Maximum Block Improvement and Polynomial Optimization

Bilian CHEN [*]   Simai HE [†]   Zhening LI [‡]   Shuzhong ZHANG [§]

October 7, 2011

Abstract

In this paper we propose an efficient method for solving the spherically constrained homogeneous polynomial optimization problem. The new approach has the following three main ingredients. First, we establish a block coordinate descent type search method for nonlinear optimization, with the novelty being that we only accept a block update that achieves the maximum improvement, hence the name of our new search method: *Maximum Block Improvement* (MBI). Convergence of the sequence produced by the MBI method to a stationary point is proven. Second, we establish that maximizing a homogeneous polynomial over a sphere is equivalent to its tensor relaxation problem, thus we can maximize a homogeneous polynomial function over a sphere by its tensor relaxation via the MBI approach. Third, we propose a scheme to reach a KKT point of the polynomial optimization, provided that a stationary solution for the relaxed tensor problem is available. Numerical experiments have shown that our new method works very efficiently: for a majority of the test instances that we have experimented with, the method finds the global optimal solution at a low computational cost.

Keywords: block coordinate descent, polynomial optimization problem, tensor form.

Mathematics Subject Classification: 90C26, 90C30, 15A69, 49M27.

---

[*]Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong. Email: blchen@se.cuhk.edu.hk.

[†]Department of Management Sciences, City University of Hong Kong, Kowloon Tong, Hong Kong. Email: simaihe@cityu.edu.hk. This author's work is supported in part by Hong Kong GRF Grant CityU143711.

[‡]Department of Mathematics, Shanghai University, Shanghai 200444, China. Email: zheningli@gmail.com. This author's work is supported in part by the Shanghai Leading Academic Discipline Project under Grant S30104.

[§]Industrial and Systems Engineering Program, University of Minnesota, Minneapolis, MN 55455, USA. Email: zhangs@umn.edu. (On leave from Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong.) This author's work is supported in part by Hong Kong GRF Grant CUHK419409.

1   Introduction

The optimization models whose objective and constraints are polynomial functions have recently attracted much research attention. This is in part due to an increased demand on the application side (cf. the sample applications in numerical linear algebra [50, 26, 28], material sciences [57], quantum physics [9, 18], and signal processing [16, 4, 53]), and in part due to its own strong theoretical appeal. Indeed, polynomial optimization is a challenging task; at the same time it is rich enough to be fruitful. For instance, even the simplest instances of polynomial optimization, such as maximizing a cubic polynomial over a sphere, is NP-hard (Nesterov [42]). However, the problem is so elementary that it can be even attempted in an undergraduate calculus class. For readers interested in polynomial optimization with simple constraints, we refer to De Klerk [12] for a survey on the computational complexity of optimizing various classes of polynomial functions over a simplex, hypercube or sphere. In particular, De Klerk et al. [13] designed a polynomial-time approximation scheme (PTAS) for minimizing polynomials of fixed degree over the simplex.

So far, a few results have been obtained for approximation algorithms with guaranteed worst-case performance ratios for higher degree polynomial optimization problems. Luo and Zhang [39] derived a polynomial-time approximation algorithm to optimize a multivariate quartic polynomial over a region defined by quadratic inequalities. Ling et al. [37] considered the problem of minimizing a biquadratic function over two spheres, and proposed polynomial-time approximation algorithms. He et al. [21] discussed the optimization of homogeneous polynomial functions of any fixed degree over quadratic constraints, and proposed approximation algorithms, with performance ratios improving that of [39, 37]. Recently, So [56] improved the approximation ratio in the case of spherically constrained homogeneous polynomial optimizations. For a general inhomogeneous polynomial optimization over convex compact sets, He et al. [22] proposed polynomial-time approximation algorithms with relative approximation ratios, which is the only result so far with regard to approximation algorithms for an inhomogeneous polynomial. Later, the authors extended their results in [23] by considering polynomials in discrete (typically binary) variables, and designed randomized approximation algorithms. For a recent treatise on the topic, one may refer to the Ph.D. thesis of Li [36].

On the computational side, polynomial optimization problems can be treated as nonlinear programming, and many existing software packages are available, including KNITRO, BARON, MINOS, SNOPT, and Matlab optimization toolbox. (We refer the interested reader to NEOS server [44] for further information.) However, these solvers are not tailor-made for polynomial optimization problems, and so the performance may vary greatly from one problem instance to another. Therefore, it is natural to wonder whether one can design efficient algorithms for specific types of polynomial optimization problems. Prajna et al. [49] presented a package named SOSTOOLS for solving sum of squares polynomial programs, based on the sum of squares (SOS) decomposition for multivariate polynomials, which can be computed using (possibly large size) semidefinite programs. More recently, Henrion et al. [25] developed a specialized tool known as GloptiPoly 3 (its former version GloptiPoly; see Henrion and Lasserre [24]) in finding global optimal solutions for polynomial optimizations based on the SOS approach (see [32, 33, 35, 47, 48] for details). GloptiPoly 3 calls the semidefinite programming (SDP) solver SeDuMi [58]. Therefore, due to the limitation to solve large SDP problems, GloptiPoly 3 may not be the right tool to deal with large size polynomials (say a 6th degree polynomial in 20 variables). However, if the polynomial optimization model in question is sparse in some way, then it is possible to exploit the sparsity in GloptiPoly 3; see [34]. As a matter of fact, SparsePOP [61] makes use of the sparsity explicitly, and is a more appropriate alternative for sparse polynomial optimization based on the SOS approach. Unfortunately, for the problems considered in this paper, the sparsity structure is not readily exploited by SparsePOP. For

more information on polynomial optimization, we refer to the recent book of Anjos and Lasserre [2], and the references therein.

Spherically constrained homogeneous polynomial optimization models have received some recent research attention, theoretically as well as numerically. One direct approach is to apply the method of Lagrange multipliers to reach a set of multivariate polynomial equations, namely the Karush-Kuhn-Tucker (KKT) system, which provides the necessary conditions for optimality; see, e.g., [16, 28, 65]. In [16], one strives to enumerate *all* the solutions of a KKT system, not only the global optimum, as all the KKT solutions will be meaningful in this application. Indeed, the authors develop special algorithms for that purpose: e.g., the subdivision methods proposed by Mourrain and Pavone [40], and the generalized normal forms algorithms designed by Mourrain and Trébuchet [41]. However, the shortcomings of these methods are apparent if the degree of the polynomial is high. An important application of spherically constrained homogeneous polynomial optimizations is the best (in the sense of least-square) rank-one approximation of tensors (sometimes also known as the rank-one factorization): find vectors $x^1, x^2, \cdots, x^d$ for the following minimization problem

$$(R_{fac}) \quad \min \quad \sum_{i_1, i_2, \cdots, i_d} \left(x_{i_1}^1 x_{i_2}^2 \cdots x_{i_d}^d - \mathcal{F}_{i_1 i_2 \cdots i_d} \right)^2,$$

where $\mathcal{F} = (\mathcal{F}_{i_1 i_2 \cdots i_d})$ is a d-th order tensor. In particular, if the tensor $\mathcal{F}$ is super-symmetric (every element $\mathcal{F}_{i_1 i_2 \cdots i_d}$ is invariant under all permutations of $(i_1, i_2, \cdots, i_d)$), then the optimal vectors $x^1, x^2, \cdots, x^d$ should coincide (namely they should be equal to each other). The main workhorse for solving the above tensor problem is known as the *Alternating Least Square* (ALS) method proposed originally by Carroll and Chang [8] and Harshman [19]. However, the ALS method is not guaranteed to converge to a global minimum or a stationary point, only to a solution where the objective function ceases to decrease. There are numerous extensions of the ALS method (e.g., incorporating a line-search procedure in the ALS procedure [46, 55]). Along a related line, De Lathauwer et al. [14] proposed higher-order power method (HOPM) on rank-one approximation of higher-order tensors, which can also be viewed as an extension of the ALS method. Following up on that approach, Kofidis and Regalia [30] devised symmetric higher-order power method (S-HOPM) to rank-one approximation of super-symmetric tensors, and proved its convergence for super-symmetric tensors whenever their corresponding polynomial forms have convexity or concavity. Furthermore, Wang and Qi [62] proposed a greedy method, which iteratively computes the best super-symmetric rank-one approximation of the residual tensors in order to obtain a successive super-symmetric rank-one decomposition. Those methods have nice properties; however they all fail to guarantee convergence for the tensor model (R_{fac}), whether the tensor is super-symmetric or not. For an overview on the recent developments on tensor decomposition, we refer to the excellent survey by Kolda and Bader [31]. Another entirely different but very interesting approach, known as the Z-eigenvalue method, was proposed by Qi et al. [54]. This heuristic cross-hill Z-eigenvalue method aims to solve homogeneous polynomial functions with degree at most three.

Motivated by the Block Coordinate Descent (BCD) method for non-differentiable minimization proposed by Tseng [60], and the Iterative Waterfilling Algorithm (IWA) for multiuser power control in digital subscriber lines by Luo and Pang [38], in this paper we shall propose a different method, to be called the *Maximum Block Improvement* (MBI), for solving spherically constrained homogeneous polynomial optimization problems. Our new method, MBI, guarantees convergence to a stationary point of the problem, which is typically also global optimal in our numerical experiences. The method actually has a general appeal: it can be applied to solve any optimization model with separate block constraints. The proposed MBI approach can naturally be regarded as

a local improvement scheme for polynomial optimization, to start from any good initial solutions. Therefore, the new MBI method can be used in combination with any approximation algorithms (such as Khot and Naor [29] and He et al. [21, 22, 23]), to achieve excellent performance in practice while enjoying the theoretical worst case performance guarantees.

The remainder of the paper is organized as follows. In Section 2, we introduce the notations and the models to be discussed throughout the paper. Then in Section 3, the general scheme of Maximum Block Improvement method will be introduced, and convergence properties will be discussed in the same section. In Section 4, we present an equivalence result between the spherically constrained homogeneous polynomial optimization and its tensor relaxation problem. This will enable the application of the MBI method to solve the polynomial optimization model. Finally, we present the results of our numerical experiments in Section 5.

2   Notations and Models

Let us consider the following multilinear tensor function

$$F(x^1, x^2, \cdots, x^d) = \sum_{1 \le i_1 \le n_1, 1 \le i_2 \le n_2, \cdots, 1 \le i_d \le n_d} \mathcal{F}_{i_1 i_2 \cdots i_d} x^1_{i_1} x^2_{i_2} \cdots x^d_{i_d}, \tag{1}$$

where $x^k \in \Re^{n_k}$ for $k = 1, 2, \ldots, d$, and $\mathcal{F} = (\mathcal{F}_{i_1 i_2 \cdots i_d}) \in \Re^{n_1 \times n_2 \times \cdots \times n_d}$ is a d-th order tensor with F being its associated multilinear function. Closely related to the tensor form $\mathcal{F}$ is a general d-th degree homogeneous polynomial function $f(x)$, where $x \in \Re^n$, with its associated tensor $\mathcal{F}$ being super-symmetric. In fact, super-symmetric tensors are bijectively related to homogeneous polynomials; see [31]. Denote F to be the multilinear function defined by the super-symmetric tensor form $\mathcal{F}$, we then have

$$f(x) = F(\underbrace{x, x, \cdots, x}_{d}). \tag{2}$$

A generic multivariate (inhomogeneous) polynomial function of degree d, denoted by $p(x)$, can be explicitly written as a summation of homogeneous polynomial functions in decreasing degrees, namely

$$p(x) := \sum_{i=1}^{d} f_i(x) + f_0 = \sum_{i=1}^{d} F_i(\underbrace{x, x, \cdots, x}_{i}) + f_0, \tag{3}$$

where $x \in \Re^n, f_0 \in \Re$, and $f_i(x) = F_i(\underbrace{x, x, \cdots, x}_{i})$ is a homogeneous polynomial function of degree i for $i = 1, 2, ..., d$.

Throughout this paper, we use F to denote a multilinear function defined by a tensor form $\mathcal{F}$, f for a homogeneous polynomial function, and p for an inhomogeneous polynomial function; see functions (1), (2) and (3). To avoid triviality, we also assume that at least one component of the tensor form, $\mathcal{F}$ in functions F and f, and $\mathcal{F}_d$ in function p is nonzero. Throughout the paper we shall use the 2-norm for vectors, matrices and tensors, which is the usual Euclidean norm, defined as

$$\|\mathcal{F}\| := \sqrt{\sum_{1 \le i_1 \le n_1, 1 \le i_2 \le n_2, \cdots, 1 \le i_d \le n_d} \mathcal{F}_{i_1 i_2 \cdots i_d}^2}.$$

The symbol '$\otimes$' represents the vector outer product. For example, for vectors $x \in \Re^{n_1}, y \in \Re^{n_2}, z \in \Re^{n_3}$, the notion $x \otimes y \otimes z$ defines a third order tensor $\mathcal{F} \in \Re^{n_1 \times n_2 \times n_3}$, whose (i, j, k)-th element $\mathcal{F}_{ijk}$ is equal to $x_i y_j z_k$.

4

In this paper we shall focus on the polynomial optimization models as considered in He et al. [21, 22], such as optimization of multilinear tensor function (1), homogeneous polynomial (2) and general inhomogeneous polynomial (3) over quadratic constraints, including spherical constraint or the Euclidean ball constraint as a special case. In particular, the authors considered a general model where an inhomogeneous polynomial function is maximized over the intersection of co-centered ellipsoids:

$$(Q) \quad \max \quad p(x)$$
$$\text{s.t.} \quad x^{\mathrm{T}} Q_j x \leq 1, \ j = 1, 2, ..., m,$$
$$x \in \Re^n,$$

where matrices $Q_j \succeq 0$ for $j = 1, 2, ..., m$, and $\sum_{j=1}^{m} Q_j \succ 0$.

In this paper we shall pay special attention to the following model

$$(H) \quad \max \quad f(x)$$
$$\text{s.t.} \quad \|x\| = 1, \ x \in \Re^n.$$

As we shall see later, (H) has applications in magnetic resonance imaging (MRI), the best rank-one approximation of the super-symmetric tensor $\mathcal{F}$, and the problem of finding the largest eigenvalue of the tensor $\mathcal{F}$; see e.g., [16, 30, 50, 52]. The multilinear tensor relaxation of (H) is

$$(T) \quad \max \quad F(x^1, x^2, \cdots, x^d)$$
$$\text{s.t.} \quad \|x^i\| = 1, \ x^i \in \Re^n, \ i = 1, 2, ..., d.$$

In He et al. [21], the above relaxation has played a crucial role in the approximation algorithms for solving (H). One of the main contributions of the current paper is to reveal an intrinsic relationship between the optimal solutions of (T) and (H). Finally, as a matter of notation, for a given optimization problem (P) we shall denote its optimal value by $v(P)$.

3   The Maximum Block Improvement (MBI) Method

Towards eventually solving (T), let us start by considering a generic optimization model in the form of

$$(G) \quad \max \quad f(x^1, x^2, \cdots, x^d)$$
$$\text{s.t.} \quad x^i \in S^i \subseteq \Re^{n_i}, \ i = 1, 2, ..., d,$$

where $f : \Re^{n_1 + n_2 + \cdots + n_d} \to \Re$ is a general continuous function, and S^i is a general set, $i = 1, 2, ..., d$. A popular special case of the model is where $S^i = \Re^{n_i}$, $i = 1, 2, ..., d$. For that special case, a method known as the *Block Coordinate Descent* (BCD) is well studied; see Tseng [60] and the references therein. The method calls for maximizing one block, say $x^i \in \Re^{n_i}$, at one time, while all other variables in other blocks are temporarily fixed. One then moves on to alter the choice of the blocks. Very recently, Wright [63] introduced an extension based on BCD. Typically, under various convexity assumptions on the objective function, one is able to show some convergence property of the BCD method (cf. [60]). In fact, the BCD method can be applied regardless of any convexity assumptions, as long as one is able to optimize over one block of variable while fixing the others. A summary of the BCD or other block search methods can be found in Bertsekas [5]. The approach has a relatively long history (sometimes it is also known as the block nonlinear Gauss-Seidel method). Without taking any precaution, the BCD method may not be convergent; see the examples in Powell [51]. In the literature, this issue of convergence has been thoroughly studied. However, the results were not entirely satisfactory. To ensure convergence, one would either require

some type of convexity (cf. the discussion in [5]), or the search routine is modified (cf. a proximal-point modification in Grippo and Sciandrone [11]). Our new block search method does not modify the objective function in the block-search subroutine, and at the same time ensures the convergence to a stationary solution within the structure of the BCD framework. The so-called ALS method for tensor decomposition problems (see Section 1) is a special form of the BCD method. We shall remark that the model is reminiscent of a noncooperative game, where S^i can be regarded as the strategy set of Player i, $i = 1, 2, ..., d$. Certainly, in the case of noncooperative game, the objective of each player may be different in general. In a channel spectrum allocation game in communication, the corresponding BCD approach is also known as the *Iterative Waterfilling Algorithm* (IWA); Luo and Pang [38] show that the IWA is convergent to the Nash equilibrium under some fairly loose conditions. It is possible that the IWA may cycle in the absence of these conditions; see an example in [20].

To simplify the analysis, we assume here that S^i is compact, $i = 1, 2, ..., d$. But that alone is insufficient to guarantee the convergence, as we know that even for the special case of the ALS, the iterates may not converge to a stationary point; see e.g., [10, 14, 15, 55]. A sufficient condition for convergence is to take a step that corresponds to the *maximum* improvement. The enhanced procedure is as follows:

Algorithm MBI: The Maximum Block Improvement Method for Solving (G)

0 *(Initialization).* Choose a feasible solution $(x_0^1, x_0^2, \cdots, x_0^d)$ with $x_0^i \in S^i$ for $i = 1, 2, \ldots, d$ and compute initial objective value $v_0 := f(x_0^1, x_0^2, \cdots, x_0^d)$. Set $k := 0$.

1 *(Block Improvement).* For each $i = 1, 2, \ldots, d$, solve

$$(G_i) \quad \max \quad f(x_k^1, \cdots, x_k^{i-1}, x^i, x_k^{i+1}, \cdots, x_k^d)$$
$$\text{s.t.} \quad x^i \in S^i,$$

and let

$$y_{k+1}^i \quad := \quad \arg\max_{x^i \in S^i} f(x_k^1, \cdots, x_k^{i-1}, x^i, x_k^{i+1}, \cdots, x_k^d),$$
$$w_{k+1}^i \quad := \quad f(x_k^1, \cdots, x_k^{i-1}, y_{k+1}^i, x_k^{i+1}, \cdots, x_k^d).$$

2 *(Maximum Improvement).* Let $w_{k+1} := \max_{1 \le i \le d} w_{k+1}^i$ and $i^* = \arg\max_{1 \le i \le d} w_{k+1}^i$. Let

$$x_{k+1}^i \quad := \quad x_k^i, \ \forall \ i \in \{1, 2, \cdots, d\}\backslash\{i^*\},$$
$$x_{k+1}^{i^*} \quad := \quad y_{k+1}^{i^*},$$
$$v_{k+1} \quad := \quad w_{k+1}.$$

3 *(Stopping Criterion).* If $|v_{k+1} - v_k| < \epsilon$, stop. Otherwise, set $k := k + 1$, and go to Step 1.

The key assumption in the above process is that problem (G_i) can be easily solved, which is the case for many applications. For instance, when $f(x^1, x^2, \cdots, x^d) = -\|\mathcal{F} - x^1 \otimes x^2 \otimes \cdots \otimes x^d\|^2$, and $S^i = \Re^{n_i}$, then (G_i) is simply a least square problem; when $f(x^1, x^2, \cdots, x^d)$ is a multilinear tensor form, and S^i is convex, then (G_i) is a convex optimization problem. We shall remark here that a major difference between MBI and IWA (or, for that matter, ALS, BCD or block nonlinear Gauss-Seidel) lies in Step 2: rather than improving among block decision variables alternatively or

cyclically, MBI chooses to update the block variables that achieve the maximum improvement. If in Step 2, solving (G_i) is a least square problem, then MBI becomes a variant of the *Alternating Least Square* (ALS) method, which is widely used in tensor decompositions (cf. [31]). Unlike the ALS method, as we show next, the MBI method guarantees to converge to a stationary point.

Theorem 3.1 *If S^i is compact for $i = 1, 2, ..., d$, then any cluster point of the iterates $(x_k^1, x_k^2, \cdots, x_k^d)$, say $(x_*^1, x_*^2, \cdots, x_*^d)$, will be a stationary point for (G); i.e.,*

$$x_*^i = \arg \max_{x^i \in S^i} f(x_*^1, \cdots, x_*^{i-1}, x^i, x_*^{i+1}, \cdots, x_*^d), \ \forall \, i = 1, 2, ..., d.$$

Proof. For each fixed $(x^1, \cdots, x^{i-1}, x^{i+1}, \cdots, x^d)$, denote $R_i(x^1, \cdots, x^{i-1}, x^{i+1}, \cdots, x^d)$ to be a best response function to x^i, namely

$$R_i(x^1, \cdots, x^{i-1}, x^{i+1}, \cdots, x^d) \in \arg \max_{x^i \in S^i} f(x^1, \cdots, x^{i-1}, x^i, x^{i+1}, \cdots, x^d).$$

Suppose that $(x_{k_t}^1, x_{k_t}^2, \cdots, x_{k_t}^d) \to (x_*^1, x_*^2, \cdots, x_*^d)$ as $t \to \infty$. Then, for any $1 \le i \le d$, we have

$$
\begin{aligned}
& f(x_{k_t}^1, \cdots, x_{k_t}^{i-1}, R_i(x_*^1, \cdots, x_*^{i-1}, x_*^{i+1}, \cdots, x_*^d), x_{k_t}^{i+1}, \cdots, x_{k_t}^d) \\
\le \ & f(x_{k_t}^1, \cdots, x_{k_t}^{i-1}, R_i(x_{k_t}^1, \cdots, x_{k_t}^{i-1}, x_{k_t}^{i+1}, \cdots, x_{k_t}^d), x_{k_t}^{i+1}, \cdots, x_{k_t}^d) \\
\le \ & f(x_{k_t+1}^1, x_{k_t+1}^2, \cdots, x_{k_t+1}^d) \\
\le \ & f(x_{k_{t+1}}^1, x_{k_{t+1}}^2, \cdots, x_{k_{t+1}}^d).
\end{aligned}
$$

By continuity, when $t \to \infty$, it follows that

$$f(x_*^1, \cdots, x_*^{i-1}, R_i(x_*^1, \cdots, x_*^{i-1}, x_*^{i+1}, \cdots, x_*^d), x_*^{i+1}, \cdots, x_*^d) \le f(x_*^1, x_*^2, \cdots, x_*^d),$$

which implies that the above should hold as an equality, since the other inequality is true by the definition of the best response function R_i. Thus, x_*^i is the best response to $(x_*^1, \cdots, x_*^{i-1}, x_*^{i+1}, \cdots, x_*^d)$, or equivalently, x_*^i is the optimal solution for the problem

$$\max_{x^i \in S^i} \ f(x_*^1, \cdots, x_*^{i-1}, x^i, x_*^{i+1}, \cdots, x_*^d),$$

for all $i = 1, 2, ..., d$. $\qquad\square$

In many applications, S^i is described by inequalities and equalities; e.g.,

$$S^i = \{x^i \in \Re^{n_i} \mid g_j^i(x^i) \le 0, \ j = 1, 2, \ldots, m_i; \ h_j^i(x^i) = 0, \ j = 1, 2, \ldots, \ell_i\},$$

where $i = 1, 2, ..., d$. It is then more convenient to use the so-called KKT conditions, instead of an abstract form of the stationarity, under some constraint qualifications (CQ).[1]

Corollary 3.2 *If $S^i = \{x^i \in \Re^{n_i} \mid g_j^i(x^i) \le 0, \ j = 1, 2, \ldots, m_i; \ h_j^i(x^i) = 0, \ j = 1, 2, \ldots, \ell_i\}$ is compact for all $i = 1, 2, ..., d$, and it satisfies a suitable constraint qualification (cf. footnote 1), then any cluster point of the iterates $(x_k^1, x_k^2, \cdots, x_k^d)$, say $(x_*^1, x_*^2, \cdots, x_*^d)$, will be a KKT point for (G).*

---

[1] The most widely used CQs include: the Slater condition; the linear independence constraint qualification (LICQ); the Mangasarian-Fromowitz constraint qualification (MFCQ); the constant rank constraint qualification (CRCQ); the constant positive linear dependence constraint qualification (CPLD); and quasi-normality constraint qualification (QNCQ). For details the reader is referred to a textbook on nonlinear programming; e.g., Bertsekas [5].

Proof. As asserted by Theorem 3.1, $(x_*^1, x_*^2, \cdots, x_*^d)$ is a stationary point. Moreover, since a constraint qualification is satisfied for S^i, we know that x_*^i is a KKT point as well. Namely, there exist u_j^i and v_j^i such that $x^i = x_*^i$ satisfies the equations:

$$
\begin{cases}
\nabla_{x^i} f(x_*^1, \cdots, x_*^{i-1}, x^i, x_*^{i+1}, \cdots, x_*^d) = \sum_{j=1}^{m_i} u_j^i \nabla g_j^i(x^i) + \sum_{j=1}^{\ell_i} v_j^i \nabla h_j^i(x^i), \\
u_j^i g_j^i(x^i) = 0, \ u_j^i \geq 0, \ j = 1, 2, \ldots, m_i, \\
x^i \in S^i,
\end{cases}
$$

where u_j^i is the Lagrangian multiplier corresponding to the inequality constraint $g_j^i(x^i) \leq 0$ for $j = 1, 2, ..., m_i$, and v_j^i is the Lagrangian multiplier corresponding to the equality constraint $h_j^i(x^i) = 0$ for $j = 1, 2, ..., \ell_i$. Therefore, $(x^1, x^2, \cdots, x^d) = (x_*^1, x_*^2, \cdots, x_*^d)$ is a solution for

$$
\begin{cases}
\nabla_{x^i} f(x^1, \cdots, x^{i-1}, x^i, x^{i+1}, \cdots, x^d) = \sum_{j=1}^{m_i} u_j^i \nabla g_j^i(x^i) + \sum_{j=1}^{\ell_i} v_j^i \nabla h_j^i(x^i), i = 1, 2, \ldots, d, \\
u_j^i g_j^i(x^i) = 0, \ u_j^i \geq 0, \ j = 1, 2, \ldots, m_i, \ i = 1, 2, \ldots, d, \\
x^i \in S^i, \ i = 1, 2, \ldots, d,
\end{cases}
$$

which is exactly the KKT system for (G). Therefore, $(x_*^1, x_*^2, \cdots, x_*^d)$ must be a KKT point of (G) as well. $\square$

Remark that since not all KKT points are stationary, Theorem 3.1 is in fact a stronger statement; however, Corollary 3.2 is convenient to use in many applications.

4 Spherically Constrained Homogeneous Polynomial Optimization

Our study of (G) in this paper is motivated by the tensor optimization model (T) considered in Section 2:

$$
(T) \quad \max \quad \sum_{1 \leq i_1 \leq n_1, 1 \leq i_2 \leq n_2, \cdots, 1 \leq i_d \leq n_d} \mathcal{F}_{i_1 i_2 \cdots i_d} x_{i_1}^1 x_{i_2}^2 \cdots x_{i_d}^d
$$
$$
\text{s.t.} \quad \|x^i\| = 1, \ x^i \in \Re^{n_i}, \ i = 1, 2, \ldots, d,
$$

which is clearly a special case of (G). Moreover, Algorithm MBI is simple to implement in this case, as optimizing one block while fixing all other blocks is a trivial problem to solve. In fact, simultaneously optimizing over *two* vectors of variables, while fixing other vectors, are also easy to implement; see [59, 64]. In particular, if d is even, then we may partition the blocks as $\{x^1, x^2\}, \cdots, \{x^{d-1}, x^d\}$, and then the subroutine reduces to an eigenvalue problem, rather than least square. (Some numerical results for the latter implementation will be presented in Section 5.) The flexibility in the design of the blocks is an important factor to consider in order for the MBI method to achieve its full efficiency.

It is not hard to verify (see e.g., Section 3.4.2 of [36]) that (T) is actually equivalent to the so-called best rank-one tensor approximation problem given as

$$
\min \quad \|\mathcal{F} - \lambda \cdot x^1 \otimes x^2 \otimes \cdots \otimes x^d\|
$$
$$
\text{s.t.} \quad \lambda \in \Re, \|x^i\| = 1, \ x^i \in \Re^{n_i}, \ i = 1, 2, \ldots, d.
$$

Traditionally, the ALS method is a popular solution method for such models (see [30, 14]). However, the convergence of the ALS method is not guaranteed in general, as we remarked before, and the new MBI method avoids the pitfalls regarding the convergence.

In the case when the given d-th order tensor $\mathcal{F} \in \Re^{n^d}$ is super-symmetric, then the corresponding super-symmetric rank-one approximation should be

$$\min \quad \left\| \mathcal{F} - \lambda \cdot \underbrace{x \otimes x \otimes \cdots \otimes x}_{d} \right\|$$
$$\text{s.t.} \quad \lambda \in \Re, \, x \in \Re^n.$$

Similar to the non-symmetric case, by imposing the vector x on the unit sphere, we can also verify that the rank-one approximation of super-symmetric tensor problem is indeed equivalent to

$$(H) \quad \max \quad f(x) = F(\underbrace{x, x, \cdots, x}_{d})$$
$$\text{s.t.} \quad \|x\| = 1, \, x \in \Re^n,$$

where F is the multilinear tensor function defined by the super-symmetric tensor form $\mathcal{F}$. In fact, the above problem (H) is also directly related to computing the maximal eigenvalue of the tensor $\mathcal{F}$; see Qi [50, 52]. The main contribution of this section is to present a new procedure, based on MBI, to effectively compute a KKT point for the best rank-one approximation of a super-symmetric tensor $\mathcal{F}$, namely (H).

4.1 Relationship between (H) and (T)

In He et al. [21], problem (T) is regarded as a relaxation of (H), and an approximate solution for (T) is used to construct an approximate solution for (H). Now we shall prove that these two problems are actually *equivalent*. In fact, we shall prove the following result.

Theorem 4.1 *Suppose that $\mathcal{F} \in \Re^{n^d}$ is a d-th order super-symmetric tensor with F being its corresponding multilinear function. Let $\mathcal{G}_i \in \Re^{m^t}$ be a t-th order super-symmetric tensor, with G_i being its corresponding multilinear function, $i = 1, 2, ..., n$. Consider a mapping $g : \Re^m \mapsto \Re^n$ where the i-th component of g is given by $g_i(x) = G_i(\underbrace{x, x, \cdots, x}_{t})$, $i = 1, 2, ..., n$. If the image set*

$g(\Re^m) \subseteq \Re^n$ is a linear subspace of $\Re^n$, then

$$\max_{\|g(x)\|=1} |F(\underbrace{g(x), g(x), \cdots, g(x)}_{d})| = \max_{\|g(x^i)\|=1, \, i=1,2,...,d} |F(g(x^1), g(x^2), \cdots, g(x^d))|.$$

Proof. Denote the linear subspace $g(\Re^m)$ to be $K \subseteq \Re^n$. It is clear that the two optimization problems in Theorem 4.1 is equivalent to

$$(H_d) \quad \max \quad |F(\underbrace{y, y, \cdots, y}_{d})|$$
$$\text{s.t.} \quad \|y\| = 1, \, y \in K,$$

and

$$(T_d) \quad \max \quad |F(y^1, y^2, \cdots, y^d)|$$
$$\text{s.t.} \quad \|y^i\| = 1, \, y^i \in K, \, i = 1, 2, \ldots, d,$$

respectively. We shall aim to prove that $v(T_d) = v(H_d)$.

9

The proof is based on the induction on the order of the tensor d. It is trivially true when $d = 1$. Suppose that $v(T_d) = v(H_d)$ for d with $d \geq 1$. Then, for the case $d+1$, denote $(\hat{y}^1, \hat{y}^2, \cdots, \hat{y}^d, \hat{y}^{d+1})$ to be an optimal solution of (T_{d+1}). By induction, we have

$$v(T_{d+1}) = \max_{\|y^i\|=1, \, y^i \in K, \, i=1,2,\ldots,d} |F(y^1, y^2, \cdots, y^d, \hat{y}^{d+1})| = \max_{\|y\|=1, \, y \in K} |F(\underbrace{y, y, \cdots, y}_{d}, \hat{y}^{d+1})|. \quad (4)$$

Denote S to be the set of all optimal solutions of (T_{d+1}) with support 1 or 2, i.e., the number of distinctive vectors of $\{y^1, y^2, \cdots, y^d, y^{d+1}\}$ is less than or equal to 2. From (4), we know that S is non-empty. By the continuity of F and compactness of the feasible region of y^i for $i = 1, 2, \ldots, d$, it is not hard to verify that S is compact. Now consider the following optimization problem:

$$(A) \qquad \max_{(y,y,\cdots,y;\, z,z,\cdots,z) \in S} y^{\mathrm{T}} z.$$

If the optimal value $v(A) < 1$, then let one of its optimal solution be $(\hat{y}, \hat{y}, \cdots, \hat{y}; \hat{z}, \hat{z}, \cdots, \hat{z})$. Clearly, $\hat{y} \neq \pm\hat{z}$, because otherwise $(\hat{y}, \hat{y}, \cdots, \hat{y}) \in S$ would have $v(A) = 1$, a contradiction to $v(A) < 1$. Now denote $\hat{w} = (\hat{y} + \hat{z})/\|\hat{y} + \hat{z}\|$. Since $\hat{y}, \hat{z} \in K$, $\hat{y} \neq \pm\hat{z}$, and K is a linear subspace of $\Re^n$, we shall have $\|\hat{w}\| = 1$ and $\hat{w} \in \mathrm{span}(\hat{y}, \hat{z}) \subset K$.

Without loss of generality, we may let $F(\hat{y}, \hat{y}, \cdots, \hat{y}; \hat{z}, \hat{z}, \cdots, \hat{z}) = v(T_{d+1})$. (Otherwise use $-\mathcal{F}$ instead of $\mathcal{F}$). Since $(\hat{y}, \hat{y}, \cdots, \hat{y}; \hat{z}, \hat{z}, \cdots, \hat{z})$ is an optimal solution for (T_{d+1}) and $\mathrm{span}(\hat{y}, \hat{z}) \subset K$, it is easy to show that $(\hat{y}, \hat{y}, \cdots, \hat{y}; \hat{w}, \hat{w}; \hat{z}, \hat{z}, \cdots, \hat{z})$ (namely, replacing the middle $(\hat{y}, \hat{z})$ by $(\hat{w}, \hat{w})$) is also optimal for (T_{d+1}). Apply this replacement procedures until either $\hat{y}$ or $\hat{z}$ exhausts, while keeping the optimality for (T_{d+1}). Without loss of generality, we may come to an optimal solution in a form of $(\hat{y}, \hat{y}, \cdots, \hat{y}; \hat{w}, \hat{w}, \cdots, \hat{w}) \in S$.

Let $\cos\theta = v(A)$ for some $\theta \in (0, \pi]$. Now we shall have

$$\hat{w}^{\mathrm{T}} \hat{y} = \cos(\theta/2) > \cos\theta = \hat{y}^{\mathrm{T}} \hat{z} = v(A),$$

which contradicts the optimality of $(\hat{y}, \hat{y}, \cdots, \hat{y}; \hat{z}, \hat{z}, \cdots, \hat{z})$ for (A). Thus $v(A)$ must be 1, implying that (A) has a solution with support 1, which proves $v(H_{d+1}) = v(T_{d+1})$. $\qquad\square$

One may be led to the question: are there interesting cases where $g(\Re^m)$ is a subspace? The answer is yes, and the most obvious case is to let $t = 1$ and $g(x) = Gx$ with $G \in \Re^{n \times m}$, and then Theorem 4.1 leads us to:

Corollary 4.2 *If $\mathcal{F} \in \Re^{m^d}$ is a d-th order super-symmetric tensor with F being its corresponding multilinear function, then*

$$\max_{\|Gx\|=1} |F(\underbrace{x, x, \cdots, x}_{d})| = \max_{\|Gx^i\|=1, \, i=1,2,\ldots,d} |F(x^1, x^2, \cdots, x^d)|.$$

In our particular context, our models (H) and (T) correspond to G being identity matrix, and $m = n$. This corollary connects to the so-called "generalized multilinear version of the Cauchy-Bouniakovski-Schwarz inequality" (Hiriart-Urruty [27]), which states that

Let F be a super-symmetric multilinear tensor form of order d (≥ 3), and A be a positive semidefinite matrix. If

$$|F(x, x, \cdots, x)| \leq (x^{\mathrm{T}} A x)^{d/2} \quad \forall x \in \Re^n,$$

then

$$|F(x^1, x^2, \cdots, x^d)|^2 \leq \prod_{i=1}^{d} (x^i)^{\mathrm{T}} A x^i \quad \forall x^i \in \Re^n, \, i = 1, 2, \ldots, d.$$

10

The above inequality was shown by Lojasiewicz (see [6]), and an alternative proof can be found in Nesterov and Nemirovski [43]. Yet, it also follows from Corollary 4.2 by setting $A = G^T G$.

Another nontrivial special case when the condition holds is when $t = 2$, $n = 4$, $m \geq 3$, $x \in \mathbf{C}^m$ is in the complex-valued domain, and $G_i(x, x)$ is *block square-free*, i.e., the vector x can be partitioned into two parts, $\tilde{x}$ and $\hat{x}$, and $g_i(x) = G_i(x, x) = (\tilde{x}^H G_i \hat{x} + \hat{x}^H G_i^H \tilde{x})/2$, $i = 1, 2, 3, 4$. In that case, Ai, Huang, and Zhang [1] proved that the joint numerical range $g(\mathbf{C}^m)$ is a convex cone. Due to the block square-free property, it is also not pointed at any direction, hence is a subspace.

For our subsequent discussion, the main purpose is to solve (H) via (T), and so we shall focus on the application of Corollary 4.2. First we remark that the absolute value sign in the objective function of (T_d) can actually be removed, since its optimal value is always nonnegative. Similarly, if d is odd, then the absolute value sign in (H_d) can also be removed, due to the symmetry of the constraint set; however, for even d, this absolute value sign in (H_d) is necessary. Ni and Wang [45] proved that Corollary 4.2 holds only for a special case $d = 4$ and $n = 2$. We have showed that this property can be extended to a super-symmetric tensor for general dimensions. Interestingly, this result also implies that the best super-symmetric rank-one decomposition of a super-symmetric tensor remains optimal even among all non-symmetric rank-one tensors.

Corollary 4.2 establishes the equivalence between (H) and (T) for odd d, as we discussed before. For an even degree d, one may consider $\mathcal{H}$ as the d-th order super-symmetric tensor associated with the homogeneous polynomial $h(x) := (x^T x)^{d/2}$, and let $f(x) := f(x) + \tau h(x)$, where $\tau = \|\mathcal{F}\|$. In that case, f becomes nonnegative on the sphere, and so we can again drop the absolute value sign without affecting the optimal solutions. In both cases, solving (H) can be equivalently transformed into solving (T), where the MBI method applies.

On the other hand, Corollary 4.2 may not hold for other symmetric convex constraints, such as hypercube or simplex; see an example below for the case of a box.

Example 4.3 *Denote $\mathcal{F}$ to be a diagonal matrix* $\mathrm{Diag}\,(-1, 1)$, *and the boxed constraints are* $-e \leq x, y \leq e$ *with* $e = (1, 1)^T$. *Then* $\max |F(x, y)| = \max |-x_1 y_1 + x_2 y_2| = 2$, *while* $\max |F(x, x)| = \max |-x_1^2 + x_2^2| = 1$.

One can further generalize Corollary 4.2 to allow the following mixed homogeneous polynomial function (see e.g., [23, 36]), i.e.

$$f(x^1, x^2, \cdots, x^s) := F(\underbrace{x^1, x^1, \cdots, x^1}_{d_1}, \underbrace{x^2, x^2, \cdots, x^2}_{d_2}, \cdots, \underbrace{x^s, x^s, \cdots, x^s}_{d_s}),$$

where $x^k \in \mathfrak{R}^{n_k}$ for $k = 1, 2, \ldots, s$, and the tensor form $F \in \mathfrak{R}^{n_1^{d_1} \times n_2^{d_2} \times \cdots \times n_s^{d_s}}$ has partial symmetric property, namely for any fixed $(x^2, x^3, \cdots, x^s)$, $F(\underbrace{\cdot, \cdot, \cdots, \cdot}_{d_1}, \underbrace{x^2, x^2, \cdots, x^2}_{d_2}, \cdots, \underbrace{x^s, x^s, \cdots, x^s}_{d_s})$ is a super-symmetric d_1-th order tensor form, and so on. Denote the order of the tensor F to be $d := \sum_{k=1}^s d_s$, then Corollary 4.2 immediately implies that

$$
\begin{array}{llll}
\max & |f(x^1, x^2, \cdots, x^s)| & = & \max & |F(y^1, y^2, \cdots, y^d)| \\
\text{s.t.} & \|x^i\| = 1, \, i = 1, 2, \ldots, s & & \text{s.t.} & \|y^i\| = 1, \, i = 1, 2, \ldots, d.
\end{array}
\tag{5}
$$

Let us call the left model in the above equation to be

$$(M) \quad \max_{\|x^i\|=1, \, i=1,2,\ldots,s} f(x^1, x^2, \cdots, x^s).$$

Clearly, (M) is a generalization of the biquadratic model considered in Ling et al. [37] (with $s = 2$ and $d_1 = d_2 = 2$), and the multi-quadratic model considered by So [56] (with $d_1 = d_2 = \cdots = d_s =$

2). Equation (5) also suggests a method to solve (M) by resorting to its multilinear form relaxation (T), where the MBI method applies. On the other hand, model (M) can also be solved by directly adopting the MBI method, given that for any fixed $(x^1, \cdots, x^{i-1}, x^{i+1}, \cdots, x^s)$, the maximization over $\|x^i\| = 1$ can be efficiently solved, which in this case is the model (H) with degree d_i. In particular, we can immediately apply the MBI method to solve the biquadratic model and the multi-quadratic model, since the corresponding subproblem is an eigenvalue problem. We will also test our MBI method in a tri-quadratic case of the model (M) in the next section.

4.2 Finding a KKT point for (H) using MBI via (T)

Corollary 4.2 suggests a way to solve the homogenous polynomial optimization model (H) by resorting to a seemingly more relaxed tensor optimization model (T). However, the equivalence is only established at optimality. Nevertheless, one may still search for a KKT solution for (H), by means of searching for a KKT solution for (T) with identical block variables. (Corollary 4.2 guarantees that such a special KKT point exists, and so the search is valid.) According to our computational experiences, this local search process works very well. In most cases, the KKT solution so-obtained is the true global optimal solution of (H).

Let us formalize this search process as follows. We shall work with the version of (T) and (H) with an absolute sign in the objective function, like in Corollary 4.2. This allows us to swap the direction from x to $-x$ without affecting its objective. As we discussed earlier, adding an absolute sign does not change the problem when d is odd, and also solves (H) when d is even if we modify the objective by adding a (constant) positive term, as we discussed in the previous subsection.

Algorithm KKT: Finding a KKT point for (H)

0 Input a KKT solution, say $(x_0^1, x_0^2, \cdots, x_0^d)$, of ($T$) with objective value f_0. Set $k := 0$ and $(r_0^1, r_0^2, \cdots, r_0^d) := (x_0^1, x_0^2, \cdots, x_0^d)$.

1 If $x_k^1 = \pm x_k^2 = \cdots = \pm x_k^d$, stop. Otherwise, find the closest but not identical pair among these d vectors, i.e., solve

$$\max_{1 \leq i < j \leq d, \, (x_k^i)^{\mathrm{T}} x_k^j \neq 1} (x_k^i)^{\mathrm{T}} x_k^j.$$

Denote its optimal solution to be (i_k, j_k), and compute $z_k := (x_k^{i_k} + x_k^{j_k})/\|x_k^{i_k} + x_k^{j_k}\|$.

2 Set $x_{k+1}^{i_k} := z_k$, $x_{k+1}^{j_k} := z_k$ and $x_{k+1}^i := x_k^i$ for $i \in \{1, 2, \cdots, d\} \backslash \{i_k, j_k\}$. Update the objective value of (T)

$$f_{k+1} := F(x_{k+1}^1, x_{k+1}^2, \cdots, x_{k+1}^d).$$

3 If $f_{k+1} > f_k$; or if $f_{k+1} = f_k$ and there is a vector x^i $(i \in \{1, 2, \cdots, d\} \backslash \{i_k, j_k\})$ such that

$$x^i \neq \frac{F(x_{k+1}^1, \cdots, x_{k+1}^{i-1}, \cdot, x_{k+1}^{i+1}, \cdots, x_{k+1}^d)}{\|F(x_{k+1}^1, \cdots, x_{k+1}^{i-1}, \cdot, x_{k+1}^{i+1}, \cdots, x_{k+1}^d)\|};$$

in either case, starting from $(x_{k+1}^1, x_{k+1}^2, \cdots, x_{k+1}^d)$, apply Algorithm MBI to yield a KKT point $(r_{k+1}^1, r_{k+1}^2, \cdots, r_{k+1}^d)$ with a larger objective value for (T). Otherwise, it is already a KKT point for (T); set $(r_{k+1}^1, r_{k+1}^2, \cdots, r_{k+1}^d) := (x_{k+1}^1, x_{k+1}^2, \cdots, x_{k+1}^d)$.

4 Let $k := k + 1$, and go to Step 1.

The following property of Algorithm KKT is immediate.

Proposition 4.4 *For Algorithm KKT, it holds that:*

1. *Each element in the sequence $\{(r_k^1, r_k^2, \cdots, r_k^d)\}$ is a KKT point for (T). The sequence of the objective values $\{f_k\}$ for (T) is nondecreasing;*

2. *If $(r_*^1, r_*^2, \cdots, r_*^d)$ is a cluster point of the sequence $\{(r_k^1, r_k^2, \cdots, r_k^d)\}$, then $r_* := r_*^1 = \pm r_*^2 = \cdots = \pm r_*^d$. Moreover, $(r_*, r_*, \cdots, r_*)$ or $(-r_*, -r_*, \cdots, -r_*)$ is a KKT point for (T), and r_* or $-r_*$ is a KKT point for (H).*

5 Numerical Experiments

In this section, we shall present some preliminary test results for the algorithms proposed in this paper. All the computations are conducted in an Intel(R) Core(TM)2 Quad CPU 2.66GHz computer with 4GB of RAM. The supporting software is Matlab 7.8.0 (R2009a) as a platform. We use Matlab Tensor Toolbox Version 2.4 [3] whenever tensor operations are called, and we use GloptiPoly 3 [25] for general polynomial optimization for the purpose of comparison and set the relaxation order of GloptiPoly 3 by default. To simplify our implementation, we use cvx v1.2 (Grant and Boyd [17]) as a modeling tool for our MBI subroutine. The (termination) precision for these algorithms is set to be 10^{-6}. For a given maximization problem dimension/structure, we run the algorithms on a number of random instances. GloptiPoly 3 produces an upper bound for the optimal value of that instance, which turns out to be equal to the optimal value in many cases, since the MBI method typically would find a KKT solution equal to the upper bound computed by GloptiPoly 3. We count the percentage of times when this happens in our tests. Moreover, the MBI method is essentially a local improvement, and so it can be started from different initial solutions. Our tests are designed to see the performance of the MBI method over various settings. Below is a list of abbreviations to understand the results summarized in the tables to follow:

mean(P):	average ratio between solution found by MBI and upper bound by GLP;
mean(T):	average cpu seconds to solve one instance;
mean(I):	average number of iterations to solve one instance;
mean(T/I):	average cpu seconds per iteration;
MBI:	the maximum block improvement method;
dim.:	the (d, n) dimension of the test problem;
GLP:	GloptiPoly 3;
# samples:	total number of test instances;
# starts:	number of times to run MBI from random initial solutions (keep the best one);
Opt:	percentage where the MBI solutions attain the upper bound of GLP.

5.1 Randomly Simulated Data

Throughout this subsection, all the data for testing problems are generated in the following manner. First, a d-th order tensor $\mathcal{F}'$ is randomly generated, with its n^d entries following i.i.d. normal distribution, we then symmetrize $\mathcal{F}'$ to form a super-symmetric tensor $\mathcal{F}$. For co-centered ellipsoidal constraints, we generate $n \times n$ matrix Q'_j $(j = 1, 2, \ldots, m)$, whose entries follow i.i.d. normal distribution, and then let $Q_j = (Q'_j)^{\mathrm{T}} Q'_j$. For comparison, we call GloptiPoly 3 to get optimal

13

Table 1: Numerical results for (E_1) when $n = 2$ and $n = 3$

dim.	# samples	# starts	GLP	MBI		
			mean(T)	Opt	mean(T)	mean(P)
$(4, 2)$	10	1	1.5961	80%	0.1257	90.83%
		2	idem	90%	0.1352	94.62%
		3	idem	100%	0.1472	100%
$(4, 3)$	10	1	31.6348	20%	0.1735	84.03%
		2	idem	50%	0.2595	92.67%
		3	idem	60%	0.3113	93.52%
		4	idem	90%	0.3466	98.97%

value and optimal solution if possible, or else we get an upper bound of the optimal value if GloptiPoly 3 fails to solve the given problem instance.

5.1.1 Multilinear Tensor Function over Spherical Constraints

In this part, we present some numerical tests on (T). In particular, we consider

$$(E_1) \quad \max \quad F(x, y, z, w) = \sum_{1 \le i, j, k, l \le n} \mathcal{F}_{ijkl} x_i y_j z_k w_l$$
$$\text{s.t.} \quad \|x\| = \|y\| = \|z\| = \|w\| = 1,$$
$$x, y, z, w \in \Re^n,$$

where tensor $\mathcal{F}$ is super-symmetric. The starting points (x_0, y_0, z_0, w_0) for Algorithm MBI in our numerical experiments are all randomly generated. In our tests, we consider all the variables in the same constraint set, and dimensions are set to be $n = 2$ or $n = 3$. Here the total dimension of the test problems is chosen to be low since for our comparison we need to use GloptiPoly 3 which only works for low dimensions.

The comparison is listed in Table 1 for (E_1). Evidently, the results show that Algorithm MBI finds good quality solutions very quickly. The more starts we used to run the MBI algorithm, the higher chance we get an optimal solution. In some cases, GloptiPoly 3 is only capable of providing an upper bound; however our MBI solution achieves these upper bound, proving the optimality of both the GloptiPoly 3 bound and the MBI solution. Besides, a majority of our simulation results show that the KKT point (x^*, y^*, z^*, w^*) of (E_1) is automatically a KKT point for the homogeneous polynomial case, namely their block variables are identical already.

5.1.2 Tests of Another Implementation of MBI

Algorithm MBI is optimizing one block while fixing all other blocks. As mentioned in Section 4, simultaneously optimizing over *two* blocks of variables while fixing other blocks works under the MBI framework as well. Indeed, the similar procedures of Algorithm MBI still perform efficiently, and also the convergence is guaranteed. For convenience, we call this modified procedures MBI'. In this part, we test the performance of our methods MBI and MBI' for (T), when $d = 6$:

$$(E_2) \quad \max \quad M(x, y, z, w, p, q) = \sum_{1 \le i, j, k, l, s, t \le n} \mathcal{M}_{ijklst} x_i y_j z_k w_l p_s q_t$$
$$\text{s.t.} \quad \|x\| = \|y\| = \|z\| = \|w\| = \|p\| = \|q\| = 1,$$
$$x, y, z, w, p, q \in \Re^n,$$

Table 2: Numerical results for (E_2) when $n = 5, 10, 15$

dim.	# samples	MBI			MBI$'$		
		mean(T)	mean(I)	mean(T/I)	mean(T)	mean(I)	mean(T/I)
$(6, 5)$	10	0.3087	57.3	0.0055	0.1438	20.4	0.0077
$(6, 10)$	10	4.7894	86.7	0.0551	1.1106	38.0	0.0297
$(6, 15)$	10	75.1913	127.3	0.5901	22.9004	68.3	0.3346

Table 3: Numerical results for (E_2) when $(d, n) = (6, 10)$

	1	2	3	4	5	6	7	8	9	10
MBI	4.3856	4.6422	4.8539	4.6369	4.6196	4.2168	4.5176	4.6628	4.5077	4.3039
MBI$'$	4.2235	4.8136	4.7079	4.5767	4.6906	4.4538	4.3806	4.7177	4.2873	4.3228

where tensor $\mathcal{M}$ is super-symmetric. In our tests, we choose blocks x, y as a group, and blocks z, w as another group, and blocks p, q as the last group, when implementing MBI$'$. Algorithms MBI and MBI$'$ start from the same point $(x_0, y_0, z_0, w_0, p_0, q_0)$ which are all randomly generated as before.

Two test sets are reported for (E_2). Table 2 reports the average computational time, and Table 3 reports the average objective value, where $(d, n) = (6, 10)$. In Table 3, we test 10 randomly instances, and each entry is the average objective value by running the corresponding algorithm 20 times. Tables 2 and 3 show that Algorithm MBI$'$ is comparable to Algorithm MBI in terms of the solution quality produced; however, Algorithm MBI$'$ requires much less computational effort on average. This means that the MBI approach is quite flexible and various innovative implementations are possible, and should in fact be encouraged.

5.1.3 General Polynomial Function over Quadratic Constraints

In this part, we report numerical tests on (Q) when $d = 4$:

$$
\begin{aligned}
(E_3) \quad \max \quad & p(x) = F_4(x, x, x, x) + F_3(x, x, x) + F_2(x, x) + F_1(x) \\
\text{s.t.} \quad & x^{\mathrm{T}} Q_j x \le 1, j = 1, 2, ..., m, \\
& x \in \Re^n,
\end{aligned}
$$

where tensors $\mathcal{F}_4 \in \Re^{n^4}$, $\mathcal{F}_3 \in \Re^{n^3}$, $\mathcal{F}_2 \in \Re^{n^2}$ and $\mathcal{F}_1 \in \Re^n$ are super-symmetric, and $Q_j \succeq 0$ for $j = 1, 2, \ldots, m$. One natural way to handle an inhomogeneous polynomial function $p(x)$ is through homogenization, e.g., the technique used in [22]. To be specific, by introducing an auxiliary new variable x_h, which is set to be 1, we can homogenize function $p(x)$ as

$$
p(x) = F\left(\begin{pmatrix} x \\ x_h \end{pmatrix}, \begin{pmatrix} x \\ x_h \end{pmatrix}, \begin{pmatrix} x \\ x_h \end{pmatrix}, \begin{pmatrix} x \\ x_h \end{pmatrix} \right) := F(\bar{x}, \bar{x}, \bar{x}, \bar{x}) = f(\bar{x}),
$$

where $f(\bar{x})$ is an $(n + 1)$-dimensional homogeneous polynomial function of degree 4, and its associated 4-th order super-symmetric tensor form $\mathcal{F} \in \Re^{(n+1)^4}$. Therefore, by denoting $\bar{x} := \begin{pmatrix} x \\ 1 \end{pmatrix}$, we

Table 4: CPU seconds of GloptiPoly 3 and MBI for (E_3) when $m = 15$

n	5	10	15	20	30	40
GLP	2.1830	14.6947	578.3944	∞	∞	∞
MBI	24.8763	42.1047	42.9723	43.6567	44.8106	44.9569

may equivalently rewrite (E_3) as

$$
\begin{aligned}
(\bar{E}_3) \quad \max \quad & f(\bar{x}) = F\left(\begin{pmatrix} x \\ 1 \end{pmatrix}, \begin{pmatrix} x \\ 1 \end{pmatrix}, \begin{pmatrix} x \\ 1 \end{pmatrix}, \begin{pmatrix} x \\ 1 \end{pmatrix} \right) \\
\text{s.t.} \quad & x^{\mathrm{T}} Q_j x \leq 1, j = 1, 2, ..., m, \\
& x \in \Re^n.
\end{aligned}
$$

We shall first call Algorithm MBI to solve the multilinear relaxation problem for $(\bar{E}_3)$, and get a KKT point, to be denoted by $(x_*^1, x_*^2, x_*^3, x_*^4)$. Then, we select the best one from those four vectors as a feasible point for the original model (E_3), namely, $x_{\mathrm{MBI}} = \arg\max_{1 \leq i \leq 4}\{p(x_*^i)\}$. Unlike the equivalence between (H) and (T) followed from Corollary 4.2 due to its special structure, $(\bar{E}_3)$ may not be equivalent to its tensor relaxation problem. Hence, in the last set of tests, starting from the point x_{MBI}, we further apply a projected gradient method [7] (denoted by 'PGM' in the table below) to improve the solution of (E_3). For an overview of gradient projection methods, one is referred to [5]. This method is also used as a supplement in [62, 54] for handling homogeneous polynomial optimization over ball constraint or spherical constraint. The reason to apply the projected gradient method is that this method converges to a KKT point of the problem concerned, and also the optimal projection from $\Re^n$ onto the ellipsoidal constraints set $E = \{x \in \Re^n \mid x^{\mathrm{T}} Q_j x \leq 1, j = 1, 2, ..., m\}$ can be formulated as an SOCP problem

$$
\begin{aligned}
\min \quad & \|x - y\| \\
\text{s.t.} \quad & x \in E,
\end{aligned}
$$

where $y \in \Re^n$ is given, which we call cvx to solve under the same computational platform. The starting points for MBI are all randomly generated as before. Two test sets are constructed for (E_3). First, we fix $m = 15$ for varying n, and we test the performance of GloptiPoly 3 and MBI in terms of the computational time, regardless of the quality of x_{MBI} obtained by running MBI once (recall the relaxation order of GloptiPoly 3 is set by default). Numerical results are listed in Table 4. Each entry is the average time of 10 randomly generated instances. From Table 4, we conclude that the computational time of MBI is insensitive to the dimension n, while GloptiPoly 3 is very sensitive to the dimension. In fact, the computational time of MBI is much less than that of GloptiPoly 3 when the dimension n gets large.

Second, we fix $m = 10$ and pick some lower dimensions n, whose problems can be efficiently solved by GloptiPoly 3. We then test the performance of MBI. Specifically, we solve (E_3) by three different approaches: (1) directly using GloptiPloy 3; (2) applying MBI with randomly generated starting points to get the point x_{MBI}; (3) using projected gradient method with the starting point x_{MBI}. Numerical results are summarized in Table 5, which show the excellent performance of the MBI method. GloptiPoly 3 is a powerful tool for solving (E_3) with low dimensions. However, the MBI method works very well for polynomial optimization over ellipsoidal constraints even in large dimensions.

Table 5: Numerical results for (E_3) when $m = 10$

dim.	# samples	GLP	MBI			MBI+PGM		
		mean(T)	Opt	mean(T)	mean(P)	Opt	mean(T)	mean(P)
$(4, 5)$	20	1.79	20%	22.81	85.57%	95%	29.68	98.21%
$(4, 10)$	20	13.01	0%	38.95	85.79%	95%	48.66	99.93%
$(4, 12)$	20	66.73	0%	41.36	89.61%	100%	50.13	100%

5.2 Applications

In this subsection, we shall test our proposed algorithms by using data from real applications, including rank-one approximation of super-symmetric tensors, and magnetic resonance imaging (MRI).

5.2.1 Rank-One Approximation of Super-Symmetric Tensors

As discussed in Section 4, homogeneous polynomial optimization over spherical constraint is equivalent to the best rank-one approximation of super-symmetric tensors, hence solvable by our methods. We consider an example in this subsection from Kofidis and Regalia (Example 1 of [30]). The authors of [30] used this example to show that their proposed method S-HOPM did not converge for the particular super-symmetric tensor $\mathcal{G} \in \Re^{3 \times 3 \times 3 \times 3}$ with entries

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

We will test this example using MBI. In our setting, the best rank-one approximation of the tensor $\mathcal{G}$ is formulated as

$$
\begin{aligned}
(E_4) \quad \max \quad & \sum_{1 \le i,j,k,l \le 3} \mathcal{G}_{ijkl} x_i x_j x_k x_l \\
\text{s.t.} \quad & \|x\| = 1,\, x \in \Re^3.
\end{aligned}
$$

Since the order of $\mathcal{G}$ is even, we choose $\eta = 6$, and construct a modified and equivalent optimization problem of (E_4)

$$
\begin{aligned}
(E_5) \quad \max \quad & \sum_{1 \le i,j,k,l \le 3} (\mathcal{G} + \eta\mathcal{H})_{ijkl} x_i x_j x_k x_l \\
\text{s.t.} \quad & \|x\| = 1,\, x \in \Re^3,
\end{aligned}
$$

where $\mathcal{H}$ is a 4-th order super-symmetric tensor associated with the homogenous polynomial $h(x) = (x^{\mathrm{T}} x)^2$. For the reformulated problem, we apply Algorithm MBI to the multilinear tensor form relaxation of (E_5):

$$
\begin{aligned}
(E_6) \quad \max \quad & G(x, y, z, w) = \sum_{1 \le i,j,k,l \le 3} (\mathcal{G} + \eta\mathcal{H})_{ijkl} x_i y_j z_k w_l \\
\text{s.t.} \quad & \|x\| = \|y\| = \|z\| = \|w\| = 1, \\
& x, y, z, w \in \Re^3.
\end{aligned}
$$

By using MBI with randomly generated starting points, we get three local maximum solutions for (E_6). For each local maxima (x, y, z, w) we found, it shares the same directions among these 4

vectors when MBI stops, i.e., $x = y = z = w$. Hence, it provides a local maxima for the original model (E_5), which is also a local maxima for (E_4). In Figure 1, the total number of iterations in each round of MBI is presented, for each local maxima we have found. Indeed, MBI converges very quickly to a local maxima. The optimal value for (E_4) is 0.8893 (recall we should subtract 6 in the function $G(x, y, z, w)$), and the optimal solutions are $x^* = \pm(0.6671, 0.2487, -0.7022)$. Hence, the best rank-one approximation for the super-symmetric tensor $\mathcal{G}$ is $0.8893\, x^* \otimes x^* \otimes x^* \otimes x^*$.
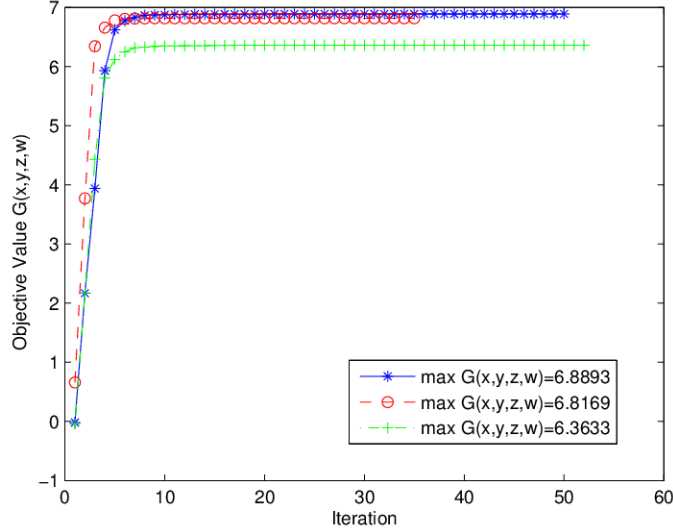

Figure 1: Convergence results of MBI for (E_6)

5.2.2 Magnetic Resonance Imaging (MRI)

Finally, we shall conclude this section by considering one real data set for polynomial optimization in Magnetic Resonance Imaging (MRI). Ghosh et al. [16] formulated a fiber detection problem in diffusion MRI by maximizing a homogeneous polynomial function over spherical constraint. In this particular case, the following polynomial optimization model is considered

$$
\begin{aligned}
\max \quad & f(x) \\
\text{s.t.} \quad & \|x\| = 1, \, x \in \Re^3,
\end{aligned}
$$

where $f(x)$ is a homogeneous polynomial of even degree d. The problem lives in three dimensions as in the real world, and all its local maxima have physical meanings for MRI.

We shall test our Algorithm MBI, by using a set of data provided by A. Ghosh and R. Deriche. The corresponding objective function $f(x)$ is

$$
\begin{aligned}
& 0.74694\, x_0{}^4 - 0.435103\, x_0{}^3 x_1 + 0.454945\, x_0{}^2 x_1{}^2 + 0.0657818\, x_0 x_1{}^3 + x_1{}^4 \\
+\ & 0.37089\, x_0{}^3 x_2 - 0.29883\, x_0{}^2 x_1 x_2 - 0.795157\, x_0 x_1{}^2 x_2 + 0.139751\, x_1{}^3 x_2 + 1.24733\, x_0{}^2 x_2{}^2 \\
+\ & 0.714359\, x_0 x_1 x_2{}^2 + 0.316264\, x_1{}^2 x_2{}^2 - 0.397391\, x_0 x_2{}^3 - 0.405544\, x_1 x_2{}^3 + 0.794869\, x_2{}^4,
\end{aligned}
$$

where $x = (x_0, x_1, x_2)^{\mathrm{T}}$. By choosing $\eta = 2$, we adopt the same procedures for rank-one approximation of super-symmetric tensors discussed in Section 5.2.1 to solve the MRI problem. GloptiPoly 3 is also called for comparison. The numerical results are reported in Table 6. MBI is able to find all the three local maxima, while GloptiPoly 3 finds only the global maximum.

18

Table 6: Numerical results for MRI

Method	KKT solution	Objective value
MBI	$\pm(0.0116, 0.9992, 0.0382)$	1.0031
	$\pm(0.3166, 0.2130, -0.9243)$	0.9213
	$\pm(0.9542, -0.1434, 0.2624)$	0.8428
GLP	$\pm(0.0116, 0.9992, 0.0382)$	1.0031

Acknowledgments

References

[1] W. Ai, Y. Huang, and S. Zhang, *New results on hermitian matrix rank-one decomposition.* Mathematical Programming, Series A, 128 (2011) 253–283.

[2] M. F. Anjos and J. B. Lasserre, *Handbook of Semidefinite, Conic and Polynomial Optimization*, Springer, New York, 2011.

[3] B. W. Bader and T. G. Kolda, *Matlab Tensor Toolbox Version 2.4*, http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox, 2010.

[4] A. Barmpoutis, B. Jian, B. C. Vemuri, and T. M. Shepherd, *Symmetric positive 4th order tensors & their estimation from diffusion weighted MRI*, in N. Karssemijer and B. Lelieveldt ed., LNCS 4584 (2007) 308–319.

[5] D. P. Bertsekas, *Nonlinear Programming*, second edition, Athena Scientific, Belmont, 1999.

[6] J. Bochnak and J. Siciak, *Polynomials and multilinear mappings in topological vector spaces*, Studia Math., 39 (1971) 59–76.

[7] P. H. Calamai and J. J. Moré, *Projected gradient methods for linearly constrained problems.* Mathematical Programming, 39 (1987) 93–116.

[8] J. D. Carroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970) 283–319.

[9] G. Dahl, J. M. Leinaas, J. Myrheim, and E. Ovrum, *A tensor product matrix approximation problem in quantum physics*, Linear Algebra and Its Applications, 420 (2007) 711–725.

[10] E. F. Gonzalez and Y. Zhang, *Accelerating the Lee-Seung algorithm for nonnegative matrix factorization*, preprint, March 2005.

[11] L. Grippo and M. Sciandrone, *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints*, Operations Research Letters, 26 (2000) 127–136.

[12] E. de Klerk, *The complexity of optimizing over a simplex, hypercube or sphere: a short survey*, Central European Journal of Operations Research, 16 (2008) 111–125.

[13] E. de Klerk, M. Laurent, and P. A. Parrilo, *A PTAS for the minimization of polynomials of fixed degree over the simplex*, Theoretical Computer Science, 261 (2006) 210–225.

[14] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \cdots, R_N)$ approximation of higher-order tensors*, SIAM Journal on Matrix Analysis and Applications, 21 (2000) 1324–1342.

[15] L. De Lathauwer, and D. Nion, *Decompositions of a higher-order tensor in block terms—Part III: Alternating least squares algorithms*, SIAM Journal on Matrix Analysis and Applications, 30 (2008) 1067–1083.

[16] 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*, Computational Diffusion MRI Workshop (CDMRI'08), New York, 2008.

[17] M. Grant and S. Boyd, *CVX: Matlab Software for Disciplined Convex Programming, version 1.2*, http://cvxr.com/cvx, 2010.

[18] L. Gurvits, *Classical deterministic complexity of Edmonds' problem and quantum entanglement*, Proceedings of the Thirty-Fifth ACM Symposium on Theory of Computing, 10–19, ACM, New York, 2003.

[19] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis*, UCLA Working Papers in Phonetics, 16, 1–84, http://publish.uwo.ca/~harshman/wpppfac0.pdf, 1970.

[20] S. He, M. Li, S. Zhang and Z.-Q. Luo, *A nonconvergent example for the iterative water-filling algorithm*, Numerical Algebra, Control and Optimization, 1 (2011) 147–150.

[21] S. He, Z. Li, S. Zhang, *Approximation algorithms for homogeneous polynomial optimization with quadratic constraints*, Mathematical Programming, Series B, 125 (2010) 353–383.

[22] S. He, Z. Li, and S. Zhang, *General constrained polynomial optimization: an approximation approach*, Technical Report SEEM2009-06, Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, 2009.

[23] S. He, Z. Li, and S. Zhang, *Approximation algorithms for discrete polynomial optimization*, Technical Report SEEM2010-01, Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, 2010.

[24] D. Henrion and J. B. Lasserre, *GloptiPoly: Global optimization over polynomials with Matlab and SeDuMi*, ACM Tran. Math. Soft., 29 (2003) 165–194.

[25] D. Henrion, J. B. Lasserre, and J. Loefberg, *GloptiPoly 3: Moments, optimization and semidefinite programming*, Optimization Methods and Software, 24 (2009) 761–779.

[26] C. J. Hillar and L.-H. Lim, *Most tensor problems are NP hard*, Preprint, http://www.stat.uchicago.edu/~lekheng/work/np.pdf, 2009.

[27] J.-B. Hiriart-Urruty, *A new series of conjectures and open questions in optimization and matrix analysis*, ESAIM: Control, Optimisation and Calculus of Variations, 15 (2009) 454–470.

[28] P. M. J. Hof, C. Scherer and P.S.C. Heuberger, *Model-Based Control: Bridging Rigorous Theory and Advanced Technology, Part 1* 49–68. Springer–Verlag, Berlin Heidelberg, 2009.

[29] S. Khot and A. Naor, *Linear equations modulo 2 and the L_1 diameter of convex bodies*, the 48th Annual IEEE Symposium on Foundations of Computer Science (2007) 318–328.

[30] E. Kofidis and P. A. Regalia, *On the best rank-1 approximation of higher order supersymmetric tensors*, SIAM Journal on Matrix Analysis and Applications, 23 (2002) 863–884.

[31] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Review, 51 (2009) 455–500.

[32] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM Journal on Optimization, 11 (2001) 796–817.

[33] J. B. Lasserre, *Polynomials nonnegative on a grid and discrete representations*, Transactions of the American Mathematical Society, 354 (2001) 631–649.

[34] J. B. Lasserre, *Convergent SDP relaxations in polynomial optimization with sparsity*, SIAM Journal on Optimization, 17 (2006) 822–843.

[35] J. B. Lasserre, *A sum of squares approximation of nonnegative polynomials*, SIAM Review, 49 (2007) 651–669.

[36] Z. Li, *Polynomial Optimization Problems—Approximation Algorithms and Applications*, Ph.D. Thesis, The Chinese Univesrity of Hong Kong, June 2011.

[37] C. Ling, J. Nie, L. Qi, and Y. Ye, *Biquadratic optimization over unit spheres and semidefinite programming relaxations*, SIAM Journal on Optimization, 20 (2009) 1286–1310.

[38] Z.-Q. Luo and J.-S. Pang, *Analysis of iterative waterfilling algorithm for multiuser power control in digital subscriber lines*, EURASIP Journal on Applied Signal Processing, 2006 (2006) 1–10.

[39] Z.-Q. Luo and S. Zhang, *A semidefinite relaxation scheme for multivariate quartic polynomial optimization with quadratic constraints*, SIAM Journal on Optimization, 20 (2010) 1716–1736.

[40] B. Mourrain and J. P. Pavone, *Subdivision methods for solving polynomial equations*, Journal of Symbolic Computation, 44 (2009) 292–306.

[41] B. Mourrain and P. Trébuchet, *Generalized normal forms and polynomial system solving*. in M. Kauers, ed., Proc. Intern. Symp. on Symbolic and Algebraic Computation (2005), New York, ACM Press 253–260.

[42] Yu. Nesterov, *Random walk in a simplex and quadratic optimization over convex polytopes*, CORE Discussion Paper, UCL, Louvain-la-Neuve, Belgium, 2003.

[43] Yu. Nesterov and A. Nemirovski, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Studies in Applied Mathematics, 1994.

[44] Network-Enabled Optimization Systems (NEOS) Server for Optimization, http://www.neos-server.org/neos.

[45] G. Ni and Y. Wang, *On the best rank-1 approximation to higher-order symmetric tensors*, Mathematical and Computer Modelling, 46 (2007) 1345–1352.

[46] D. Nion and L. De Lathauwer, *An enhanced line search scheme for complex-valued tensor decompositions*, Application in DS-CDMA, Signal Processing, 88 (2008) 749–755.

[47] P. A. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. Dissertation, California Institute of Technology, CA, 2000.

[48] P. A. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Mathematical Programming, Series B, 96 (2003) 293–320.

[49] S. Prajna, A. Papachristodoulou, P.A. Parrilo, *Introducing SOSTOOLS: a general purpose sum of squares programming solver*, Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, 2002.

[50] L. Qi, *Eigenvalues of a real supersymmetric tensor,* Journal of Symbolic Computation, 40 (2005) 1302–1324.

[51] M. J. D. Powell, *On search directions for minimization algorithms*, Mathematical Programming, 4 (1973) 193–201.

[52] L. Qi, *Eigenvalues and invariants of tensors*, Journal of Mathematical Analysis and Applications, 325 (2007) 1363–1377.

[53] L. Qi and K. L. Teo, *Multivariate polynomial minimization and its application in signal processing*, Journal of Global Optimization, 26 (2003) 419–433.

[54] L. Qi, F. Wang, and Y. Wang, *Z-eigenvalue methods for a global polynomial optimization problem*, Mathematical Programming, Series A, 118 (2009) 301–316.

[55] M. Rajih, P. Comon, and R. A. Harshman, *Enhanced line search: A novel method to accelerate PARAFAC*, SIAM Journal on Matrix Analysis and Applications, 30 (2008) 1128–1147.

[56] A. M.-C. So, *Deterministic approximation algorithms for sphere constrained homogeneous polynomial optimization problems*, Mathematical Programming, Series B, 129 (2011) 357–382.

[57] S. Soare, J.W. Yoon, and O. Cazacu, *On the use of homogeneous polynomials to develop anisotropic yield functions with applications to sheet forming*, International Journal of Plasticity, 24 (2008) 915–944.

[58] J. F. Sturm, *SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11 & 12 (1999) 625–653.

[59] J. F. Sturm and S. Zhang, *On cones of nonnegative quadratic functions*, Mathematics of Operations Research, 28 (2003) 246–267.

[60] P. Tseng, *Convergence of a block coordinate descent method for nondifferentiable minimization*, Journal of Optimizatoin Theory and Applications, 109 (2001) 475–494.

[61] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto, *Algorithm 883: SparsePOP: A sparse semidefinite programming relaxation of polynomial optimization problems*, ACM Transactions on Mathematical Software, 35 (2008) 1–13.

[62] Y. Wang and L. Qi, *On the successive supersymmetric rank-1 decomposition of higher-order supersymmetric tensors*, Numerical Linear Algebra with Applications, 14 (2007) 503–519.

[63] S. Wright, *Accelerated block-coordinate relaxation for regularized optimization*, Working Paper, University of Wisconsin, 2010.

[64] Y. Ye and S. Zhang, *New results on quadratic minimization*, SIAM Journal on Optimization, 14 (2003) 245–267.

[65] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM Journal on Matrix Analysis and Applications, 23 (2001) 534–550.