1032	0.3826	1.0000	0.0000	10	10
3	3	3	0.5481	1.0151	0.5623	1.0000	0.0772	10	10
2	3	4	0.6125	0.8665	0.5385	1.0000	0.0033	10	10
3	4	5	1.4403	2.5975	0.0659	0.2169	0.0010	10	10
5	5	5	2.6163	3.1757	0.2227	0.5908	0.0031	10	10
4	5	6	7.7050	2.4806	0.2613	0.8483	0.0728	10	10

of AO-ADMM. Actually, the DNN approach always finds a global optimal solution. Thus, we use

$$\text{rt} := \frac{f_{\text{dnn}} - f_{\text{ao}}}{f_{dnn}} \tag{4.1}$$

to measure the relative gap between the objective function value returned by AO-ADMM and the optimal function value. mean(rt), max(rt) and min(rt) are respectively, the mean, maximum, and minimum among the ten simulations. bet counts the number of times DNN being better than AO-ADMM in objective function values, and tight counts the number of times DNN finding a global optimal solution.

For the case of nonnegative tensors, AO-ADMM performed quite well in finding approximate global optimal solutions. (We know that randomly generated tensors tend to be easy for alternating minimization [23].) However, when the tensor has negative components, the gap becomes large. Moreover, for tensors with negative components, we see from Table 6 that even the computation time consumed by DNN is not much worse than that of AO-ADMM.

4.2. Relaxation hierarchy. In this section, we present two concrete examples for our DNN relaxation hierarchy (cf. Proposition 3.5), particularly on problems for which the zeroth order relaxation is not tight. If a higher order relaxation is used, we solve the dual problem (3.9).

Table 7
Computational results for Example 4.6.

η	f_{app}	f_{dnn}	**apperr**	**apperrnm**	$\mathbf{x}^*$	Time
0	0.6416	0.6999	5.838×10^{-2}	2.592×10^{-2}	$(0.9328, 0, 0.3603)^\mathsf{T}$	0.124
1	0.6795	0.6800	5.064×10^{-4}	2.248×10^{-4}	$(0.8892, 0, 0.4576)^\mathsf{T}$	0.797
2	0.6798	0.6798	5.832×10^{-6}	2.589×10^{-6}	$(0.8848, 0, 0.4660)^\mathsf{T}$	0.937

Table 8
Computational results for Example 4.7.

η	f_{app}	f_{dnn}	**apperr**	**apperrnm**	$\mathbf{x}^*$	Time
0	2.000	2.005	2.292×10^{-3}	9.298×10^{-4}	$(1, 0, 0)^\mathsf{T}$	1.031
1	2.000	2.001	5.343×10^{-4}	2.163×10^{-4}	$(0, 1, 0)^\mathsf{T}$	1.719
2	2.000	2.000	1.019×10^{-5}	4.125×10^{-6}	$(0, 1, 0)^\mathsf{T}$	2.390

Example 4.6. This example comes from [25, Example 1]. It is a tensor $\mathcal{A}$ in $\mathrm{Sym}(\otimes^4\mathbb{R}^3)$ with the independent entries being

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

The zeroth order relaxation is not tight, while the second order relaxation is tight. The computational result is given in Table 7.

Example 4.7. This example comes from [41, Example 3.8]. It is a tensor $\mathcal{A}$ in $\mathrm{Sym}(\otimes^6\mathbb{R}^3)$ with the nonzero independent entries being

$$a_{111111} = 2, a_{111122} = 1/3, a_{111133} = 2/5, a_{112222} = 1/3, a_{112233} = 1/6,$$
$$a_{113333} = 2/5, a_{222222} = 2, a_{222233} = 2/5, a_{223333} = 2/5, a_{333333} = 1.$$

This is a nonnegative tensor that is related to the Motzkin polynomial. The relaxation is not tight until the second order, but the rank-one tensor found by the zeroth order is a best nonnegative rank-one approximation (cf. [41, Example 3.3]). The numerical computation is given in Table 8 and is consistent with [41, Example 3.3].

4.3. Best nonnegative rank-one approximation of tensors. In this section, we present some numerical results for some concrete examples from the literature. A larger set of examples were tested. Similar numerical behaviors were observed. The zeroth order DNN relaxation can always return a global optimal solution. Due to space limitation, a small portion of representative examples are shown here.

Example 4.8. This example comes from [12, Example 3]. This is a tensor $\mathcal{A}$ in $\otimes^4\mathbb{R}^2$ with nonzero entries being

$$a_{1111} = 25.1, a_{1212} = 25.6, a_{2121} = 24.8, a_{2222} = 23.$$

This is a nonnegative and nonsymmetric tensor. The relaxation is tight, and the best nonnegative rank-one approximation tensor is the best rank-one approximation tensor (cf. [52]), which is found as

$$\lambda = 25.6000,\ \mathbf{x}_*^1 = \mathbf{x}_*^3 = (1, 0)^\mathsf{T},\ \mathbf{x}_*^2 = \mathbf{x}_*^4 = (0, 1)^\mathsf{T}.$$

The errors $\mathbf{apperr}(\mathcal{A}) = 9.1676 \times 10^{-6}$ and $\mathbf{apperrnm}(\mathcal{A}) = 4.7616 \times 10^{-6}$. The numerical computation is consistent[2] with [41, Example 3.11].

Example 4.9. This example comes from [12, Example 2]. This is a symmetric tensor $\mathcal{A}$ in $\mathrm{Sym}(\otimes^3\mathbb{R}^2)$ with the independent entries being

$$a_{111} = 1.5578, a_{222} = 1.1226, a_{112} = -2.4443, a_{221} = -1.0982.$$

The relaxation is tight. The best nonnegative rank-one approximation tensor found is

$$\lambda = 1.5578,\ \mathbf{x}_* = (1, 0)^\mathsf{T}.$$

The errors $\mathbf{apperr}(\mathcal{A}) = 3.5924 \times 10^{-6}$ and $\mathbf{apperrnm}(\mathcal{A}) = 1.1142 \times 10^{-6}$.

Example 4.10. This example comes from [27, Example 3.6]. It is a symmetric tensor $\mathcal{A}$ in $\mathrm{Sym}(\otimes^3\mathbb{R}^3)$ with the independent entries being

$$a_{111} = -0.1281, a_{112} = 0.0516, a_{113} = -0.0954, a_{122} = -0.1958, a_{123} = -0.1790,$$
$$a_{133} = -0.2676, a_{222} = 0.3251, a_{223} = 0.2513, a_{233} = 0.1773, a_{333} = 0.0338.$$

The relaxation is tight. The best nonnegative rank-one approximation tensor found is

$$\lambda = 0.6187,\ \mathbf{x}_* = (0, 0.8275, 0.5615)^\mathsf{T}.$$

The errors $\mathbf{apperr}(\mathcal{A}) = 2.9194 \times 10^{-6}$ and $\mathbf{apperrnm}(\mathcal{A}) = 2.9194 \times 10^{-6}$.

Example 4.11. This example comes from [41, Example 3.5]. The symmetric tensor $\mathcal{A} \in \mathrm{Sym}(\otimes^m\mathbb{R}^n)$ has the entries

$$a_{i_1 \ldots i_m} = \sum_{j=1}^{m} \frac{(-1)^{i_j}}{i_j}.$$

The numerical computations are recorded in Table 9. We can see that in all cases, the method can find a very good best nonnegative rank-one approximation.

Table 9
Computational results for Example 4.11.

m	n	Time	λ	**apperr**	m	n	Time	λ	**apperr**
3	10	1.4	9.48	4.67×10^{-9}	5	10	5.2	114.86	1.71×10^{-8}
3	20	7.8	19.24	1.96×10^{-5}	5	20	14:15.0	480.17	3.74×10^{-9}
3	30	24.3	28.72	2.58×10^{-4}	6	5	0.27	46.66	1.73×10^{-5}
3	50	6:21.4	47.16	6.00×10^{-8}	6	10	3.7	386.04	2.69×10^{-5}
4	10	0.58	33.49	4.93×10^{-8}	6	20	4:49.3	2319.15	2.50×10^{-7}
4	20	3.3	97.60	1.33×10^{-6}	7	5	1.5	103.01	6.43×10^{-7}
4	30	22.2	179.55	6.59×10^{-9}	7	10	2:23.7	1278.41	2.05×10^{-5}
4	50	11:4.4	382.44	7.43×10^{-8}	8	5	1.7	225.31	2.85×10^{-7}
5	5	0.34	20.82	3.15×10^{-5}	8	10	2:48.2	4186.13	2.06×10^{-4}

[2] We remark on the different errors obtained in our computation and that in [41]. Actually, Nie and Wang reported smaller approximation error. This is due to the facts that (i) the SDP solvers are different (SDPNAL vs. SDPNAL+). Our formulation has an extra nonnegative constraint on the matrix variable. Although the SDPs have the same optimal values, the numerical computations adopt different termination accuracy tolerances. (ii) When we extract the solution for $\mathbf{x}_*$, we also take the absolute values to make sure that $\mathbf{x}_* \geq 0$. This will introduce another difference.

4.4. Copositivitiy of tensors.

In this section, we test some tensors for their copositivities. Let f_{dnn} be the optimal value of the DNN relaxation and f_{app} be the approximation value found as before. Then

1. if $f_{\text{dnn}} \geq 0$, then we can conclude that the tensor is copositive;
2. if $f_{\text{app}} < 0$, then we can conclude that the tensor is not copositive.

Example 4.12. This example comes from [51, page 237]. It is a tensor $\mathcal{A}$ in $\text{Sym}(\otimes^3\mathbb{R}^3)$ with nonzero entries being

$$a_{113} = 2, a_{223} = 2, a_{123} = -1.$$

It can be shown that

$$\langle \mathcal{A}, \mathbf{x}^{\otimes 3} \rangle = 6x_3(x_1^2 + x_2^2 - x_1x_2),$$

and hence $\mathcal{A}$ is copositive. We have that

$$f_{\text{dnn}} = 9.3650 \times 10^{-15} \text{ and } f_{\text{app}} = 2.3094.$$

Therefore we can conclude that the numerical computation gives the correct answer.

Example 4.13. This example comes from [51, Theorem 10]. It is a tensor $\mathcal{A} \in \text{Sym}(\otimes^m\mathbb{R}^n)$ such that for all $i \in \{1, \dots, n\}$

$$a_{ii\dots i} \geq -\sum\{a_{ii_2\dots i_m} : (i, i_2, \dots, i_m) \neq (i, i, \dots, i) \text{ and } a_{ii_2\dots i_m} < 0\}.$$

Tensors satisfying the above assumption are always copositive (cf. [51]). For each case, we first randomly generate a tensor $\mathcal{A} \in \text{Sym}(\otimes^m\mathbb{R}^n)$ and then set for all $i \in \{1, \dots, n\}$

$$a_{ii\dots i} = 10^{-6} - \sum\{a_{ii_2\dots i_m} : (i, i_2, \dots, i_m) \neq (i, i, \dots, i) \text{ and } a_{ii_2\dots i_m} < 0\}.$$

It is simulated rep times for each case, and prob represents the percentage of the instances which are certified as copositive. From the theory, we know that prob should be one. The numerical computations are recorded in Table 10, in which min, mean, and max represent, respectively, the minimum, mean, and maximum values among the simulations.

Example 4.14 (random examples). For $\text{Sym}(\otimes^m\mathbb{R}^n)$, randomly generated instances were tested in this example. Each entry of the tensor is generated randomly uniformly from $[-1, 1]$. The numerical computations are recorded in Table 11.

TABLE 10
Computational results for Example 4.13.

m	n	**rep**	Time (min;mean;max)	f_{dnn} (min;mean;max)	**prob**
3	2	100	0.0650; 0.1218; 0.5040	0.0019; 0.5527; 1.7766	1.0000
3	4	100	0.1170; 0.3354; 4.4760	0.5590; 2.0039; 3.3429	1.0000
4	4	100	0.1000; 0.2808; 1.6110	2.0387; 3.4413; 5.3520	1.0000
4	6	100	0.3090; 0.4830; 1.1150	5.5450; 9.5029; 14.0782	1.0000
4	8	100	0.4920; 0.8504; 1.3890	14.0392; 19.7660; 23.7444	1.0000
4	10	100	0.5260; 1.1967; 1.8570	28.4651; 36.6117; 45.1736	1.0000

Table 11
Computational results for Example 4.14.

m	n	**rep**	Time (min;mean;max)	f_{dnn} (min;mean;max)	**prob**
3	2	100000	0.0170; 0.0836; 1.4910	-0.9901; 0.4744; 2.4283	0.7976
3	3	100	0.0530; 0.2025; 4.9460	-1.9355; 0.7035; 2.5131	0.7100
3	4	100	0.0840; 0.3150; 4.4830	-1.0134; 1.1267; 3.4869	0.8200
4	2	100	0.0280; 0.0883; 0.9500	-0.6436; 0.7512; 2.7042	0.8700
4	3	100	0.0410; 0.1184; 0.7480	-0.8643; 1.1732; 3.8440	0.8900
4	4	100	0.0600; 0.2185; 1.5470	-0.3611; 2.1712; 5.6694	0.9800
5	2	100	0.0500; 0.2081; 4.3240	-0.9681; 0.7340; 3.5141	0.7200
5	3	100	0.0820; 0.7420; 16.2170	-2.8147; 1.5161; 5.6667	0.7600
5	4	100	0.1220; 1.1890; 34.4940	-3.3983; 2.6409; 9.2966	0.7600

5. Conclusions.

This article studied the problem of minimizing a multiform over the nonnegative multisphere. This problem is a special polynomial optimization problem. Although standard SOS relaxation method can be employed to solve this problem, there are computational advantages to consider the more specialized approach in this paper. Taking the biquadratic case for example, i.e., $d_1 = d_2 = 2$, the matrix in the resulting SDP is of dimension $\frac{(n_1+n_2+1)(n_1+n_2)}{2}$. However, the matrix dimension in the DNN relaxation method introduced here is n_1n_2. We can see that the latter method provides a linear matrix conic optimization problem with matrix size that is about half of the former when $n_1 = n_2$, and the ratio is even smaller when $n_1 \ll n_2$ or $n_2 \ll n_1$. Given the current limitations of SDP solvers for handling large scale problems with high-dimensional matrix variables (cf. [17, 59, 62, 66]), the DNN method proposed in this article is promising. Our approach is made practical by the recent solver SDPNAL+ [61, 65], which is designed to efficiently handle large-scale problems with a particular focus on DNN problems having moderate matrix dimensions while allowing the number of linear constraints to be very large.

The method is applied to the problem of finding the best nonnegative rank-one approximation of a given tensor and the problem of testing the copositivity of a given tensor. Based on the promising numerical results, we are motivated to carry out further investigations of the DNN relaxation methods for multiform optimization over the nonnegative multisphere in the future.

Acknowledgments. The major part of this work was carried out during the first author's postdoctoral stint in the Department of Mathematics at the National University of Singapore. We are grateful to the associate editor and anonymous referees for their valuable comments and suggestions that have helped to improve this paper.

REFERENCES

[1] E. Allman, J. Rhodes, B. Sturmfels, and P. Zwiernik, *Tensors of nonnegative rank two*, Linear Algebra Appl., 473 (2015), pp. 37–53.

[2] N. Asgarian and R. Greiner, *Using Rank-*1 *Biclusters to Classify Microarray Data*, Department of Computing Science and the Alberta Ingenuity Center for Machine Learning, University of Alberta, Edmonton, Canada, 2006.

[3] A. Ben-Tal and A. Nemirovskii, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, SIAM, Philadelphia, PA, 2001.

[4] S. Bergmann, J. Ihmels, and N. Barkai, *Iterative signature algorithm for the analysis of large-scale gene expression data*, Phys. Rev. E, 67 (2003), 031902.

[5] M. Biggs, A. Ghodsi, and S. Vavasis, *Nonnegative matrix factorization via rank-one downdating*, in Proceedings of the International Conference on Machine Learning, 2008.

[6] J. Bochnak, M. Coste, and M.-F. Roy, *Real Algebraic Geometry*, Springer, Berlin, 1998.

[7] R. Bro and S. De Jong, *A fast non-negativity-constrained least squares algorithm*, J. Chemometrics, 11 (1997), pp. 393–401.

[8] A. Cichocki, R. Zdunek, A. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Application to Exploratory Multi-Way Data Analysis and Blind Separation*, Wiley, Hoboken, NJ, 2009.

[9] J. Cohen and U. Rothblum, *Nonnegative ranks, decompositions and factorizations of nonnegative matrices*, Linear Algebra Appl., 190 (1993), pp. 149–168.

[10] R. Curto and L. Fialkow, *Solution of the Truncated Complex Moment Problem for Flat Data*, Mem. Amer. Math. Soc. 119, American Mathematical Society, Providence, RI, 1996.

[11] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[12] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-*1 *and rank-*$(r_1, r_2, \ldots, r_n)$ *approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[13] O. Debals, M. Van Barel, and L. De Lathauwer, *Nonnegative matrix factorization using nonnegative polynomial approximations*, IEEE Signal Process. Lett., 24 (2017), pp. 948–952.

[14] P. Dickinson and L. Gijben, *On the computational complexity of membership problems for the completely positive cone and its dual*, Comput. Optim. Appl., 57 (2014), pp. 403–415.

[15] D. FitzGerald, M. Cranitch, and E. Coyle, *Non-negative tensor factorisation for sound source separation*, in Proceedings of the Irish Signals and Systems Conference, 2005.

[16] M. Friedlander and K. Hatz, *Computing nonnegative tensor factorizations*, Comput. Optim. Appl., 23 (2008), pp. 631–647.

[17] K. Fujisawa, Y. Futakata, M. Kojima, S. Matsuyama, S. Nakamura, K. Nakata, and M. Yamashita, *SDPA-M (Semidefinite Programming Algorithm in MATLAB)*, http://homepage.mac.com/klabtitech/sdpa-homepage/download.html.

[18] M. Garey and D. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, Freeman, New York, NY, 1980.

[19] N. Gillis, *Approximation et sous-approximation de matrices par factorisation positive: algorithmes, complexité et applications*, Master's Thesis, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 2006.

[20] G. Hardy, J. Littlewood, and G. Pólya, *Inequalities*, 2nd ed., Cambridge University Press, Cambridge, UK, 1952.

[21] J. Håstad, *Tensor rank is NP-complete*, J. Algorithms, 11 (1990), pp. 644–654.

[22] T. Hazan, S. Polak, and A. Shashua, *Sparse image coding using a* 3*D non-negative tensor factorization*, in Proceedings of the 10th IEEE International Conference on Computer Vision, Vol. 1, IEEE Computer Society Press, 2005, pp. 50–57.

[23] S. Hu and G. Li, *Convergence rate analysis for the higher order power method in best rank one approximations of tensors*, Numer. Math., 140 (2018), pp. 993–1031.

[24] K. Huang, N. Sidiropoulos, and A. Liavas, *A flexible and efficient algorithmic framework for constrained matrix and tensor factorization*, IEEE Trans. Signal Process., 64 (2016), pp. 5052–5065.

[25] 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.

[26] T. Kolda and B. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.

[27] T. Kolda and J. Mayo, *Shifted power method for computing tensor eigenpairs*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1095–1124.

[28] J. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.

[29] C. Lawson and R. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[30] D. Lee and H. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.

[31] B. Levin, *On Calculating Maximum Rank-One under Approximations for Positive Arrays*, Technical report B-48, Division of Biostatistics, Columbia University, New York.

[32] L.-H. Lim, *Tensors and hypermatrices*, in Handbook of Linear Algebra, L. Hogben, ed., CRC Press, Boca Raton, FL 2013, Chapter 15.

[33] L.-H. Lim and P. Comon, *Nonnegative approximations of nonnegative tensors*, J. Chemometrics, 23 (2009), pp. 432–441.

[34] 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.

[35] T. Motzkin and E. Straus, *Maxima for graphs and a new proof of a theorem of Túran*, Canad. J. Math., 17 (1965), pp. 533–540.

[36] K. Murty and S. Kabadi, *Some NP-complete problems in quadratic and nonlinear programming*, Math. Program., 39 (1987), pp. 117–129.

[37] J. Nie, *Sums of squares methods for minimizing polynomial forms over spheres and hypersurfaces*, Front. Math. China, 7 (2012), pp. 321–346.

[38] J. Nie, *Optimality conditions and finite convergence of Lasserre's hierarchy*, Math. Program., 146 (2014), pp. 97–121.

[39] J. Nie, J. Demmel, and B. Sturmfels, *Minimizing polynomials via sums of squares over the gradient ideal*, Math. Program., 106 (2006), pp. 587–606.

[40] J. Nie and M. Schweighofer, *On the complexity of Putinar's Positivstellensatz*, J. Complexity, 23 (2007), pp. 135–150.

[41] J. Nie and L. Wang, *Semidefinite relaxations for best rank-*1 *tensor approximations*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1155–1179.

[42] J. Nie, Z. Yang, and X. Zhang, *A complete semidefinite algorithm for detecting copositive matrices and tensors*, SIAM J. Optim., 28 (2018), pp. 2902–2921.

[43] J. Nie and X. Zhang, *Positive maps and separable matrices*, SIAM J. Optim., 26 (2016), pp. 1236–1256.

[44] P. Paatero, *A weighted non-negative least squares algorithm for three-way "PARAFAC" factor analysis*, Chemometrics. Intell. Lab. Syst., 38 (1997), pp. 223–242.

[45] P. Paatero and U. Tapper, *Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics, 5 (1994), pp. 111–126.

[46] P. Parrilo and B. Sturmfels, *Minimizing polynomial functions*, in Algorithmic and Quantitative Real Algebraic Geometry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 60 American Mathematical Society, Providence, RI 2003, pp. 83–99.

[47] J. Pena, J. Vera, and L. Zuluaga, *Completely positive reformulations for polynomial optimization*, Math. Program., 151 (2014), pp. 405–431.

[48] G. Pólya, *Über positive Darstellung von Polynomen Vierteljschr*, Naturforsch. Ges. Zürich., 73 (1928), pp. 141–145.

[49] V. Powers and B. Reznick, *A new bound for Pólya's theorem with applications to polynomials positive on polyhedra*, J. Pure Appl. Algebra, 164 (2001), pp. 221–229.

[50] M. Putinar, *Positive polynomials on compact semi-algebraic sets*, Indiana Univ. Math. J., 42 (1993), pp. 969–984.

[51] L. Qi, *Symmetric nonnegative tensors and copositive tensors*, Linear Algebra Appl., 439 (2013), pp. 228–238.

[52] Y. Qi, P. Comon, and L.-H. Lim, *Uniqueness of nonnegative tensor approximations*, IEEE Trans. Inform. Theory, 62 (2016), pp. 2170–2183.

[53] B. Reznick, *Some concrete aspects of Hilbert's* 17*th problem*, Contemp. Math., 253 (2000), pp. 251–272.

[54] J.-P. Royer, N. Thirion-Moreau, and P. Comon, *Computing the polyadic decomposition of nonnegative third order tensors*, Signal Process., 91 (2011), pp. 2159–2171.

[55] A. Shashua and T. Hazan, *Non-negative tensor factorization with applications to statistics and computer vision*, in Proceedings of the 22nd International Conference on Machine Learning, 2005, pp. 792–799.

[56] A. Shashua and A. Levin, *Linear image coding for regression and classification using the tensor-rank principle*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2001.

[57] A. Shashua, R. Zass, and T. Hazan, *Multi-way clustering using super-symmetric non-negative tensor factorization*, in Proceedings of the 9th European Conference on Computer Vision, Part IV, 2006, pp. 595–608.

[58] N. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. Papalexakis, and C. Faloutsos, *Tensor decomposition for signal processing and machine learning*, IEEE Trans. Signal Process., 65 (2017), pp. 3551–3582.

[59] J. Sturm, *SeDuMi* 1.02*: A Matlab toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11 & 12 (1999), pp. 625–653.

[60] D. Sun, K.-C. Toh, and L. Yang, *A convergent* 3*-block semi-proximal alternating direction method of multipliers for conic programming with* 4*-type constraints*, SIAM J. Optim., 25 (2015), pp. 882–915.

[61] D. Sun, K.-C. Toh, Y. Yuan, and X.-Y. Zhao, *SDPNAL+: A MATLAB software for semidefinite programming with bound constraints (version* 1.0*)*, Optim. Methods Softw., in press.

[62] K.-C. Toh, M. Todd, and R. Tutuncu, *SDPT3: A Matlab software package for semidefinite programming*, Optim. Methods Softw., 11 (1999), pp. 545–581.

[63] S. Vavasis, *On the complexity of nonnegative matrix factorization*, SIAM J. Optim., 20 (2009), pp. 1364–1377.

[64] M. Welling and M. Weber, *Positive tensor factorization*, Pattern Recogn. Lett., 22 (2001), pp. 1255–1261.

[65] L. Yang, D. Sun, and K.-C. Toh, *SDPNAL+: A majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints*, Math. Program. Comput., 7 (2015), pp. 331–366.

[66] X.-Y. Zhao, D. Sun, and K.-C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optim., 20 (2010), pp. 1737–1765.

[67] G. Zhou, A. Cichocki, Q. Zhao, and S. Xie, *Efficient nonnegative Tucker decompositions: Algorithms and uniqueness*, IEEE Trans. Image Process., 24 (2015), pp. 4990–5003.