

INHOMOGENEOUS POLYNOMIAL OPTIMIZATION OVER A CONVEX SET: AN APPROXIMATION APPROACH

SIMAI HE, ZHENING LI, AND SHUZHONG ZHANG

ABSTRACT. In this paper, we consider computational methods for optimizing a multivariate inhomogeneous polynomial function over a general convex set. The focus is on the design and analysis of polynomial-time approximation algorithms. The methods are able to deal with optimization models with polynomial objective functions in any fixed degrees. In particular, we first study the problem of maximizing an inhomogeneous polynomial over the Euclidean ball. A polynomial-time approximation algorithm is proposed for this problem with an assured (relative) worst-case performance ratio, which is dependent only on the dimensions of the model. The method and approximation ratio are then generalized to optimize an inhomogeneous polynomial over the intersection of a finite number of co-centered ellipsoids. Furthermore, the constraint set is extended to a general convex compact set. Specifically, we propose a polynomial-time approximation algorithm with a (relative) worst-case performance ratio for polynomial optimization over some convex compact sets, e.g., a polytope. Finally, numerical results are reported, revealing good practical performance of the proposed algorithms for solving some randomly generated instances.

1. INTRODUCTION

In this paper, we consider the general constrained polynomial optimization model

$$(G) \quad \begin{array}{ll} \max & p(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in S, \end{array}$$

where $p(\mathbf{x})$ is a multivariate inhomogeneous polynomial function of $\mathbf{x} \in \Re^n$, and $S \subseteq \Re^n$ is a given compact set, typically defined by some polynomial equalities or inequalities. This general optimization problem is one of the fundamental models in the field of optimization. Such problems and formulations are encountered frequently in the literature, with a wide spectrum of applications: quantum mechanics [6, 9], biomedical engineering such as magnetic resonance imaging [3, 7], signal processing [26, 39], numerical linear algebra [32, 38], tensor decomposition [18],

Received by the editor June 29, 2011 and, in revised form, September 11, 2012 and June 13, 2013.

2010 *Mathematics Subject Classification.* Primary 90C26, 90C59, 65Y20, 68W25, 15A69.

Key words and phrases. Polynomial optimization, approximation algorithm, inhomogeneous polynomial, tensor optimization.

The research of the first author was supported in part by Hong Kong GRF #CityU143711.

The research of the second author was supported in part by Natural Science Foundation of China #11371242, Natural Science Foundation of Shanghai #12ZR1410100, and Ph.D. Programs Foundation of Chinese Ministry of Education #20123108120002.

The research of the third author was supported in part by National Science Foundation of USA #CMMI1161242.

investment science [2, 27, 36], among many others. For example, the least square formulation to the sensor network localization problem proposed in Luo and Zhang [25] takes the form of

$$\begin{aligned} \min \quad & \sum_{i,j \in \mathcal{S}} \left(\|\mathbf{x}^i - \mathbf{x}^j\|_2^2 - d_{ij}^2 \right)^2 + \sum_{i \in \mathcal{S}, j \in \mathcal{A}} \left(\|\mathbf{x}^i - \mathbf{a}^j\|_2^2 - d_{ij}^2 \right)^2 \\ \text{s.t.} \quad & \mathbf{x}^i \in G, i \in \mathcal{S}, \end{aligned}$$

where \mathcal{A} and \mathcal{S} denote the set of anchor nodes and sensor nodes respectively, d_{ij} 's are (possibly noisy) distance measurements, \mathbf{a}^j 's denote the known positions of anchor nodes, while \mathbf{x}^i 's represent the positions of sensor nodes in the compact region $G \subseteq \mathbb{R}^3$ to be estimated. Another example is the portfolio management problem involving more than the first two moments (e.g. the skewness and the kurtosis of the investment returns). That problem has been receiving much attention in the literature (cf., De Athayde and Flôre [2], Prakash et al. [36], Kleniati et al. [16]). In particular, a very general model in [16] is

$$\begin{aligned} \max \quad & \alpha \sum_{i=1}^n \mu_i x_i - \beta \sum_{i,j=1}^n \sigma_{ij} x_i x_j + \gamma \sum_{i,j,k=1}^n \varsigma_{ijk} x_i x_j x_k - \delta \sum_{i,j,k,\ell=1}^n \kappa_{ijk\ell} x_i x_j x_k x_\ell \\ \text{s.t.} \quad & \mathbf{e}^T \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \in \mathbb{R}^n, \end{aligned}$$

where $(\mu_i), (\sigma_{ij}), (\varsigma_{ijk}), (\kappa_{ijk\ell})$ are the first four moments of the given n assets, and \mathbf{e} is the all-one vector. The nonnegative parameters $\alpha, \beta, \gamma, \delta$ measure the investor's preference to the four moments, and they sum up to one, i.e., $\alpha + \beta + \gamma + \delta = 1$.

The polynomial optimization problems are typically nonconvex, highly nonlinear, and NP-hard in general. The search for general and efficient algorithms for polynomial optimization has been a priority for many mathematical optimizers. For example, generic solution methods based on nonlinear programming and global optimization have been studied and tested; see e.g. Qi [37] and Qi et al. [40]. Tensor relaxation and local improvement methods have been discussed by Chen et al. [5]. In recent years, an entirely different and systematic approach based on the *sum of squares* (SOS) was proposed by Lasserre [19, 20] and Parrilo [34, 35]. The SOS approach has a strong theoretical appeal, since it can in principle solve any general polynomial optimization model to any precision, by resorting to (possibly large) semidefinite programs (SDP). However, the SDP problems required to be solved by the SOS approach grow exponentially, making the SOS approach only viable for low dimensional problems. Henrion et al. [13] developed a specialized tool known as GloptiPoly 3 for finding a global optimal solution for polynomial optimization based on the SOS approach. For an overview on the recent theoretical developments, we refer to the excellent survey by Laurent [21].

On the other side, the intractability of general polynomial optimizations therefore motivates the search for suboptimal, or more formally, approximate solutions. In the case that the objective polynomial is quadratic, a well-known example is the SDP relaxation and randomization approach for the max-cut problem due to Goemans and Williamson [8], where essentially a 0.878-approximation ratio of the model $\max_{\mathbf{x} \in \{1, -1\}^n} \mathbf{x}^T \mathbf{F} \mathbf{x}$ is shown with \mathbf{F} being the Laplacian of a given graph. Note that the approach in [8] has been generalized subsequently by many authors, including Nesterov [30], Ye [44, 45], Nemirovski et al. [29], Zhang [47], Alon and Naor [1], Zhang and Huang [48], Luo et al. [24], and He et al. [12]. In particular, when the matrix \mathbf{F} is only known to be positive semidefinite, Nesterov [30] derived

a 0.636-approximation bound for $\max_{\mathbf{x} \in \{1, -1\}^n} \mathbf{x}^T \mathbf{F} \mathbf{x}$. Nemirovski et al. [29] proposed an $\Omega(1/\log m)$ -approximation bound for maximizing a quadratic form over the intersection of m co-centered ellipsoids. Their models are further studied and generalized by Luo et al. [24] and He et al. [12].

Beyond quadratic objectives, the approximation methods for high degree polynomial optimizations have attracted much attention recently. Luo and Zhang [25] considered quartic optimization, and showed that optimizing a homogenous quartic form over the intersection of some co-centered ellipsoids is essentially equivalent to its (quadratic) SDP relaxation problem, which is itself also NP-hard. However, this gives a handle on the design of approximation algorithms with provable worst-case approximation ratios. Ling et al. [23] considered a special quartic optimization model. Basically, the problem is to minimize a biquadratic function over two spherical constraints. In [23], approximate solutions as well as exact solutions using the SOS method are considered. The approximation bounds in [23] are indeed comparable to the bound in [25]. A breakthrough in approximation methods for any degree polynomial objective was due to He et al. [10], where the authors proposed tensor relaxation methods and derived polynomial-time approximation algorithms for any fixed degree homogeneous polynomial with quadratic constraints, including the spherical constraints and the intersection of a finite number of co-centered ellipsoids, and the approximation ratios beats that in [25] and [23] specialized to degree 4. The result of [10] is then extended to homogeneous polynomial optimization over discrete, typically binary variables, as well as binary variables mixed with spherical constraints; see [11]. In the meantime, Zhang et al. [49] studied cubic form optimization with spherical constraints, and proposed polynomial-time approximation algorithms, and the ratio is comparable to that in [10], specialized to degree 3. Very recently, So [41] improved the approximation ratios for any fixed degree spherical constrained homogeneous polynomial optimization problems studied in [10].

All the above successful approximation methods only tackle optimization of *homogeneous* polynomials, while the study of approximation algorithms for *inhomogeneous* polynomial optimization is scarcely found in the literature, despite the fact that inhomogeneous polynomial optimization is commonplace as we discussed earlier. Nemirovski et al. [29] discussed inhomogeneous quadratic function maximization over homogeneous quadric constraints, and proposed polynomial-time randomized approximation algorithms. However, their model and its analysis is essentially homogeneous quadratic polynomial optimization. On the other front, Nie [33] discussed approximation bounds on the gap between the optimal value of a general polynomial optimization problem and that of Lasserre's relaxation. However, no feasible solution can be generated for the original polynomial optimization problem. In fact, the bounds in [33] are complementary to the approximation bounds studied in this paper. Meanwhile, De Klerk and Laurent [14] obtained some error bounds for SDP approaches to inhomogeneous polynomial minimization on the hypercube. In this paper, we propose polynomial-time approximation algorithms for optimizing an inhomogeneous polynomial over a general compact set. In fact, extending the solution methods and the corresponding analysis from *homogeneous* polynomial optimization to the general *inhomogeneous* polynomials is not straightforward. Technically, a homogenous polynomial function allows one to *scale* the overall function value along a given direction, which is an essential

operation in proving the quality bound of the approximation algorithms. The current paper breaks its path from the preceding practices, by directly dealing with a *homogenizing* variable. Although homogenization is a natural way to deal with inhomogeneous polynomials, it is quite a different matter when it comes to the worst-case performance ratio analysis. In fact, the usual homogenization does not lead to any assured performance ratio. Here we point out a specific route via homogenization and tensor relaxation to get around this difficulty, and we actually provide a general scheme to approximately solve such problems.

The paper is organized as follows. First in Section 2, we introduce the notations and the models. Then in Section 3, we analyze the model where the constraint set is the Euclidean ball and propose polynomial-time approximation algorithms with guaranteed performance ratios, which serves as a basis for the subsequent analysis. In Section 4, the discussion is extended to cover the problem where the constraint set is the intersection of a finite number of co-centered ellipsoids. In Section 5, the approximation bounds will be derived even under some very general compact sets, e.g., a polytope. It turns out that for such general problems, it is still possible to derive relative approximation ratios, which is still dependent on the problem dimensions only. Finally, we report our numerical experiment results in Section 6.

2. NOTATIONS, MODELS, AND THE ORGANIZATION

To make a clear visual distinction, in this paper we use the boldface letters to denote vectors, matrices, and tensors in general (e.g., the decision variable \mathbf{x} , the data matrix \mathbf{Q} , and the tensor form \mathbf{F}), while the usual lowercase letters are reserved for scalars (e.g. the homogenizing variable x_h to be introduced later). The Greek letters ξ , η and β are reserved to denote binary variables (e.g. $\xi \in \mathbb{B}^d := \{1, -1\}^d$). The letter τ , however, is reserved to indicate the *approximation ratio*, which is a key ingredient throughout this paper.

Let us start by defining the following multilinear function

$$F(\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^d) = \sum_{1 \leq i_1 \leq n_1, 1 \leq i_2 \leq n_2, \dots, 1 \leq i_d \leq n_d} F_{i_1 i_2 \dots i_d} x_{i_1}^1 x_{i_2}^2 \dots x_{i_d}^d,$$

where $\mathbf{x}^k \in \mathbb{R}^{n_k}$ for $k = 1, 2, \dots, d$, and $\mathbf{F} = (F_{i_1 i_2 \dots i_d}) \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ is a d -th order tensor with F being its associated multilinear function.

Closely related to the tensor \mathbf{F} is a general d -th degree homogeneous polynomial function $f(\mathbf{x})$ of $\mathbf{x} \in \mathbb{R}^n$. We call the tensor *super-symmetric* (see [17]) if $F_{i_1 i_2 \dots i_d}$ is invariant under all permutations of $\{i_1, i_2, \dots, i_d\}$. As any quadratic function uniquely determines a symmetric matrix, a given homogeneous polynomial function $f(\mathbf{x})$ also uniquely determines a super-symmetric tensor. For a given super-symmetric tensor form \mathbf{F} , we denote by F the multilinear function induced by it, and by f the homogeneous polynomial function, i.e., $f(\mathbf{x}) = F(\underbrace{\mathbf{x}, \mathbf{x}, \dots, \mathbf{x}}_d)$.

The Frobenius norm of the tensor form \mathbf{F} is the usual Euclidean norm (or the 2-norm) defined as

$$\|\mathbf{F}\| := \sqrt{\sum_{1 \leq i_1 \leq n_1, 1 \leq i_2 \leq n_2, \dots, 1 \leq i_d \leq n_d} F_{i_1 i_2 \dots i_d}^2}.$$

In this paper we exclusively use only the 2-norm for the vectors, the matrices, and the tensors.

A generic d -th degree multivariate (inhomogeneous) polynomial function $p(\mathbf{x})$, can be explicitly written as a summation of homogenous polynomial functions in decreasing degrees, namely

$$p(\mathbf{x}) := \sum_{i=1}^d f_i(\mathbf{x}) + f_0 = \sum_{i=1}^d F_i(\overbrace{\mathbf{x}, \mathbf{x}, \dots, \mathbf{x}}^i) + f_0,$$

where $\mathbf{x} \in \mathbb{R}^n$, $f_0 \in \mathbb{R}$ and $f_i(\mathbf{x}) = F_i(\overbrace{\mathbf{x}, \mathbf{x}, \dots, \mathbf{x}}^i)$ is a homogenous polynomial function of degree i for $i = 1, 2, \dots, d$.

One natural way to deal with inhomogeneous polynomial functions is through *homogenization*; that is, we introduce a new variable, to be denoted by x_h in this paper, which is actually set to be 1, to yield a homogeneous form

$$(2.1) \quad p(\mathbf{x}) = \sum_{i=1}^d f_i(\mathbf{x}) + f_0 = \sum_{i=1}^d f_i(\mathbf{x}) x_h^{d-i} + f_0 x_h^d = f(\bar{\mathbf{x}}),$$

where $f(\bar{\mathbf{x}})$ is an $(n+1)$ -dimensional homogeneous polynomial function of degree d , with variable $\bar{\mathbf{x}} \in \mathbb{R}^{n+1}$. Throughout this paper, the ‘bar’ notation, e.g., $\bar{\mathbf{x}}$, is reserved for an $(n+1)$ -dimensional vector, with the underlying letter \mathbf{x} referring to the vector of its first n components, and the subscript ‘ h ’ (the subscript of x_h) referring to its last component. For instance, if $\bar{\mathbf{x}} = (x_1, x_2, \dots, x_n, x_{n+1})^T \in \mathbb{R}^{n+1}$, then $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ and $x_h = x_{n+1} \in \mathbb{R}$.

In this paper we study optimization of a generic inhomogeneous polynomial $p(\mathbf{x})$, under three types of constraint sets: **(B)** The Euclidean ball: $\{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq 1\}$; **(Q)** The intersection of co-centered ellipsoids: $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^T \mathbf{Q}_i \mathbf{x} \leq 1, i = 1, 2, \dots, m\}$; **(G)** A general compact set $S \subseteq \mathbb{R}^n$. These three models and results are discussed in Sections 3, 4 and 5, respectively. Since all these models are NP-hard in general, we shall propose polynomial-time approximation algorithms and derive worst-case performance ratios.

Throughout this paper, for any maximization model (P) defined as $\max_{\mathbf{x} \in S} f(\mathbf{x})$, we use $v_{\max}(P)$ to denote its optimal value, and $v_{\min}(P)$ to denote the optimal value of its minimization counterpart, i.e., $v_{\max}(P) := \max_{\mathbf{x} \in S} f(\mathbf{x})$ and $v_{\min}(P) := \min_{\mathbf{x} \in S} f(\mathbf{x})$.

Definition 2.1. A maximization model (P) admits a polynomial-time approximation algorithm with approximation ratio $\tau \in (0, 1]$, if $v_{\max}(P) \geq 0$ and a feasible solution $\mathbf{y} \in S$ can be found in polynomial-time such that $f(\mathbf{y}) \geq \tau v_{\max}(P)$.

Definition 2.2. A maximization model (P) admits a polynomial-time approximation algorithm with *relative* approximation ratio $\tau \in (0, 1]$, if a feasible solution $\mathbf{y} \in S$ can be found in polynomial-time such that $f(\mathbf{y}) - v_{\min}(P) \geq \tau (v_{\max}(P) - v_{\min}(P))$.

Regarding the relative approximation ratio in Definition 2.2, in some cases it is convenient to use the equivalent form: $v_{\max}(P) - f(\mathbf{y}) \leq (1 - \tau) (v_{\max}(P) - v_{\min}(P))$. Since this paper is mostly concerned with the relative approximation ratios, we may, without loss of generality, assume that the inhomogeneous polynomial function $p(\mathbf{x})$ has no constant term, i.e., $f_0 = 0$ in (2.1). The degree of the objective polynomial function $p(\mathbf{x})$, d , is deemed a *fixed* parameter in our subsequent discussions.

3. POLYNOMIAL OPTIMIZATION WITH THE EUCLIDEAN BALL CONSTRAINT

Our first model under consideration is to maximize a generic multivariate polynomial function subject to the Euclidean ball constraint, i.e.,

$$(B) \quad \begin{array}{ll} \max & p(\mathbf{x}) \\ \text{s.t.} & \|\mathbf{x}\| \leq 1, \mathbf{x} \in \mathbb{R}^n. \end{array}$$

Since we assume $p(\mathbf{x})$ to have no constant term, the optimal value of this problem is obviously nonnegative, i.e., $v_{\max}(B) \geq 0$.

The complexity to solve Problem (B) can be summarized by the following proposition.

Proposition 3.1. *If $d \leq 2$, then Problem (B) can be solved in polynomial-time; if $d \geq 3$, then Problem (B) is NP-hard; if $d \geq 4$, then there is no polynomial-time approximation algorithm with a positive approximation ratio unless $P = NP$.*

Proof. For $d \leq 2$, Problem (B) is a standard trust region subproblem. As such it is well known to be solvable in polynomial-time (cf. [42, 43, 46] and the references therein). For $d = 3$, in a special case when $p(\mathbf{x})$ is a homogeneous polynomial function, it is easy to see that Problem (B) is equivalent to $\max_{\|\mathbf{x}\|=1} p(\mathbf{x})$, which is shown to be NP-hard by Nesterov [31].

Consider now $d \geq 4$. As a subclass of Problem (B), we may let $p(\mathbf{x})$ be a fourth order homogeneous polynomial function, i.e., $p(\mathbf{x}) = F(\mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x})$, where \mathbf{F} is a super-symmetric fourth order tensor. We call $F(\mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x})$ to be positive semidefinite if $F(\mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$. It is well known that checking the positive semidefiniteness of $F(\mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x})$ is co-NP-complete (cf. [25]). If we are able to find a polynomial-time approximation algorithm to get a positive approximation ratio τ for $-F(\mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x})$, then this algorithm can be used to check the positive semidefiniteness of tensor \mathbf{F} . To see why, suppose this algorithm returns a feasible solution \mathbf{y} with $-F(\mathbf{y}, \mathbf{y}, \mathbf{y}, \mathbf{y}) > 0$, then $F(\mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x})$ is not positive semidefinite. Otherwise the algorithm must return a feasible solution \mathbf{y} with $0 \geq -F(\mathbf{y}, \mathbf{y}, \mathbf{y}, \mathbf{y}) \geq \tau v_{\max}(B)$, which implies $v_{\max}(B) = 0$; hence, $F(\mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x})$ is positive semidefinite in this case. Therefore, such algorithms cannot exist unless $P = NP$. \square

We remark that, even when $d = 3$, there is no polynomial-time approximation algorithm with a positive approximation ratio unless $P = NP$ (see Proposition 5.2.1 in Ph.D. thesis of Li [22]). This strengthens the result in Proposition 3.1. These facts rule out any polynomial-time approximation algorithms with a positive approximation ratio for Problem (B). However, a positive *relative* approximation ratio is still possible, which is the main target of this section. Below we shall first present a polynomial-time algorithm for approximately solving Problem (B), which admits a (relative) worst-case performance ratio. In fact, we shall present a general scheme aimed at solving the inhomogeneous polynomial optimization Problem (B). This scheme breaks down to the following four major steps:

Step 1: Introduce a proper (equivalent) model with a homogenous objective.

Step 2: Solve a tensor relaxation model with the objective being a multilinear function.

Step 3: Adjust to get a solution based on the solution of the relaxed model.

Step 4: Assemble a solution for the original inhomogeneous model.

Some of these steps can be designed separately. The algorithm below can be considered as one realization of the general scheme for solving Problem (B), with

each step being carried out by a specific procedure. We shall first present the specialized algorithm, and then in the remainder of the section, we shall elaborate on these four general steps, and prove that in combination they lead to a polynomial-time approximation algorithm with a quality-assured solution.

Algorithm 3.2.

Input: an n -dimensional d -th degree polynomial function $p(\mathbf{x})$ with $p(\mathbf{0}) = 0$.

- (1) Rewrite $p(\mathbf{x}) = F(\underbrace{\bar{\mathbf{x}}, \bar{\mathbf{x}}, \dots, \bar{\mathbf{x}}}_d)$ when $x_h = 1$ as in (3.2), where \mathbf{F} is an $(n+1)$ -dimensional d -th order super-symmetric tensor.
- (2) Approximately solve the problem

$$\begin{aligned} \max \quad & F(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \dots, \bar{\mathbf{x}}^d) \\ \text{s.t.} \quad & \|\bar{\mathbf{x}}^k\| = 1, k = 1, 2, \dots, d \end{aligned}$$

using Algorithm 1 of [10], and get a feasible solution $(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d)$.

- (3) Compute $(\bar{\mathbf{z}}^1, \bar{\mathbf{z}}^2, \dots, \bar{\mathbf{z}}^d) = \arg \max \left\{ F\left(\left(\beta_1 \mathbf{y}_1^{1/d}\right), \left(\beta_2 \mathbf{y}_2^{2/d}\right), \dots, \left(\beta_d \mathbf{y}_d^{d/d}\right)\right), \beta \in \mathbb{B}^d \right\}$.
- (4) Compute $\mathbf{z} = \arg \max \left\{ p(\mathbf{0}); p\left(\frac{\mathbf{z}(\beta)}{z_h(\beta)}\right), \beta \in \mathbb{B}^d \text{ and } \beta_1 = \prod_{k=2}^d \beta_k = 1 \right\}$, where

$$\bar{\mathbf{z}}(\beta) = \beta_1(d+1)\bar{\mathbf{z}}^1 + \sum_{k=2}^d \beta_k \bar{\mathbf{z}}^k.$$

Output: a feasible solution \mathbf{z} .

Recall that in Section 2, for any given vector $\bar{\mathbf{a}} \in \mathbb{R}^{n+1}$, we denote $\mathbf{a} \in \mathbb{R}^n$ to be its first n components, and $a_h \in \mathbb{R}$ to be its last component. In Step 2 of Algorithm 3.2, an algorithm in [10] is called to approximately solve spherically constrained multilinear function optimization, which is actually a deterministic polynomial-time algorithm. In our analysis, the degree of $p(\mathbf{x})$ is deemed a fixed parameter, and thus Algorithm 3.2 runs in polynomial-time, and is deterministic too. A straightforward computation shows that the time complexity of Algorithm 3.2 is $O(n^d)$, which is the same order in evaluating the function $p(\mathbf{x})$. Our main result in this section is the following:

Theorem 3.3. *Problem (B) admits a polynomial-time approximation algorithm (Algorithm 3.2) with relative approximation ratio τ_B , i.e., a feasible solution \mathbf{z} can be found in polynomial-time, such that*

$$p(\mathbf{z}) - v_{\min}(B) \geq \tau_B (v_{\max}(B) - v_{\min}(B)),$$

$$\text{where } \tau_B := 2^{-\frac{5d}{2}}(d+1)!d^{-2d}(n+1)^{-\frac{d-2}{2}} = \Omega\left(n^{-\frac{d-2}{2}}\right).$$

Although homogenization is a natural way to deal with inhomogeneous polynomial functions, the worst-case performance ratio does not follow straightforwardly. What is lacking is that an inhomogeneous polynomial function does not allow one to scale the overall function value along a given direction, which is, however, an essential operation to prove the quality bound of the approximation algorithms (cf. [10, 12, 24, 25, 29]). Below we shall study in detail how a particular implementation of these four steps of the scheme (which becomes Algorithm 3.2) leads to

the promised worst-case relative performance ratio. As we shall see later, our solution scheme can be applied to solve a very general polynomial optimization model (Section 5).

3.1. Homogenization. The method of homogenization depends on the form of the polynomial $p(\mathbf{x})$. If $p(\mathbf{x})$ is given as a summation of homogeneous polynomial functions of different degrees, i.e., $f_i(\mathbf{x})$ ($1 \leq i \leq d$) is a homogeneous polynomial function of degree i , then we may first write

$$(3.1) \quad f_i(\mathbf{x}) = F_i(\underbrace{\mathbf{x}, \mathbf{x}, \dots, \mathbf{x}}_i)$$

with \mathbf{F}_i being an i -th order super-symmetric tensor. Then by introducing a homogenizing variable x_h , which is always equal to 1, we may rewrite $p(\mathbf{x})$ as

$$(3.2) \quad \begin{aligned} p(\mathbf{x}) &= \sum_{i=1}^d F_i(\underbrace{\mathbf{x}, \mathbf{x}, \dots, \mathbf{x}}_i) x_h^{d-i} = F\left(\underbrace{\begin{pmatrix} \mathbf{x} \\ x_h \end{pmatrix}, \begin{pmatrix} \mathbf{x} \\ x_h \end{pmatrix}, \dots, \begin{pmatrix} \mathbf{x} \\ x_h \end{pmatrix}}_d\right) \\ &= F(\underbrace{\bar{\mathbf{x}}, \bar{\mathbf{x}}, \dots, \bar{\mathbf{x}}}_d) = f(\bar{\mathbf{x}}), \end{aligned}$$

where \mathbf{F} is an $(n+1)$ -dimensional d -th order super-symmetric tensor, whose last component is 0 (since $p(\mathbf{x})$ has no constant term).

If the polynomial $p(\mathbf{x})$ is given in terms of summation of monomials, we should first group them according to their degrees, and then rewrite the summation of monomials in each group as homogeneous polynomial function. After that, we then proceed according to (3.1) and (3.2) to obtain the tensor form \mathbf{F} , as required.

Finally in this step, we may equivalently reformulate Problem (B) as

$$\begin{aligned} (\bar{B}) \quad & \max \quad f(\bar{\mathbf{x}}) \\ \text{s.t.} \quad & \bar{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ x_h \end{pmatrix}, \\ & \|\mathbf{x}\| \leq 1, \mathbf{x} \in \Re^n, \\ & x_h = 1. \end{aligned}$$

Obviously, we have $v_{\max}(B) = v_{\max}(\bar{B})$ and $v_{\min}(B) = v_{\min}(\bar{B})$.

3.2. Tensor relaxation. Tensor relaxation (multilinear relaxation) refers to the approach that one relaxes the homogeneous polynomial function to the (separate) multilinear tensor form. This relaxation proves to be effective (see e.g. [10, 15]). Now we relax Problem (\bar{B}) to an inhomogeneous multilinear tensor form optimization problem as:

$$\begin{aligned} (T_B) \quad & \max \quad F(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \dots, \bar{\mathbf{x}}^d) \\ \text{s.t.} \quad & \bar{\mathbf{x}}^k = \begin{pmatrix} \mathbf{x}^k \\ x_h^k \end{pmatrix}, k = 1, 2, \dots, d, \\ & \|\mathbf{x}^k\| \leq 1, \mathbf{x}^k \in \Re^n, k = 1, 2, \dots, d, \\ & x_h^k = 1, k = 1, 2, \dots, d. \end{aligned}$$

Obviously, we have $v_{\max}(T_B) \geq v_{\max}(\bar{B}) = v_{\max}(B)$. Before proceeding, let us first settle the computational complexity issue for solving Problem (T_B).

Proposition 3.4. *Problem (T_B) is NP-hard whenever $d \geq 3$.*

Proof. Notice that the problem

$$\begin{aligned} \max \quad & F(\mathbf{x}, \mathbf{y}, \mathbf{z}) \\ \text{s.t.} \quad & \|\mathbf{x}\| = \|\mathbf{y}\| = \|\mathbf{z}\| = 1, \\ & \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^n \end{aligned}$$

is proven to be NP-hard by He et al. [10] and Zhang et al. [49]. When $d = 3$ and a special case where $\mathbf{F} = (F_{i,j,k})$ has the form $F_{n+1,j,k} = F_{i,n+1,k} = F_{i,j,n+1} = 0$ for all $1 \leq i, j, k \leq n+1$, Problem (T_B) is equivalent to the above model, and thus is NP-hard. \square

Problem (T_B) is still difficult to solve, and moreover it remains inhomogeneous, since x_h^k is required to be 1. To our best knowledge, no polynomial-time approximation algorithm is available in the literature to solve this problem. Furthermore, we shall relax the constraint $x_h^k = 1$, and introduce the following parameterized and homogenized problem:

$$\begin{aligned} (T_B(t)) \quad \max \quad & F(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \dots, \bar{\mathbf{x}}^d) \\ \text{s.t.} \quad & \|\bar{\mathbf{x}}^k\| \leq t, \bar{\mathbf{x}}^k \in \mathbb{R}^{n+1}, k = 1, 2, \dots, d. \end{aligned}$$

Obviously, Problem (T_B) can be relaxed to Problem $(T_B(\sqrt{2}))$, since if $\bar{\mathbf{x}}$ is feasible for Problem (T_B) then $\|\bar{\mathbf{x}}\|^2 = \|\mathbf{x}\|^2 + x_h^2 \leq 1 + 1 = 2$. Consequently, $v_{\max}(T_B(\sqrt{2})) \geq v_{\max}(T_B)$.

Both the objective and the constraints are now homogeneous, and it is easy to see for all $t > 0$, Problem $(T_B(t))$ is equivalent to each other by a simple scaling method. Moreover, Problem $(T_B(1))$ is equivalent to

$$\begin{aligned} \max \quad & F(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \dots, \bar{\mathbf{x}}^d) \\ \text{s.t.} \quad & \|\bar{\mathbf{x}}^k\| = 1, \bar{\mathbf{x}}^k \in \mathbb{R}^{n+1}, k = 1, 2, \dots, d, \end{aligned}$$

which was studied in [10, 41], with available approximation algorithms to cope with. Here we quote one corresponding result as follows:

Theorem 3.5 (He, Li, and Zhang [10]). *Suppose F is a d -th order multilinear form. The problem*

$$\begin{aligned} \max \quad & F(\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^d) \\ \text{s.t.} \quad & \|\mathbf{x}^k\| = 1, \mathbf{x}^k \in \mathbb{R}^{n_k}, k = 1, 2, \dots, d \end{aligned}$$

admits a polynomial-time approximation algorithm with approximation ratio $\frac{1}{\sqrt{n_1 n_2 \dots n_{d-2}}}$.

The intrinsic algorithm of Theorem 3.5 is described in Algorithm 1 of [10]. By applying this result, Problem $(T_B(1))$ admits a polynomial-time approximation algorithm with approximation ratio $(n+1)^{-\frac{d-2}{2}}$. Therefore, for all $t > 0$, Problem $(T_B(t))$ also admits a polynomial-time approximation algorithm with approximation ratio $(n+1)^{-\frac{d-2}{2}}$, and $v_{\max}(T_B(t)) = t^d v_{\max}(T_B(1))$. After this relaxation step (Step 2 in Algorithm 3.2), we are able to find a feasible solution $(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d)$ of Problem $(T_B(1))$ in polynomial-time, such that

$$\begin{aligned} (3.3) \quad F(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d) &\geq (n+1)^{-\frac{d-2}{2}} v_{\max}(T_B(1)) \\ &= 2^{-\frac{d}{2}} (n+1)^{-\frac{d-2}{2}} v_{\max}(T_B(\sqrt{2})) \\ &\geq 2^{-\frac{d}{2}} (n+1)^{-\frac{d-2}{2}} v_{\max}(T_B). \end{aligned}$$

Theorem 3.5 (or specifically Algorithm 1 of [10]) is the engine enabling *Step 2* of our scheme. Problem $(T_B(1))$ itself is an independent and very interesting topic; for details, one is referred to [10, 41]. In fact, any polynomial-time approximation algorithm of Problem $(T_B(1))$ can be used as an engine to yield a realization (algorithm) of our scheme. As will become evident later, any improvement of the approximation ratio of Problem $(T_B(1))$ leads to the improvement of relative approximation ratio in Theorem 3.3. For example, recently So [41] improved the approximation bound of Problem $(T_B(1))$ to $\Omega\left(\left(\frac{\log n}{n}\right)^{-\frac{d-2}{2}}\right)$, albeit the algorithm is mainly of a theoretical interest. Consequently, the relative approximation ratio under our scheme is improved to $\Omega\left(\left(\frac{\log n}{n}\right)^{-\frac{d-2}{2}}\right)$ too. Of course, one may apply any other favorite algorithm to solve the relaxation Problem $(T_B(1))$. For instance, the *alternating least square* method (see [18]) and *maximum block improvement* method (see [5]) can be other alternatives for Step 2.

3.3. Adjustment of the homogenizing component. The approximate solution $(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d)$ of Problem $(T_B(1))$ satisfies $\|\bar{\mathbf{y}}^k\| \leq 1$ for all $1 \leq k \leq d$, which implies $\|\mathbf{y}^k\| \leq 1$, but in general we do not have any control on the size of y_h^k , and thus $(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d)$ may not be a feasible solution for Problem (T_B) . The following lemma plays the role of a bridge in our analysis, to ensure that the construction of a feasible solution to the inhomogeneous Problem (T_B) is possible.

Lemma 3.6. *Suppose $\bar{\mathbf{x}}^k \in \mathbb{R}^{n+1}$ with $|x_h^k| \leq 1$ for all $1 \leq k \leq d$. Let $\eta_1, \eta_2, \dots, \eta_d$ be independent random variables, each takes values 1 and -1 with $\mathbb{E}[\eta_k] = x_h^k$ for $k = 1, 2, \dots, d$, and let $\xi_1, \xi_2, \dots, \xi_d$ be the i.i.d. random variables, taking values 1 and -1 with equal probability (thus the mean is 0). If the last component of the tensor \mathbf{F} is 0, then we have*

$$(3.4) \quad \mathbb{E} \left[\prod_{k=1}^d \eta_k F \left(\begin{pmatrix} \eta_1 \mathbf{x}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \eta_2 \mathbf{x}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \eta_d \mathbf{x}^d \\ 1 \end{pmatrix} \right) \right] = F(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \dots, \bar{\mathbf{x}}^d)$$

and

$$(3.5) \quad \mathbb{E} \left[F \left(\begin{pmatrix} \xi_1 \mathbf{x}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \xi_2 \mathbf{x}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \xi_d \mathbf{x}^d \\ 1 \end{pmatrix} \right) \right] = 0.$$

Proof. The claimed equations readily result from the following observations:

$$\begin{aligned} & \mathbb{E} \left[\prod_{k=1}^d \eta_k F \left(\begin{pmatrix} \eta_1 \mathbf{x}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \eta_2 \mathbf{x}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \eta_d \mathbf{x}^d \\ 1 \end{pmatrix} \right) \right] \\ &= \mathbb{E} \left[F \left(\begin{pmatrix} \eta_1^2 \mathbf{x}^1 \\ \eta_1 \end{pmatrix}, \begin{pmatrix} \eta_2^2 \mathbf{x}^2 \\ \eta_2 \end{pmatrix}, \dots, \begin{pmatrix} \eta_d^2 \mathbf{x}^d \\ \eta_d \end{pmatrix} \right) \right] \quad (\text{multilinearity of } F) \\ &= F \left(\mathbb{E} \left[\begin{pmatrix} \mathbf{x}^1 \\ \eta_1 \end{pmatrix} \right], \mathbb{E} \left[\begin{pmatrix} \mathbf{x}^2 \\ \eta_2 \end{pmatrix} \right], \dots, \mathbb{E} \left[\begin{pmatrix} \mathbf{x}^d \\ \eta_d \end{pmatrix} \right] \right) \quad (\text{independence of } \eta_k \text{'s}) \\ &= F(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \dots, \bar{\mathbf{x}}^d) \end{aligned}$$

and

$$\begin{aligned}
& \mathbb{E} \left[F \left(\begin{pmatrix} \xi_1 \mathbf{x}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \xi_2 \mathbf{x}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \xi_d \mathbf{x}^d \\ 1 \end{pmatrix} \right) \right] \\
&= F \left(\mathbb{E} \left[\begin{pmatrix} \xi_1 \mathbf{x}^1 \\ 1 \end{pmatrix} \right], \mathbb{E} \left[\begin{pmatrix} \xi_2 \mathbf{x}^2 \\ 1 \end{pmatrix} \right], \dots, \mathbb{E} \left[\begin{pmatrix} \xi_d \mathbf{x}^d \\ 1 \end{pmatrix} \right] \right) \quad (\text{independence of } \xi_k \text{'s}) \\
&= F \left(\begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \right) \quad (\text{zero-mean of } \xi_k \text{'s}) \\
&= 0,
\end{aligned}$$

where the last equality is due to the fact that the last component of \mathbf{F} is 0. \square

Lemma 3.6 suggests that one may enumerate the 2^d possible combinations of $(\xi_1 \mathbf{y}^1), (\xi_2 \mathbf{y}^2), \dots, (\xi_d \mathbf{y}^d)$ and pick the one with the largest value of function F (or via a simple randomization procedure), to generate a feasible solution for the inhomogeneous multilinear tensor form optimization Problem (T_B) from a feasible solution of the homogeneous multilinear tensor form optimization Problem $(T_B(1))$, with a controlled quality deterioration. It plays a key role in proving the approximation ratio for Problem (T_B) , which is a by-product in this section.

Theorem 3.7. *Problem (T_B) admits a polynomial-time approximation algorithm with approximation ratio $\tau_1 := 2^{-\frac{3d}{2}}(n+1)^{-\frac{d-2}{2}}$.*

Proof. Let $(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d)$ be the feasible solution found in Step 2 of Algorithm 3.2 satisfying (3.3), and let $\boldsymbol{\eta} = (\eta_1, \eta_2, \dots, \eta_d)^T$ with all η_k 's being independent and taking values 1 and -1 such that $\mathbb{E}[\eta_k] = y_h^k$. By applying Lemma 3.6, (3.4) explicitly implies

$$\begin{aligned}
& F(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d) \\
&= \sum_{\boldsymbol{\beta} \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = 1} \text{Prob}\{\boldsymbol{\eta} = \boldsymbol{\beta}\} F \left(\begin{pmatrix} \beta_1 \mathbf{y}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \beta_2 \mathbf{y}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \beta_d \mathbf{y}^d \\ 1 \end{pmatrix} \right) \\
&\quad - \sum_{\boldsymbol{\beta} \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = -1} \text{Prob}\{\boldsymbol{\eta} = \boldsymbol{\beta}\} F \left(\begin{pmatrix} \beta_1 \mathbf{y}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \beta_2 \mathbf{y}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \beta_d \mathbf{y}^d \\ 1 \end{pmatrix} \right),
\end{aligned}$$

and (3.5) explicitly implies

$$\sum_{\boldsymbol{\beta} \in \mathbb{B}^d} F \left(\begin{pmatrix} \beta_1 \mathbf{y}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \beta_2 \mathbf{y}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \beta_d \mathbf{y}^d \\ 1 \end{pmatrix} \right) = 0.$$

For any constant c , combining the above two equalities we have

(3.6)

$$\begin{aligned}
& F(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d) \\
&= \sum_{\boldsymbol{\beta} \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = 1} (c + \text{Prob}\{\boldsymbol{\eta} = \boldsymbol{\beta}\}) F \left(\begin{pmatrix} \beta_1 \mathbf{y}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \beta_2 \mathbf{y}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \beta_d \mathbf{y}^d \\ 1 \end{pmatrix} \right) \\
&\quad + \sum_{\boldsymbol{\beta} \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = -1} (c - \text{Prob}\{\boldsymbol{\eta} = \boldsymbol{\beta}\}) F \left(\begin{pmatrix} \beta_1 \mathbf{y}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \beta_2 \mathbf{y}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \beta_d \mathbf{y}^d \\ 1 \end{pmatrix} \right).
\end{aligned}$$

If we let

$$c = \max_{\beta \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = -1} \text{Prob}\{\eta = \beta\},$$

then the coefficients of each term in (3.6) will be nonnegative. Therefore we shall be able to find $\tilde{\beta} \in \mathbb{B}^d$ (by enumerating the 2^d possible combinations if necessary, or by randomization), such that

$$(3.7) \quad F\left(\begin{pmatrix} \tilde{\beta}_1 \mathbf{y}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \tilde{\beta}_2 \mathbf{y}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \tilde{\beta}_d \mathbf{y}^d \\ 1 \end{pmatrix}\right) \geq \tau_0 F(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d),$$

where

$$\begin{aligned} \tau_0 &:= \left(\sum_{\beta \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = 1} (c + \text{Prob}\{\eta = \beta\}) + \sum_{\beta \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = -1} (c - \text{Prob}\{\eta = \beta\}) \right)^{-1} \\ &\geq \left(2^{d-1}c + \sum_{\beta \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = 1} \text{Prob}\{\eta = \beta\} + (2^{d-1} - 1)c \right)^{-1} \\ &\geq (2^{d-1} + 1 + 2^{d-1} - 1)^{-1} = 2^{-d}. \end{aligned}$$

Let us denote $\bar{\mathbf{z}}^k := \begin{pmatrix} \tilde{\beta}_k \mathbf{y}^k \\ 1 \end{pmatrix}$ for $k = 1, 2, \dots, d$. As $\|\mathbf{z}^k\| = \|\tilde{\beta}_k \mathbf{y}^k\| \leq 1$, we know that $(\bar{\mathbf{z}}^1, \bar{\mathbf{z}}^2, \dots, \bar{\mathbf{z}}^d)$ is a feasible solution for Problem (T_B) , and

$$\begin{aligned} F(\bar{\mathbf{z}}^1, \bar{\mathbf{z}}^2, \dots, \bar{\mathbf{z}}^d) &\geq \tau_0 F(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d) \\ &\geq 2^{-d} 2^{-\frac{d}{2}} (n+1)^{-\frac{d-2}{2}} v_{\max}(T_B) \\ &= 2^{-\frac{3d}{2}} (n+1)^{-\frac{d-2}{2}} v_{\max}(T_B). \end{aligned} \quad \square$$

One may notice that our proposed algorithm for solving Problem (T_B) is similar to Steps 2 and 3 of Algorithm 3.2, with only minor modification at Step 3, namely we choose a solution in the set $\arg \max \left\{ F\left(\begin{pmatrix} \beta_1 \mathbf{y}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \beta_2 \mathbf{y}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \beta_d \mathbf{y}^d \\ 1 \end{pmatrix}\right), \beta \in \mathbb{B}^d \right\}$, instead of choosing one in $\arg \max \left\{ F\left(\begin{pmatrix} \beta_1 \mathbf{y}^1/d \\ 1 \end{pmatrix}, \begin{pmatrix} \beta_2 \mathbf{y}^2/d \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \beta_d \mathbf{y}^d/d \\ 1 \end{pmatrix}\right), \beta \in \mathbb{B}^d \right\}$. The reason to divide d at Step 3 in Algorithm 3.2 (to solve Problem (B)) will become clear later. Finally, we remark again that it is unnecessary to enumerate all possible 2^d combinations in this step, as (3.6) suggests that a simple randomization process will serve the same purpose, especially when d is large. In the latter case, we will end up with a *polynomial-time randomized approximation algorithm*; otherwise, the computational complexity of the procedure is deterministic and is polynomial-time (for fixed d).

3.4. Returning a solution to the inhomogeneous model. Finally, we are led to the last step of the scheme. Step 4 of Algorithm 3.2 suggests a polarization formula $\bar{\mathbf{z}}(\beta) = \beta_1(d+1)\bar{\mathbf{z}}^1 + \sum_{k=2}^d \beta_k \bar{\mathbf{z}}^k$ with $\beta \in \mathbb{B}^d$ and $\beta_1 = \prod_{k=2}^d \beta_k = 1$. In fact, searching over all $\beta \in \mathbb{B}^d$ will possibly improve the solution, although the worst-case performance ratio will remain the same. Moreover, one may choose $\bar{\mathbf{z}}^1$ or any other $\bar{\mathbf{z}}^k$ to play the same role here; alternatively, one may enumerate $\beta_\ell(d+1)\bar{\mathbf{z}}^\ell + \sum_{1 \leq k \leq d, k \neq \ell} \beta_k \bar{\mathbf{z}}^k$ over all $\beta \in \mathbb{B}^d$ and $1 \leq \ell \leq d$ and take the best possible solution; again, this will not change the theoretical performance ratio. The polarization formula at Step 4 of Algorithm 3.2 works for all d , and we shall

complete the final stage of the proof of Theorem 3.3. That is, we shall prove that by letting $\mathbf{z} = \arg \max \left\{ p(\mathbf{0}); p\left(\frac{\mathbf{z}(\beta)}{z_h(\beta)}\right), \beta \in \mathbb{B}^d \text{ and } \beta_1 = \prod_{k=2}^n \beta_k = 1 \right\}$ with $\bar{\mathbf{z}}(\beta) := \beta_1(d+1)\bar{\mathbf{z}}^1 + \sum_{k=2}^d \beta_k \bar{\mathbf{z}}^k$, we have

$$(3.8) \quad p(\mathbf{z}) - v_{\min}(B) \geq \tau_B (v_{\max}(B) - v_{\min}(B)).$$

First, the solution $(\bar{\mathbf{z}}^1, \bar{\mathbf{z}}^2, \dots, \bar{\mathbf{z}}^d)$ as established at Step 3 satisfies $\|\mathbf{z}^k\| \leq 1/d$ (recall that we divided d in each term at Step 3), and $z_h^k = 1$ for all $1 \leq k \leq d$, and

$$(3.9) \quad \begin{aligned} F(\bar{\mathbf{z}}^1, \bar{\mathbf{z}}^2, \dots, \bar{\mathbf{z}}^d) &\geq d^{-d} 2^{-\frac{3d}{2}} (n+1)^{-\frac{d-2}{2}} v_{\max}(T_B) \\ &\geq 2^{-\frac{3d}{2}} d^{-d} (n+1)^{-\frac{d-2}{2}} v_{\max}(B). \end{aligned}$$

It is easy to see that

$$(3.10) \quad 2 \leq |z_h(\beta)| \leq 2d \quad \text{and} \quad \|\mathbf{z}(\beta)\| \leq (d+1)/d + (d-1)/d = 2.$$

Thus $\bar{\mathbf{z}}(\beta)/z_h(\beta)$ is a feasible solution of Problem (\bar{B}) , and so $f(\bar{\mathbf{z}}(\beta)/z_h(\beta)) \geq v_{\min}(\bar{B}) = v_{\min}(B)$. Moreover, we shall argue below that

$$(3.11) \quad \beta_1 = 1 \implies f(\bar{\mathbf{z}}(\beta)) \geq (2d)^d v_{\min}(B).$$

If this were not the case, then $f(\bar{\mathbf{z}}(\beta)/(2d)) < v_{\min}(B) \leq 0$. Notice that $\beta_1 = 1$ implies $z_h(\beta) > 0$, and so we would have

$$f\left(\frac{\bar{\mathbf{z}}(\beta)}{z_h(\beta)}\right) = \left(\frac{2d}{z_h(\beta)}\right)^d f\left(\frac{\bar{\mathbf{z}}(\beta)}{2d}\right) \leq f\left(\frac{\bar{\mathbf{z}}(\beta)}{2d}\right) < v_{\min}(B),$$

which contradicts the feasibility of $\bar{\mathbf{z}}(\beta)/z_h(\beta)$.

Suppose $\xi_1, \xi_2, \dots, \xi_d$ are i.i.d. random variables, each taking values 1 and -1 with equal probability. By Lemma 1 of [10], noticing $f(\bar{\mathbf{z}}(-\xi)) = f(-\bar{\mathbf{z}}(\xi)) = (-1)^d f(\bar{\mathbf{z}}(\xi))$, we have

$$\begin{aligned} d!F((d+1)\bar{\mathbf{z}}^1, \bar{\mathbf{z}}^2, \dots, \bar{\mathbf{z}}^d) &= \mathbb{E} \left[\prod_{i=1}^d \xi_i f(\bar{\mathbf{z}}(\xi)) \right] \\ &= \frac{1}{4} \mathbb{E} \left[f(\bar{\mathbf{z}}(\xi)) \mid \xi_1 = 1, \prod_{i=2}^d \xi_i = 1 \right] - \frac{1}{4} \mathbb{E} \left[f(\bar{\mathbf{z}}(\xi)) \mid \xi_1 = 1, \prod_{i=2}^d \xi_i = -1 \right] \\ &\quad - \frac{1}{4} \mathbb{E} \left[f(\bar{\mathbf{z}}(\xi)) \mid \xi_1 = -1, \prod_{i=2}^d \xi_i = 1 \right] + \frac{1}{4} \mathbb{E} \left[f(\bar{\mathbf{z}}(\xi)) \mid \xi_1 = -1, \prod_{i=2}^d \xi_i = -1 \right] \\ &= \frac{1}{4} \mathbb{E} \left[f(\bar{\mathbf{z}}(\xi)) \mid \xi_1 = 1, \prod_{i=2}^d \xi_i = 1 \right] - \frac{1}{4} \mathbb{E} \left[f(\bar{\mathbf{z}}(\xi)) \mid \xi_1 = 1, \prod_{i=2}^d \xi_i = -1 \right] \\ &\quad - \frac{1}{4} \mathbb{E} \left[f(\bar{\mathbf{z}}(-\xi)) \mid \xi_1 = 1, \prod_{i=2}^d \xi_i = (-1)^{d-1} \right] \\ &\quad + \frac{1}{4} \mathbb{E} \left[f(\bar{\mathbf{z}}(-\xi)) \mid \xi_1 = 1, \prod_{i=2}^d \xi_i = (-1)^d \right]. \end{aligned}$$

By inserting and canceling a constant term, the above expression further leads to

$$\begin{aligned}
d!F((d+1)\bar{z}^1, \bar{z}^2, \dots, \bar{z}^d) &= \mathbb{E} \left[\prod_{i=1}^d \xi_i f(\bar{z}(\xi)) \right] \\
&= \frac{1}{4} \mathbb{E} \left[(f(\bar{z}(\xi)) - (2d)^d v_{\min}(B)) \mid \xi_1 = 1, \prod_{i=2}^d \xi_i = 1 \right] \\
&\quad - \frac{1}{4} \mathbb{E} \left[(f(\bar{z}(\xi)) - (2d)^d v_{\min}(B)) \mid \xi_1 = 1, \prod_{i=2}^d \xi_i = -1 \right] \\
&\quad + \frac{(-1)^{d-1}}{4} \mathbb{E} \left[(f(\bar{z}(\xi)) - (2d)^d v_{\min}(B)) \mid \xi_1 = 1, \prod_{i=2}^d \xi_i = (-1)^{d-1} \right] \\
&\quad + \frac{(-1)^d}{4} \mathbb{E} \left[(f(\bar{z}(\xi)) - (2d)^d v_{\min}(B)) \mid \xi_1 = 1, \prod_{i=2}^d \xi_i = (-1)^d \right] \\
&\leq \frac{1}{2} \mathbb{E} \left[(f(\bar{z}(\xi)) - (2d)^d v_{\min}(B)) \mid \xi_1 = 1, \prod_{i=2}^d \xi_i = 1 \right],
\end{aligned}$$

where the last inequality is due to (3.11). Therefore, there is a binary vector $\tilde{\beta} \in \mathbb{B}^d$ with $\tilde{\beta}_1 = \prod_{i=2}^d \tilde{\beta}_i = 1$, such that

$$\begin{aligned}
f(\bar{z}(\tilde{\beta})) - (2d)^d v_{\min}(B) &\geq 2d!F((d+1)\bar{z}^1, \bar{z}^2, \dots, \bar{z}^d) \\
&\geq 2^{-\frac{3d}{2}+1} (d+1)! d^{-d} (n+1)^{-\frac{d-2}{2}} v_{\max}(B),
\end{aligned}$$

where the last step is due to (3.9).

Below we shall argue

$$\mathbf{z} = \arg \max \left\{ p(\mathbf{0}); p\left(\frac{\mathbf{z}(\beta)}{z_h(\beta)}\right), \beta \in \mathbb{B}^d \text{ and } \beta_1 = \prod_{k=2}^n \beta_k = 1 \right\}$$

satisfies (3.8). Indeed, if $-v_{\min}(B) \geq \tau_B (v_{\max}(B) - v_{\min}(B))$, then $\mathbf{0}$ trivially satisfies (3.8), and so does \mathbf{z} in this case. Otherwise, if $-v_{\min}(B) < \tau_B (v_{\max}(B) - v_{\min}(B))$, then we have

$$v_{\max}(B) > (1 - \tau_B) (v_{\max}(B) - v_{\min}(B)) \geq \frac{v_{\max}(B) - v_{\min}(B)}{2},$$

which implies

$$\begin{aligned}
f\left(\frac{\bar{z}(\tilde{\beta})}{2d}\right) - v_{\min}(B) &\geq (2d)^{-d} 2^{-\frac{3d}{2}+1} (d+1)! d^{-d} (n+1)^{-\frac{d-2}{2}} v_{\max}(B) \\
&\geq \tau_B (v_{\max}(B) - v_{\min}(B)).
\end{aligned}$$

The above inequality also implies that $f(\bar{z}(\tilde{\beta})/(2d)) > 0$. Recall that $\tilde{\beta}_1 = 1$ implies $z_h(\tilde{\beta}) > 0$, and thus $2d/z_h(\tilde{\beta}) \geq 1$ by (3.10). Therefore, we have

$$p(\mathbf{z}) \geq p\left(\frac{\mathbf{z}(\tilde{\beta})}{z_h(\tilde{\beta})}\right) = f\left(\frac{\bar{z}(\tilde{\beta})}{z_h(\tilde{\beta})}\right) = \left(\frac{2d}{z_h(\tilde{\beta})}\right)^d f\left(\frac{\bar{z}(\tilde{\beta})}{2d}\right) \geq f\left(\frac{\bar{z}(\tilde{\beta})}{2d}\right).$$

This shows that the solution \mathbf{z} satisfies (3.8) in both cases. Putting all the pieces together, Theorem 3.3 is proven by construction.

4. POLYNOMIAL OPTIMIZATION WITH CONVEX QUADRATIC CONSTRAINTS

In this section, we consider an extension of Problem (B), namely

$$(Q) \quad \begin{aligned} & \max && p(\mathbf{x}) \\ & \text{s.t.} && \mathbf{x}^T \mathbf{Q}_i \mathbf{x} \leq 1, \quad i = 1, 2, \dots, m, \\ & && \mathbf{x} \in \mathbb{R}^n, \end{aligned}$$

where $\mathbf{Q}_i \succeq 0$ for all $1 \leq i \leq m$, and $\sum_{i=1}^m \mathbf{Q}_i \succ 0$. By assuming that there is no constant term in the polynomial function $p(\mathbf{x})$, we know that $v_{\min}(Q) \leq 0 \leq v_{\max}(Q)$. Like in the previous section, we shall provide a polynomial-time randomized algorithm for approximately solving Problem (Q), with a worst-case relative performance ratio.

Algorithm 4.1.

Input: an n -dimensional d -th degree polynomial function $p(\mathbf{x})$ with $p(\mathbf{0}) = 0$, matrices $\mathbf{Q}_i \succeq 0$ for $i = 1, 2, \dots, m$ with $\sum_{i=1}^m \mathbf{Q}_i \succ 0$.

(1) Rewrite $p(\mathbf{x}) = F(\underbrace{\bar{\mathbf{x}}, \bar{\mathbf{x}}, \dots, \bar{\mathbf{x}}}_d)$ when $x_h = 1$ as in (3.2), where \mathbf{F} is an $(n+1)$ -dimensional d -th order super-symmetric tensor.

(2) Approximately solve the problem

$$\begin{aligned} & \max && F(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \dots, \bar{\mathbf{x}}^d) \\ & \text{s.t.} && (\bar{\mathbf{x}}^k)^T \begin{bmatrix} \mathbf{Q}_i & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \bar{\mathbf{x}}^k \leq 1, \quad k = 1, 2, \dots, d, \quad i = 1, 2, \dots, m \end{aligned}$$

using Algorithm 3 of [10], and get a feasible solution $(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d)$.

(3) Compute $(\bar{\mathbf{z}}^1, \bar{\mathbf{z}}^2, \dots, \bar{\mathbf{z}}^d) = \arg \max \left\{ F \left((\beta_1 \mathbf{y}_1^{1/d}), (\beta_2 \mathbf{y}_2^{1/d}), \dots, (\beta_d \mathbf{y}_d^{1/d}) \right), \beta \in \mathbb{B}^d \right\}$.

(4) Compute $\mathbf{z} = \arg \max \left\{ p(\mathbf{0}); p \left(\frac{\mathbf{z}(\beta)}{z_h(\beta)} \right), \beta \in \mathbb{B}^d \text{ and } \beta_1 = \prod_{k=2}^d \beta_k = 1 \right\}$, where

$$\bar{\mathbf{z}}(\beta) = \beta_1(d+1)\bar{\mathbf{z}}^1 + \sum_{k=2}^d \beta_k \bar{\mathbf{z}}^k.$$

Output: a feasible solution \mathbf{z} .

Theorem 4.2. Problem (Q) admits a polynomial-time randomized approximation algorithm (Algorithm 4.1) with relative approximation ratio τ_Q , i.e., a feasible solution \mathbf{z} can be found in polynomial-time with high probability, such that

$$p(\mathbf{z}) - v_{\min}(Q) \geq \tau_Q (v_{\max}(Q) - v_{\min}(Q)),$$

where $\tau_Q := \Omega \left(n^{-\frac{d-2}{2}} (\log m)^{-(d-1)} \right)$.

Our scheme for solving general Problem (Q) is similar to that for Problem (B) in the previous section. The main difference lies in Step 2, where a different relaxation model requires a different solution method to cope with. The method in question is Algorithm 3 of [10], a polynomial-time randomized algorithm. Therefore, Algorithm 4.1 is also a randomized algorithm, different to the deterministic one for Algorithm 3.2.

Theorem 4.3 (He, Li, and Zhang [10]). *Suppose F is a d -th order multilinear form, matrices $\mathbf{Q}_{i_k}^k \succeq 0$ and $\sum_{i_k=1}^{m_k} \mathbf{Q}_{i_k}^k \succ 0$ for all $1 \leq k \leq d, 1 \leq i_k \leq m_k$. The problem*

$$\begin{aligned} \max \quad & F(\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^d) \\ \text{s.t.} \quad & (\mathbf{x}^k)^T \mathbf{Q}_{i_k}^k \mathbf{x}^k \leq 1, \quad k = 1, 2, \dots, d, \quad i_k = 1, 2, \dots, m_k, \\ & \mathbf{x}^k \in \mathbb{R}^{n_k}, \quad k = 1, 2, \dots, d \end{aligned}$$

admits a polynomial-time randomized approximation algorithm with approximation ratio $\Omega\left(\left(\sqrt{n_1 n_2 \dots n_{d-2}} (\log m)^{-(d-1)}\right)^{-1}\right)$, where $m = \max_{1 \leq k \leq d} \{m_k\}$.

The proof of Theorem 4.2 is similar to that of Theorem 3.3. Here we shall only illustrate the main ideas and skip the details. By homogenizing $p(\mathbf{x})$, we may rewrite Problem (Q) as

$$\begin{aligned} (\bar{Q}) \quad \max \quad & f(\bar{\mathbf{x}}) \\ \text{s.t.} \quad & \bar{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ x_h \end{pmatrix}, \\ & \mathbf{x}^T \mathbf{Q}_i \mathbf{x} \leq 1, \quad \mathbf{x} \in \mathbb{R}^n, \quad i = 1, 2, \dots, m, \\ & x_h = 1, \end{aligned}$$

which can be relaxed to the inhomogeneous multilinear tensor form problem

$$\begin{aligned} (T_Q) \quad \max \quad & F(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \dots, \bar{\mathbf{x}}^d) \\ \text{s.t.} \quad & \bar{\mathbf{x}}^k = \begin{pmatrix} \mathbf{x}^k \\ x_h^k \end{pmatrix}, \quad k = 1, 2, \dots, d, \\ & (\mathbf{x}^k)^T \mathbf{Q}_i \mathbf{x}^k \leq 1, \quad \mathbf{x}^k \in \mathbb{R}^n, \quad k = 1, 2, \dots, d, \quad i = 1, 2, \dots, m, \\ & x_h^k = 1, \quad k = 1, 2, \dots, d, \end{aligned}$$

where $F(\underbrace{\bar{\mathbf{x}}, \bar{\mathbf{x}}, \dots, \bar{\mathbf{x}}}_d) = f(\bar{\mathbf{x}})$ with \mathbf{F} being super-symmetric. We then further relax

Problem (T_Q) to the homogeneous multilinear tensor form Problem $(T_Q(\sqrt{2}))$, with

$$\begin{aligned} (T_Q(t)) \quad \max \quad & F(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \dots, \bar{\mathbf{x}}^d) \\ & (\bar{\mathbf{x}}^k)^T \hat{\mathbf{Q}}_i \bar{\mathbf{x}}^k \leq t^2, \quad k = 1, 2, \dots, d, \quad i = 1, 2, \dots, m, \\ & \bar{\mathbf{x}}^k \in \mathbb{R}^{n+1}, \quad k = 1, 2, \dots, d, \end{aligned}$$

where $\hat{\mathbf{Q}}_i = \begin{bmatrix} \mathbf{Q}_i & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$ for $i = 1, 2, \dots, m$.

By Theorem 4.3, for any $t > 0$, Problem $(T_Q(t))$ admits a polynomial-time approximation randomized algorithm with approximation ratio τ_Q , and $v_{\max}(T_Q(t)) = t^d v_{\max}(T_Q(1))$. Thus the approximate solution $(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d)$ found by Step 2 of Algorithm 4.1 satisfies

$$F(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d) \geq \tau_Q v_{\max}(T_Q(1)) = (\sqrt{2})^{-d} \tau_Q v_{\max}(T_Q(\sqrt{2})) \geq \tau_Q v_{\max}(T_Q).$$

By noticing that $(y_h^k)^2 \leq (\bar{\mathbf{y}}^k)^T \hat{\mathbf{Q}}_1 \bar{\mathbf{y}}^k \leq 1$, we may again apply Lemma 3.6 to $(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d)$, and use the same argument as in the proof of Theorem 3.7.

Let $c = \max_{\boldsymbol{\beta} \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = -1} \text{Prob}\{\boldsymbol{\eta} = \boldsymbol{\beta}\}$, where $\boldsymbol{\eta} = (\eta_1, \eta_2, \dots, \eta_d)^T$ and its components are independent random variables, each takes values 1 and -1 with

$E[\eta_k] = y_h^k$ for $k = 1, 2, \dots, d$. Then we are able to find $\tilde{\beta} \in \mathbb{B}^d$, such that

$$\begin{aligned} F\left(\begin{pmatrix} \tilde{\beta}_1 \mathbf{y}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \tilde{\beta}_2 \mathbf{y}^2 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} \tilde{\beta}_d \mathbf{y}^d \\ 1 \end{pmatrix}\right) &\geq \tau_0 F(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d) \\ &\geq 2^{-d} F(\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \dots, \bar{\mathbf{y}}^d) \\ &\geq \tau_Q v_{\max}(T_Q). \end{aligned}$$

This proves the following theorem as a byproduct.

Theorem 4.4. *Problem (T_Q) admits a polynomial-time randomized approximation algorithm with approximation ratio τ_Q .*

To prove the main theorem in this section (Theorem 4.2), we only need to check the feasibility of \mathbf{z} generated by Algorithm 4.1, while the worst-case performance ratio can be proven by a similar argument in Section 3.4. Indeed, $(\bar{\mathbf{z}}^1, \bar{\mathbf{z}}^2, \dots, \bar{\mathbf{z}}^d)$ at Step 3 of Algorithm 4.1 satisfies

$$(\mathbf{z}^k)^T \mathbf{Q}_i \mathbf{z}^k \leq 1/d^2 \quad \forall 1 \leq i \leq m, 1 \leq k \leq d.$$

For any binary vector $\beta \in \mathbb{B}^d$, as $\bar{\mathbf{z}}(\beta) = \beta_1(d+1)\bar{\mathbf{z}}^1 + \sum_{k=2}^d \beta_k \bar{\mathbf{z}}^k$, we have $2 \leq |z_h(\beta)| \leq 2d$. Noticing by the Cauchy-Schwarz inequality,

$$|(\mathbf{z}^j)^T \mathbf{Q}_i \mathbf{z}^k| \leq \|\mathbf{Q}_i^{1/2} \mathbf{z}^j\| \cdot \|\mathbf{Q}_i^{1/2} \mathbf{z}^k\| \leq 1/d^2 \quad \forall 1 \leq i \leq m, 1 \leq j, k \leq d,$$

it follows that

$$(\mathbf{z}(\beta))^T \mathbf{Q}_i \mathbf{z}(\beta) \leq 2d \cdot 2d \cdot 1/d^2 = 4 \quad \forall 1 \leq i \leq m.$$

Thus $\mathbf{z}(\beta)/z_h(\beta)$ is a feasible solution to Problem (Q) , which implies \mathbf{z} is also feasible.

We remark here that Problem (Q) includes as a special case the optimization of a general polynomial function over a central-symmetric polytope:

$$\begin{aligned} \max \quad & p(\mathbf{x}) \\ \text{s.t.} \quad & -1 \leq (\mathbf{a}^i)^T \mathbf{x} \leq 1, \quad i = 1, 2, \dots, m, \\ & \mathbf{x} \in \mathbb{R}^n, \end{aligned}$$

with $\text{rank}(\mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^m) = n$.

Before concluding this section, let us embark on a treatment for a special low degree case of Problem (Q) , where an *absolute* approximation ratio is possible (assuming the objective polynomial has no constant term) instead of a mere *relative* one as stipulated in Theorem 4.2. In fact, for Problem (Q) , when $d = 2$, an $\Omega(1/\log m)$ -approximation ratio was easily derived from the homogenous relaxation problem by Nemirovski et al. [29]. The following result presents a very useful case when $d = 3$. However, for $d \geq 4$, Proposition 3.1 has ruled out such approximation ratios, even for the case $m = 1$.

Theorem 4.5. *If $d = 3$ and the quadratic form in $p(\mathbf{x})$ is positive semidefinite, then Problem (Q) admits a polynomial-time randomized approximation algorithm with approximation ratio $\Omega\left(\frac{1}{\sqrt{n} \log^2 m}\right)$.*

Proof. Denote the cubic, quadratic and linear forms in $p(\mathbf{x})$ to be $f_3(\mathbf{x})$, $f_2(\mathbf{x})$ and $f_1(\mathbf{x})$, respectively, with $f_2(\mathbf{x})$ being positive semidefinite. By homogenization as (3.2), we have

$$p(\mathbf{x}) = f_3(\mathbf{x}) + f_2(\mathbf{x}) + f_1(\mathbf{x}) = F \left(\begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \right) = f \left(\begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \right),$$

where tensor $\mathbf{F} \in \mathbb{R}^{(n+1)^3}$ is super-symmetric and its last component is 0. As before, Problem (Q) can be relaxed to the multilinear form optimization Problem $(T_Q(\sqrt{2}))$ when $d = 3$. By Theorem 4.3, we may find a feasible solution $\left(\begin{pmatrix} \mathbf{y}^1 \\ y_h^1 \end{pmatrix}, \begin{pmatrix} \mathbf{y}^2 \\ y_h^2 \end{pmatrix}, \begin{pmatrix} \mathbf{y}^3 \\ y_h^3 \end{pmatrix} \right)$ of Problem $(T_Q(1/3))$ in polynomial-time, satisfying $(\mathbf{y}^k)^T \mathbf{Q}_i \mathbf{y}^k \leq \frac{1}{9}$ and $|y_h^k| \leq \frac{1}{3}$ for $k = 1, 2, 3$ and $i = 1, 2, \dots, m$, such that

$$\begin{aligned} F \left(\begin{pmatrix} \mathbf{y}^1 \\ y_h^1 \end{pmatrix}, \begin{pmatrix} \mathbf{y}^2 \\ y_h^2 \end{pmatrix}, \begin{pmatrix} \mathbf{y}^3 \\ y_h^3 \end{pmatrix} \right) &\geq \Omega \left(\frac{1}{\sqrt{n} \log^2 m} \right) v(T_Q(1/3)) \\ &= \Omega \left(\frac{1}{\sqrt{n} \log^2 m} \right) \frac{v(T_Q(\sqrt{2}))}{54\sqrt{2}} \\ &\geq \Omega \left(\frac{1}{\sqrt{n} \log^2 m} \right) v(Q). \end{aligned}$$

Then by Lemma 3.6 and (3.7), we may adjust $\left(\begin{pmatrix} \mathbf{y}^1 \\ y_h^1 \end{pmatrix}, \begin{pmatrix} \mathbf{y}^2 \\ y_h^2 \end{pmatrix}, \begin{pmatrix} \mathbf{y}^3 \\ y_h^3 \end{pmatrix} \right)$ to $\left(\begin{pmatrix} \mathbf{z}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{z}^2 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{z}^3 \\ 1 \end{pmatrix} \right)$, satisfying $(\mathbf{z}^k)^T \mathbf{Q}_i \mathbf{z}^k \leq \frac{1}{9}$ for $k = 1, 2, 3$ and $i = 1, 2, \dots, m$, such that

$$F \left(\begin{pmatrix} \mathbf{z}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{z}^2 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{z}^3 \\ 1 \end{pmatrix} \right) \geq 2^{-3} F \left(\begin{pmatrix} \mathbf{y}^1 \\ y_h^1 \end{pmatrix}, \begin{pmatrix} \mathbf{y}^2 \\ y_h^2 \end{pmatrix}, \begin{pmatrix} \mathbf{y}^3 \\ y_h^3 \end{pmatrix} \right) \geq \Omega \left(\frac{1}{\sqrt{n} \log^2 m} \right) v(Q).$$

It is easy to verify the following link identity (see also Lemma 1 of [10])

$$\begin{aligned} (4.1) \quad & f \left(\begin{pmatrix} \mathbf{z}^1 + \mathbf{z}^2 + \mathbf{z}^3 \\ 3 \end{pmatrix} \right) + f \left(\begin{pmatrix} \mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3 \\ -1 \end{pmatrix} \right) + f \left(\begin{pmatrix} \mathbf{z}^2 - \mathbf{z}^3 - \mathbf{z}^1 \\ -1 \end{pmatrix} \right) + f \left(\begin{pmatrix} \mathbf{z}^3 - \mathbf{z}^1 - \mathbf{z}^2 \\ -1 \end{pmatrix} \right) \\ &= 24F \left(\begin{pmatrix} \mathbf{z}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{z}^2 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{z}^3 \\ 1 \end{pmatrix} \right) \geq \Omega \left(\frac{1}{\sqrt{n} \log^2 m} \right) v(Q). \end{aligned}$$

Therefore, the largest term of $f(\cdot)$ in (4.1) is at least in order of $\Omega \left(\frac{1}{\sqrt{n} \log^2 m} \right) v(Q)$.

If $f \left(\begin{pmatrix} \mathbf{z}^1 + \mathbf{z}^2 + \mathbf{z}^3 \\ 3 \end{pmatrix} \right)$ is this largest term, then $\mathbf{y} := \frac{\mathbf{z}^1 + \mathbf{z}^2 + \mathbf{z}^3}{3}$ satisfies the constraint of Problem (Q), and

$$p(\mathbf{y}) = f \left(\begin{pmatrix} \mathbf{y} \\ 1 \end{pmatrix} \right) = \frac{f \left(\begin{pmatrix} 3\mathbf{y} \\ 3 \end{pmatrix} \right)}{27} \geq \Omega \left(\frac{1}{\sqrt{n} \log^2 m} \right) v(Q).$$

Otherwise by symmetricity we may, without loss of generality, assume $f \left(\begin{pmatrix} \mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3 \\ -1 \end{pmatrix} \right)$ is this largest term, then $\mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3$ satisfies the constraint of Problem (Q), and

$$\begin{aligned} p(\mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3) &= f_3(\mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3) + f_2(\mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3) + f_1(\mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3) \\ &\geq f_3(\mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3) - f_2(\mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3) + f_1(\mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3) \\ &= f \left(\begin{pmatrix} \mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3 \\ -1 \end{pmatrix} \right) \geq \Omega \left(\frac{1}{\sqrt{n} \log^2 m} \right) v(Q), \end{aligned}$$

where the first inequality is due to the fact that $f_2(\mathbf{x})$ is positive semidefinite. Finally,

$$\arg \max \{p(\mathbf{y}), p(\mathbf{z}^1 - \mathbf{z}^2 - \mathbf{z}^3), p(\mathbf{z}^2 - \mathbf{z}^3 - \mathbf{z}^1), p(\mathbf{z}^3 - \mathbf{z}^1 - \mathbf{z}^2)\}$$

is the approximate solution with approximation ratio $\Omega\left(\frac{1}{\sqrt{n} \log^2 m}\right)$. \square

In fact, the condition of Theorem 4.5 that the quadratic form is positive semidefinite can be made weaker: the quadratic form is not negative semidefinite, and the absolute value of its smallest eigenvalue is bounded by some constant multiplying its largest eigenvalue. Its proof can be similarly derived as of Theorem 4.5, and is omitted here.

5. POLYNOMIAL OPTIMIZATION OVER A GENERAL COMPACT SET

In this section we consider polynomial optimization model in a generic constraint format:

$$(G) \quad \begin{array}{ll} \max & p(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in S, \end{array}$$

where $S \subseteq \mathbb{R}^n$ is a general convex compact set. We are concerned with *polynomial-time approximation algorithms* for solving Problem (G). Our approach makes use of the famous *Löwner-John ellipsoids* (see e.g. [4, 28]):

Theorem 5.1 (Löwner-John). *Let $S \subseteq \mathbb{R}^n$ be a bounded convex set with nonempty interior. Then:*

- (1) *there exists a unique largest volume ellipsoid $E_{in} = \{\mathbf{A}\mathbf{x} + \mathbf{a} \mid \|\mathbf{x}\| \leq 1, \mathbf{x} \in \mathbb{R}^n\} \subset G$, whose n times linear-size larger ellipsoid $\{n\mathbf{A}\mathbf{x} + \mathbf{a} \mid \|\mathbf{x}\| \leq 1, \mathbf{x} \in \mathbb{R}^n\} \supset G$, and if in addition G is central-symmetric, then $\{\sqrt{n}\mathbf{A}\mathbf{x} + \mathbf{a} \mid \|\mathbf{x}\| \leq 1, \mathbf{x} \in \mathbb{R}^n\} \supset G$;*
- (2) *there exists a unique smallest volume ellipsoid $E_{out} = \{\mathbf{B}\mathbf{x} + \mathbf{b} \mid \|\mathbf{x}\| \leq 1, \mathbf{x} \in \mathbb{R}^n\} \supset G$, whose n times linear-size smaller ellipsoid $\{\mathbf{B}\mathbf{x}/n + \mathbf{b} \mid \|\mathbf{x}\| \leq 1, \mathbf{x} \in \mathbb{R}^n\} \subset G$, and if in addition G is central-symmetric, then $\{\mathbf{B}\mathbf{x}/\sqrt{n} + \mathbf{b} \mid \|\mathbf{x}\| \leq 1, \mathbf{x} \in \mathbb{R}^n\} \subset G$.*

Armed with the above theorem, if we are able to find the Löwner-John ellipsoid (either E_{in} or E_{out}) of the feasible region S in polynomial-time, then the following algorithm approximately solves Problem (G) with a worst-case performance ratio.

Algorithm 5.2.

Input: an n -dimensional d -th degree polynomial function $p(\mathbf{x})$ and a set $S \subseteq \mathbb{R}^n$.

- (1) Find a scalar $t \in \mathbb{R}$, a vector $\mathbf{b} \in \mathbb{R}^n$, and a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ with $\text{rank}(\mathbf{A}) = m \leq n$, such that two co-centered ellipsoids $E_1 = \{\mathbf{A}\mathbf{u} + \mathbf{b} \mid \|\mathbf{u}\| \leq 1, \mathbf{u} \in \mathbb{R}^m\}$ and $E_2 = \{\mathbf{A}\mathbf{u} + \mathbf{b} \mid \|\mathbf{u}\| \leq t, \mathbf{u} \in \mathbb{R}^m\}$ satisfy $E_1 \subseteq S \subseteq E_2$.
- (2) Compute polynomial function $p_0(\mathbf{u}) = p(\mathbf{A}\mathbf{u} + \mathbf{b})$, where $\mathbf{u} \in \mathbb{R}^m$.
- (3) Apply Algorithm 3.2 with input $p_0(\mathbf{x}) - p_0(\mathbf{0})$ and output $\mathbf{z} \in \mathbb{R}^m$.

Output: a feasible solution $\mathbf{A}\mathbf{z} + \mathbf{b}$.

The key result in this section is the following theorem.

Theorem 5.3. *Suppose that $\{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq 1\} \subseteq S \subseteq \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq t\}$ for some $t \geq 1$. Then, Problem (G) admits a polynomial-time approximation algorithm with relative approximation ratio $\tau_G(t)$, i.e., a feasible solution $\mathbf{z} \in S$ can be found in polynomial-time, satisfying*

$$(5.1) \quad p(\mathbf{z}) - v_{\min}(G) \geq \tau_G(t) (v_{\max}(G) - v_{\min}(G)),$$

where $\tau_G(t) := (d+1)!(2d)^{-2d}(n+1)^{-\frac{d-2}{2}}(t^2+1)^{-\frac{d}{2}}$.

Proof. By homogenizing the object function of Problem (G), we get the equivalent problem

$$\begin{aligned} (\bar{G}) \quad & \max \quad f(\bar{\mathbf{x}}) \\ \text{s.t.} \quad & \bar{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ x_h \end{pmatrix}, \\ & \mathbf{x} \in S, x_h = 1, \end{aligned}$$

where $f(\bar{\mathbf{x}}) = p(\mathbf{x})$ if $x_h = 1$, and $f(\bar{\mathbf{x}})$ is an $(n+1)$ -dimensional homogeneous polynomial function of degree d .

If we write $f(\bar{\mathbf{x}}) = F(\overbrace{\bar{\mathbf{x}}, \bar{\mathbf{x}}, \dots, \bar{\mathbf{x}}}^d)$ with \mathbf{F} being super-symmetric, Problem (\bar{G}) can be relaxed to the inhomogeneous multilinear tensor form problem

$$\begin{aligned} \max \quad & F(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \dots, \bar{\mathbf{x}}^d) \\ \text{s.t.} \quad & \bar{\mathbf{x}}^k = \begin{pmatrix} \mathbf{x}^k \\ x_h^k \end{pmatrix}, k = 1, 2, \dots, d, \\ & \mathbf{x}^k \in S, x_h^k = 1, k = 1, 2, \dots, d. \end{aligned}$$

Recall that we have previously defined

$$\begin{aligned} (T_B(t)) \quad & \max \quad F(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \dots, \bar{\mathbf{x}}^d) \\ \text{s.t.} \quad & \|\bar{\mathbf{x}}^k\| \leq t, \bar{\mathbf{x}}^k \in \mathbb{R}^{n+1}, k = 1, 2, \dots, d. \end{aligned}$$

As $\mathbf{x}^k \in S \subseteq \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq t\}$, it follows that $\|\bar{\mathbf{x}}^k\| \leq \sqrt{t^2+1}$. Thus Problem $(T_B(\sqrt{t^2+1}))$ is a relaxation of Problem (\bar{G}) and $v_{\max}(T_B(\sqrt{t^2+1})) \geq v_{\max}(\bar{G}) = v_{\max}(G)$. The rest of the proof follows similarly as that in Section 3.4. \square

Observe that any ellipsoid can be linearly transformed into the Euclidean ball, by a variable transformation if necessary, we are led to the main result in this section.

Corollary 5.4. *Let $S \subseteq \mathbb{R}^n$ be a given compact set. Suppose two co-centered ellipsoids $E_1 = \{\mathbf{A}\mathbf{u} + \mathbf{b} \mid \|\mathbf{u}\| \leq 1, \mathbf{u} \in \mathbb{R}^n\}$ and $E_2 = \{\mathbf{A}\mathbf{u} + \mathbf{b} \mid \|\mathbf{u}\| \leq t, \mathbf{u} \in \mathbb{R}^n\}$ can be found in polynomial-time, satisfying $E_1 \subseteq S \subseteq E_2$. Then Problem (G) admits a polynomial-time approximation algorithm (Algorithm 5.2) with relative approximation ratio $\tau_G(t)$.*

We would like to remark that the set S in Theorem 5.3 and Corollary 5.4 does not need to be convex, as long as the two required ellipsoids are in place. Therefore, Problem (G) generally includes cases of nonconvex set S . However, the famous Löwner-John theorem guarantees the existence of such inner and outer ellipsoids required in Corollary 5.4 for any bounded convex set, with $t = n$ for S being noncentral-symmetric, and $t = \sqrt{n}$ for S being central-symmetric. Thus, if we are able to find a pair of ellipsoids (E_1, E_2) in polynomial-time for S , then Problem (G) can be solved by a polynomial-time approximation algorithm with relative approximation ratio $\tau_G(t)$. Indeed, it is possible to compute in polynomial-time

the Löwner-John ellipsoids in several interesting cases. Below is a list of such cases (assuming S is bounded); for details one is referred to [4, 28]:

- $S = \{\mathbf{x} \in \mathbb{R}^n \mid (\mathbf{a}^i)^T \mathbf{x} \leq b_i, i = 1, 2, \dots, m\}$;
- $S = \text{conv}\{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m\}$, where $\mathbf{x}^i \in \mathbb{R}^n, i = 1, 2, \dots, m$;
- $S = \bigcap_{i=1}^m E_i$, where E_i is an ellipsoid in $\mathbb{R}^n, i = 1, 2, \dots, m$;
- $S = \text{conv}\{\bigcup_{i=1}^m E_i\}$, where E_i is an ellipsoid in $\mathbb{R}^n, i = 1, 2, \dots, m$;
- $S = \sum_{i=1}^m E_i := \{\sum_{i=1}^m \mathbf{x}^i \mid \mathbf{x}^i \in E_i, i = 1, 2, \dots, m\}$, where E_i is an ellipsoid in $\mathbb{R}^n, i = 1, 2, \dots, m$.

By Corollary 5.4, and the computability of the Löwner-John ellipsoids ([4, 28]) as we discussed above, we conclude that for Problem (G) with the constraint set S belonging to any of the above cases, then there is a polynomial-time approximation algorithm with a relative approximation quality assurance. In particular, the ratio is $\tau_G(\sqrt{m}) = \Omega\left(n^{-\frac{d-2}{2}} m^{-\frac{d}{2}}\right)$ for the last case, and $\tau_G(n) = \Omega\left(n^{-\frac{3d-2}{2}}\right)$ for the other cases.

We shall also remark that Problem (Q) : $\max_{\mathbf{x}^T \mathbf{Q}_i \mathbf{x} \leq 1, i=1,2,\dots,m} p(\mathbf{x})$, discussed in Section 4, may in principle be solved by directly applying Corollary 5.4 as well. If we adopt that approach (Algorithm 5.2), then the relative approximation ratio will be $\tau_G(\sqrt{n}) = \Omega\left(n^{-\frac{2d-2}{2}}\right)$, which will prevail if m is exceedingly large. By taking the best of the two, the quality ratio in Theorem 4.2 can be improved to $\Omega\left(\max\left\{n^{-\frac{d-2}{2}}(\log m)^{-(d-1)}, n^{-\frac{2d-2}{2}}\right\}\right)$.

Our investigation quite naturally leads to a question which is of a general geometric interest itself. Consider the intersection of m co-centered ellipsoids in \mathbb{R}^n as a geometric structure to be considered. Let $\mathcal{E}_{m,n}$ be the collection of all such ellipsoids, or more specifically,

$$\mathcal{E}_{m,n} := \left\{ \bigcap_{i=1}^m E_i \mid E_i = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^T \mathbf{Q}_i \mathbf{x} \leq 1\} \text{ with } \mathbf{Q}_i \succeq 0 \right. \\ \left. \text{for } i = 1, 2, \dots, m, \text{ and } \sum_{i=1}^m \mathbf{Q}_i \succ 0 \right\}.$$

For any central-symmetric, convex and compact set S centered at \mathbf{b} , there exist $E_{m,n} \in \mathcal{E}_{m,n}$ and $t \geq 1$, such that $\mathbf{b} + E_{m,n} \subseteq S \subseteq \mathbf{b} + tE_{m,n}$. Obviously, one can naturally define

$$t(S; m, n) := \inf \{t \mid E_{m,n} \in \mathcal{E}_{m,n} \text{ such that } \mathbf{b} + E_{m,n} \subseteq S \subseteq \mathbf{b} + tE_{m,n}\}$$

$$\theta(m, n) := \sup \{t(S; m, n) \mid S \text{ is a central-symmetric convex compact set in } \mathbb{R}^n\}.$$

The famous Löwner-John theorem states that $\theta(1, n) = \sqrt{n}$. Naturally, $\theta(\infty, n) = 1$, because any central-symmetric convex set can be expressed by the intersection of an infinite number of co-centered ellipsoids. It is interesting to compute $\theta(m, n)$ for general m and n . It is of course trivial to observe that $\theta(m, n)$ is monotonically decreasing in m for any fixed n . Anyway, if we were able to compute $\theta(m, n)$, then Theorem 4.2 suggests a polynomial-time randomized approximation algorithm of Problem (G) with relative approximation ratio $(\theta(m, n))^{-d} \tau_Q = \Omega\left((\theta(m, n))^{-d} n^{-\frac{d-2}{2}} (\log m)^{-(d-1)}\right)$.

6. NUMERICAL RESULTS

In this section we shall test the performance of the approximation algorithms proposed in this paper, to give the readers an impression about how our algorithms work in practice. We shall focus on Problem (B) with $d = 4$, i.e., maximizing an inhomogeneous quartic polynomial function over the Euclidean ball. All the numerical computations are performed on an Intel(R) Core(TM) i7-2620M CPU 2.70GHz+2.70GHz computer with 4GB of RAM. The supporting software is MATLAB 7.12.0.635 (R2011a).

Specifically, the model being tested is

$$(E) \quad \begin{aligned} \max \quad & p(\mathbf{x}) = F_4(\mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x}) + F_3(\mathbf{x}, \mathbf{x}, \mathbf{x}) + F_2(\mathbf{x}, \mathbf{x}) + F_1(\mathbf{x}) \\ \text{s.t.} \quad & \|\mathbf{x}\| \leq 1, \mathbf{x} \in \mathbb{R}^n, \end{aligned}$$

where $\mathbf{F}_4 \in \mathbb{R}^{n^4}$, $\mathbf{F}_3 \in \mathbb{R}^{n^3}$, $\mathbf{F}_2 \in \mathbb{R}^{n^2}$, and $\mathbf{F}_1 \in \mathbb{R}^n$, are super-symmetric tensors of order 4, 3, 2 and 1, respectively. The fourth order tensor \mathbf{F}_4 is generated randomly, with its n^4 entries taken from i.i.d. standard normal distributions, followed by averaging the corresponding entries to make it being super-symmetric; the other lower order tensors \mathbf{F}_3 , \mathbf{F}_2 and \mathbf{F}_1 are generated in the same manner. We then apply Algorithm 3.2 to get a feasible solution which has a guaranteed worst-case performance ratio.

For the purpose of making a comparison, we also compute an upper bound of the optimal value of Problem (E). As in (3.2), we let $F(\bar{\mathbf{x}}, \bar{\mathbf{x}}, \bar{\mathbf{x}}, \bar{\mathbf{x}}) = f(\bar{\mathbf{x}}) = p(\mathbf{x})$ when $x_h = 1$, and $\mathbf{F} \in \mathbb{R}^{(n+1)^4}$ is super-symmetric. Problem (E) can be relaxed to

$$\begin{aligned} \max \quad & F(\bar{\mathbf{x}}, \bar{\mathbf{x}}, \bar{\mathbf{x}}, \bar{\mathbf{x}}) \\ \text{s.t.} \quad & \|\bar{\mathbf{x}}\| \leq \sqrt{2}, \bar{\mathbf{x}} \in \mathbb{R}^{n+1}. \end{aligned}$$

Let $\mathbf{y} = \text{vec}(\bar{\mathbf{x}}\bar{\mathbf{x}}^T) \in \mathbb{R}^{(n+1)^2}$, and rewrite \mathbf{F} as an $(n+1)^2 \times (n+1)^2$ matrix \mathbf{F}' ; Problem (E) is further relaxed to

$$\begin{aligned} \max \quad & F'(\mathbf{y}, \mathbf{y}) = \mathbf{y}^T \mathbf{F}' \mathbf{y} \\ \text{s.t.} \quad & \|\mathbf{y}\| \leq 2, \mathbf{y} \in \mathbb{R}^{(n+1)^2}. \end{aligned}$$

The optimal value of the above problem is $4\lambda_{\max}(\mathbf{F}')$, which we use as an upper bound.

By Theorem 3.3, Algorithm 3.2 possesses a theoretic worst-case relative performance ratio of $2^{-10} \cdot 5! \cdot 4^{-8}(n+1)^{-1} = \Omega(1/n)$. The numerical results of Problem (E) are listed in Table 1, in which the objective value of the feasible solution and upper bound are from the average of 10 random generated instances. Based on the observation of the simulation results, by comparing with the upper bound $4\lambda_{\max}(\mathbf{F}')$ (which might be very loose), the absolute performance ratio is about $\Omega(1/\sqrt{n})$, rather than a theoretical relative ratio $\Omega(1/n)$.

With regard to the computational efforts, we report that Algorithm 3.2 ran fairly fast. For instance, for $n = 70$ we were able to get a feasible solution within seconds, while computing the upper bound of the optimal value ($4\lambda_{\max}(\mathbf{F}')$) cost much more computational time. For $n \geq 80$, however, our computer reported to run out of memory in the experiments: a problem purely due to the sheer size of the input data.

TABLE 1. Numerical results of Problem (E)

n	3	5	10	20	30	40	50	60	70
Obj. val. of feas. sol. v	0.342	0.434	0.409	0.915	0.671	0.499	0.529	0.663	0.734
Upper bound \bar{v}	10.5	16.1	26.7	51.7	74.4	97.8	121.1	143.6	167.1
(Absolute) ratio v/\bar{v} (%)	3.26	2.70	1.53	1.77	0.90	0.51	0.44	0.46	0.44
$n \cdot \text{ratio}$	0.098	0.135	0.153	0.354	0.271	0.204	0.218	0.277	0.307
$\sqrt{n} \cdot \text{ratio}$	0.056	0.060	0.048	0.079	0.049	0.032	0.031	0.036	0.037

6.1. Local improvements and the SOS method. The theoretical worst-case performance ratios that we have developed so far are certainly very conservative, as observed in the previous testings. It will be desirable to design a more realistic test procedure to know how good the solutions actually are. In fact, we observe that the Euclidean norm of the solution obtained by Algorithm 3.2 is usually far from one. Therefore, it is straightforward to do one more step of line search along the direction of the solution obtained. It is simply a univariate quartic polynomial optimization over an interval, albeit the theoretical performance ratio cannot be improved. Another point to note is that we can always improve the quality of the solution by applying a local improvement subroutine on our heuristic solution. Here in this subsection, we are going to apply recently developed local improvement procedure *maximum block improvement* (MBI) by Chen et al. [5], starting from the approximate solution generated by Algorithm 3.2.

In order to evaluate the true quality of our approximate solution it is desirable to probe the optimal value, instead of using the loose upper bound. For this purpose we call the SOS approach for general polynomial optimization problems. As mentioned earlier, the SOS method works quite well for low dimension problems. It either outputs the optimal value, or an upper bound of the optimal value for a maximization problem, each of which can be a benchmark to test the performance of our algorithm for solving Problem (E).

In this set of experiments we restrict ourselves to the low dimensional cases, say $n \leq 15$. We take the feasible approximate solution as a starting point to be followed by the local improvement subroutine MBI in [5] to obtain a local optimal solution. For comparison, we apply GloptiPoly 3 of Henrion et al. [13] (the SOS method) to get the optimal value or its upper bound for the same test instance.

For the case $n = 15$, we generate 15 random instances of Problem (E). The objective value of the feasible solution obtained by Algorithm 3.2, that of the feasible solution followed by the MBI method, and the optimal value or upper bound generated by GloptiPoly 3, as described above, are shown in Figure 1. In fact, the optimal value lies between the objective value by Algorithm 3.2 with the MBI method (the red square dot) and the upper bound by GloptiPoly 3 (the green triangle dot) for each instance in Figure 1. In most cases, GloptiPoly 3 is able to provide optimality certificate. The results are quite telling: Algorithm 3.2 followed by the local improvement procedure yields near optimal solutions in many cases, at least for low dimension problems.

In another setting, for each $n = 5, 10, 15$, we generate 100 random instances of Problem (E), and count the number of instances that the objective value of the feasible solution followed by the MBI method *equals* (within tolerance 10^{-6}) that obtained by GloptiPoly 3 (either the optimal value or its upper bound). Note that

when the two values are equal, the optimality guarantees since they are lower and upper bounds of the optimal value. The outputs of the testings is 37 for $n = 5$, 28 for $n = 10$, and 30 for $n = 15$. In this sense, our heuristic algorithm (with local improvement) and the SOS method are indeed complementary to each other.

Regarding the computational complexity, as mentioned in Section 3, for Algorithm 3.2 it is $O(n^4)$ for Problem (E). However, for the SOS method, the first relaxation requires solving an SDP problem of dimension $O(n^4)$, which is much more demanding than what Algorithm 3.2 is supposed to do. Though Algorithm 3.2 only attempts to find an approximate solution, it can handle much larger problems than an exact algorithm does. In some cases, the approximate solutions meet the bounds provided by GloptiPloy 3, thus the optimality of the solutions are certified. The approximation algorithms as proposed in this paper can therefore be viewed as complementary to the exact solution methods.

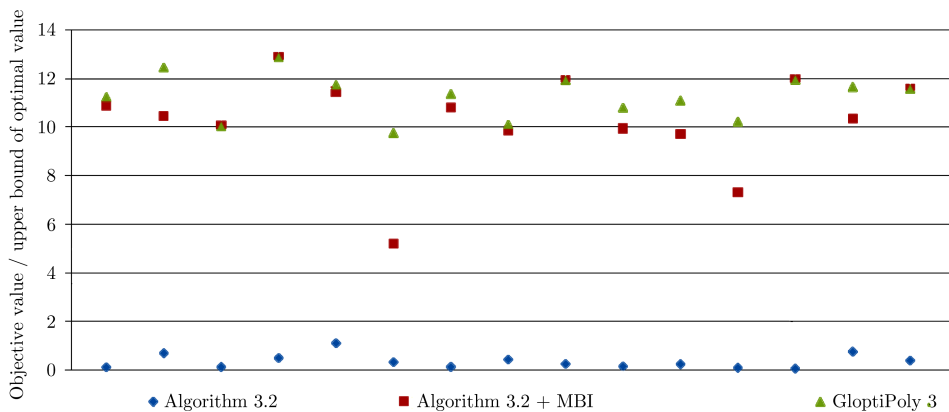


FIGURE 1. Numerical results of Problem (E) when $n = 15$

REFERENCES

- [1] N. Alon and A. Naor, *Approximating the cut-norm via Grothendieck's inequality*, SIAM J. Comput. **35** (2006), no. 4, 787–803 (electronic), DOI 10.1137/S0097539704441629. MR2203567 (2006k:68176)
- [2] G.M. de Athayde and R.G. Flôres, Jr., *Incorporating skewness and kurtosis in portfolio optimization: A multidimensional efficient set*, in S. Satchell and A. Scowcroft (Eds.), *Advances in Portfolio Construction and Implementation*, Butterworth-Heinemann, Oxford, UK, Chapter 10, 243–257, 2003.
- [3] A. Barmapoutis, B. Jian, B. C. Vemuri, and T. M. Shepherd, *Symmetric positive 4th order tensors & their estimation from diffusion weighted MRI*, Proceedings of the 20th International Conference on Information Processing in Medical Imaging, 308–319, 2007.
- [4] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, 2004. MR2061575 (2005d:90002)
- [5] B. Chen, S. He, Z. Li, and S. Zhang, *Maximum block improvement and polynomial optimization*, SIAM J. Optim. **22** (2012), no. 1, 87–107, DOI 10.1137/110834524. MR2902686
- [6] G. Dahl, J. M. Leinaas, J. Myrheim, and E. Ovrup, *A tensor product matrix approximation problem in quantum physics*, Linear Algebra Appl. **420** (2007), no. 2-3, 711–725, DOI 10.1016/j.laa.2006.08.026. MR2278245 (2007i:81029)

- [7] A. Ghosh, E. Tsigaridas, M. Descoteaux, P. Comon, B. Mourrain, and R. Deriche, *A Polynomial based approach to extract the maxima of an antipodally symmetric spherical function and its application to extract fiber directions from the orientation distribution function in diffusion MRI*, Proceedings of the 11th International Conference on Medical Image Computing and Computer Assisted Intervention, 237–248, 2008.
- [8] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. Assoc. Comput. Mach. **42** (1995), no. 6, 1115–1145, DOI 10.1145/227683.227684. MR1412228 (97g:90108)
- [9] L. Gurvits, *Classical deterministic complexity of Edmond’s problem and quantum entanglement*, Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, ACM, New York, 2003, pp. 10–19 (electronic), DOI 10.1145/780542.780545. MR2121068 (2005k:68092)
- [10] S. He, Z. Li, and S. Zhang, *Approximation algorithms for homogeneous polynomial optimization with quadratic constraints*, Math. Program. **125** (2010), no. 2, Ser. B, 353–383, DOI 10.1007/s10107-010-0409-z. MR2733568 (2011j:90138)
- [11] S. He, Z. Li, and S. Zhang, *Approximation algorithms for discrete polynomial optimization*, Journal of the Operations Research Society of China, 1, 3–36, 2013.
- [12] S. He, Z.-Q. Luo, J. Nie, and S. Zhang, *Semidefinite relaxation bounds for indefinite homogeneous quadratic optimization*, SIAM J. Optim. **19** (2008), no. 2, 503–523, DOI 10.1137/070679041. MR2425027 (2009h:90046)
- [13] D. Henrion, J.-B. Lasserre, and J. Löfberg, *GloptiPoly 3: moments, optimization and semidefinite programming*, Optim. Methods Softw. **24** (2009), no. 4-5, 761–779, DOI 10.1080/10556780802699201. MR2554910 (2010i:90086)
- [14] E. de Klerk and M. Laurent, *Error bounds for some semidefinite programming approaches to polynomial minimization on the hypercube*, SIAM J. Optim. **20** (2010), no. 6, 3104–3120, DOI 10.1137/100790835. MR2735946 (2011m:90123)
- [15] S. Khot and A. Naor, *Linear equations modulo 2 and the L_1 diameter of convex bodies*, Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, 318–328, 2007.
- [16] P.-M. Kleniati, P. Parpas, and B. Rustem, *Partitioning procedure for polynomial optimization*, J. Global Optim. **48** (2010), no. 4, 549–567, DOI 10.1007/s10898-010-9529-5. MR2735297 (2012c:90106)
- [17] E. Kofidis and P. Regalia, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl. **23** (2001/02), no. 3, 863–884 (electronic), DOI 10.1137/S0895479801387413. MR1896822 (2003a:15029)
- [18] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev. **51** (2009), no. 3, 455–500, DOI 10.1137/07070111X. MR2535056 (2010j:15027)
- [19] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim. **11** (2000/01), no. 3, 796–817, DOI 10.1137/S1052623400366802. MR1814045 (2002b:90054)
- [20] J. B. Lasserre, *Polynomials nonnegative on a grid and discrete optimization*, Trans. Amer. Math. Soc. **354** (2002), no. 2, 631–649, DOI 10.1090/S0002-9947-01-02898-7. MR1862561 (2002g:90086)
- [21] M. Laurent, *Sums of squares, moment matrices and optimization over polynomials*, Emerging Applications of Algebraic Geometry, IMA Vol. Math. Appl., vol. 149, Springer, New York, 2009, pp. 157–270, DOI 10.1007/978-0-387-09686-5_7. MR2500468 (2010j:13054)
- [22] Z. Li, *Polynomial Optimization Problems—Approximation Algorithms and Applications*, Ph.D. Thesis, The Chinese University of Hong Kong, Shatin, Hong Kong, 2011.
- [23] C. Ling, J. Nie, L. Qi, and Y. Ye, *Biquadratic optimization over unit spheres and semidefinite programming relaxations*, SIAM J. Optim. **20** (2009), no. 3, 1286–1310, DOI 10.1137/080729104. MR2546345 (2010h:90087)
- [24] Z.-Q. Luo, N. D. Sidiropoulos, P. Tseng, and S. Zhang, *Approximation bounds for quadratic optimization with homogeneous quadratic constraints*, SIAM J. Optim. **18** (2007), no. 1, 1–28 (electronic), DOI 10.1137/050642691. MR2299671 (2008b:90074)
- [25] Z.-Q. Luo and S. Zhang, *A semidefinite relaxation scheme for multivariate quartic polynomial optimization with quadratic constraints*, SIAM J. Optim. **20** (2010), no. 4, 1716–1736, DOI 10.1137/090772952. MR2600236 (2011g:90147)

- [26] B. Marić, Z.-Q. Luo, and T. N. Davidson, *Blind constant modulus equalization via convex optimization*, IEEE Trans. Signal Process. **51** (2003), no. 3, 805–818, DOI 10.1109/TSP.2002.808112. MR1963878 (2004d:94004)
- [27] D. Maringer and P. Parpas, *Global optimization of higher order moments in portfolio selection*, J. Global Optim. **43** (2009), no. 2-3, 219–230, DOI 10.1007/s10898-007-9224-3. MR2471883 (2009k:91085)
- [28] A. Nemirovski, *Lectures on Modern Convex Optimization*, The H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 2005.
- [29] A. Nemirovski, C. Roos, and T. Terlaky, *On maximization of quadratic form over intersection of ellipsoids with common center*, Math. Program. **86** (1999), no. 3, Ser. A, 463–473, DOI 10.1007/s101070050100. MR1733748 (2000m:90025)
- [30] Yu. Nesterov, *Semidefinite relaxation and nonconvex quadratic optimization*, Optim. Methods Softw. **9** (1998), no. 1-3, 141–160, DOI 10.1080/10556789808805690. MR1618100 (99a:90144)
- [31] Yu. Nesterov, *Random Walk in a Simplex and Quadratic Optimization over Convex Polytopes*, CORE Discussion Paper 2003/71, Université catholique de Louvain, Louvain-la-Neuve, Belgium, 2003.
- [32] Q. Ni, L. Qi, and F. Wang, *An eigenvalue method for testing positive definiteness of a multivariate form*, IEEE Trans. Automat. Control **53** (2008), no. 5, 1096–1107, DOI 10.1109/TAC.2008.923679. MR2445667 (2009g:93110)
- [33] J. Nie, *An approximation bound analysis for Lasserre’s relaxation in multivariate polynomial optimization*, Journal of the Operations Research Society of China **1** (2013), 313–332.
- [34] P. A. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. Dissertation, California Institute of Technology, Pasadena, CA, 2000.
- [35] P. A. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Math. Program. **96** (2003), no. 2, Ser. B, 293–320, DOI 10.1007/s10107-003-0387-5. Algebraic and Geometric Methods in Discrete Optimization. MR1993050 (2004g:90075)
- [36] A. J. Prakash, C. H. Chang, and T. E. Pactwa, *Selecting a portfolio with skewness: Recent evidence from US, European, and Latin American equity markets*, Journal of Banking & Finance, **27**, 1375–1390, 2003.
- [37] L. Qi, *Extrema of a real polynomial*, J. Global Optim. **30** (2004), no. 4, 405–433, DOI 10.1007/s10898-004-6875-1. MR2115027 (2006a:90114)
- [38] L. Qi, *Eigenvalues and invariants of tensors*, J. Math. Anal. Appl. **325** (2007), no. 2, 1363–1377, DOI 10.1016/j.jmaa.2006.02.071. MR2270090 (2007k:15046)
- [39] L. Qi and K. L. Teo, *Multivariate polynomial minimization and its application in signal processing*, J. Global Optim. **26** (2003), no. 4, 419–433, DOI 10.1023/A:1024778309049. MR1989748 (2004e:90131)
- [40] L. Qi, Z. Wan, and Y.-F. Yang, *Global minimization of normal quartic polynomials based on global descent directions*, SIAM J. Optim. **15** (2004), no. 1, 275–302 (electronic), DOI 10.1137/S1052623403420857. MR2112986 (2005j:90080)
- [41] Anthony M.-C. So, *Deterministic approximation algorithms for sphere constrained homogeneous polynomial optimization problems*, Math. Program. **129** (2011), no. 2, Ser. B, 357–382, DOI 10.1007/s10107-011-0464-0. MR2837886 (2012h:52024)
- [42] J. F. Sturm and S. Zhang, *On cones of nonnegative quadratic functions*, Math. Oper. Res. **28** (2003), no. 2, 246–267, DOI 10.1287/moor.28.2.246.14485. MR1980662 (2004c:90035)
- [43] W. Sun and Y.-X. Yuan, *Optimization Theory and Methods*, Springer Optimization and Its Applications, vol. 1, Springer, New York, 2006. Nonlinear programming. MR2232297 (2007c:90002)
- [44] Y. Ye, *Approximating quadratic programming with bound and quadratic constraints*, Math. Program. **84** (1999), no. 2, Ser. A, 219–226. MR1690021 (2000e:90041)
- [45] Y. Ye, *Approximating global quadratic optimization with convex quadratic constraints*, J. Global Optim. **15** (1999), no. 1, 1–17, DOI 10.1023/A:1008370723217. MR1706356 (2000f:90076)
- [46] Y.-X. Yuan, *A review of trust region algorithms for optimization*, ICIAM 99 (Edinburgh), Oxford Univ. Press, Oxford, 2000, pp. 271–282. MR1824450 (2002a:90076)
- [47] S. Zhang, *Quadratic maximization and semidefinite relaxation*, Math. Program. **87** (2000), no. 3, Ser. A, 453–465, DOI 10.1007/s101070050006. MR1757558 (2001i:90059)

- [48] S. Zhang and Y. Huang, *Complex quadratic optimization and semidefinite programming*, SIAM J. Optim. **16** (2006), no. 3, 871–890, DOI 10.1137/04061341X. MR2197560 (2006i:90047)
- [49] X. Zhang, L. Qi, and Y. Ye, *The cubic spherical optimization problems*, Math. Comp. **81** (2012), no. 279, 1513–1525, DOI 10.1090/S0025-5718-2012-02577-4. MR2904588

DEPARTMENT OF MANAGEMENT SCIENCES, CITY UNIVERSITY OF HONG KONG, HONG KONG
E-mail address: `simaihe@cityu.edu.hk`

(CORRESPONDING AUTHOR) DEPARTMENT OF MATHEMATICS, UNIVERSITY OF PORTSMOUTH,
PORTSMOUTH PO1 3HF, UNITED KINGDOM
E-mail address: `zheningli@gmail.com`

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING, UNIVERSITY OF MINNESOTA, MINNEAPOLIS, MINNESOTA 55455
E-mail address: `zhangs@umn.edu`