

Approximation Methods for Complex Polynomial Optimization

Bo JIANG ^{*} Zhening LI [†] Shuzhong ZHANG [‡]

January 18, 2014

Abstract

Complex polynomial optimization problems arise from real-life applications including radar code design, MIMO beamforming, and quantum mechanics. In this paper, we study complex polynomial optimization models where the objective function takes one of the following three forms: (1) multilinear; (2) homogeneous polynomial; (3) symmetric conjugate form. On the constraint side, the decision variables belong to one of the following three sets: (1) the m -th roots of complex unity; (2) the complex unity; (3) the Euclidean sphere. We first discuss the multilinear objective function. Polynomial-time approximation algorithms are proposed for such problems with assured worst-case performance ratios, which depend only on the dimensions of the model. Then we introduce complex homogenous polynomial functions and establish key linkages between complex multilinear forms and the complex polynomial functions. Approximation algorithms for the above-mentioned complex polynomial optimization models with worst-case performance ratios are presented. Numerical results are reported to illustrate the effectiveness of the proposed approximation algorithms.

Keywords: polynomial optimization, complex programming, complex tensor, approximation algorithm, tensor relaxation.

Mathematics Subject Classification: 90C59, 90C10, 15A69, 90C26.

This paper is dedicated to Masao Fukushima in celebration of his 65th birthday.

^{*}Research Center for Management Science and Data Analytics, School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China. Email: isyebojiang@gmail.com.

[†]Department of Mathematics, University of Portsmouth, Portsmouth PO1 3HF, United Kingdom. Email: zheningli@gmail.com.

[‡]Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, MN 55455, USA. Email: zhangs@umn.edu.

1 Introduction

Polynomial optimization has received much attention in the recent years. The reason for this surge of interests is twofold. On the one hand, there is an emerging wide range of applications for polynomial optimization, for instance from biomedical engineering, control theory, graph theory, investment science, material science, quantum mechanics, signal processing, speech recognition; for specific references, see e.g. [20]. On the other hand, polynomial optimization has been found to be deeply rooted in a theoretical sense. Following the seminal work of Lasserre [19] and Parrilo [24], sum of squares (SOS) methods have become a cornerstone for general polynomial optimization. Recent developments can be found in the handbook by Anjos and Lasserre [3]. Since most of polynomial optimization problems are NP-hard, on the front of approximate solutions, various approximation algorithms have been proposed for solving certain types of high degree polynomial optimization models; we refer interested readers to the recent monograph of Li et al. [20].

Hitherto, polynomial optimization models under investigation are mostly in the domain of real numbers. Motivated by applications from signal processing, in this paper we set out to study several new classes of discrete and continuous polynomial optimization models in the complex domain. The detailed descriptions of these models can be found in Section 2. As a matter of fact, there are scattered results on complex polynomial optimization in the literature. When the objective function is quadratic, the MAX-3-CUT problem is a typical instance for the 3rd roots of unity constraint. Unity circle constrained complex optimization arises from the study of robust optimization as well as control theory [28, 5]. In particular, complex quadratic form optimization over unity constraints studied by Toker and Ozbay [28] are called complex programming. If the degree of complex polynomial is beyond quadratic, say quartic, several applications in signal processing can be found in the literature. Maricic et al. [23] proposed a quartic polynomial model for blind channel equalization in digital communication. Aittomäki and Koivunen [1] discussed the problem of beam-pattern synthesis in array signal processing problem and formulated it to be a complex quartic minimization problem. Chen and Vaidyanathan [7] studied MIMO radar waveform optimization with prior information of the extended target and clutter, by relaxing a quartic complex model. Most recently, Aubry et al. [4] managed to design a radar waveform sharing an ambiguity function behavior by resorting to a complex optimization problem. In quantum entanglement, Hilling and Sudbery [13] formulated a typical problem as a complex form optimization problem under spherical constraint, which is one of the three classes of models studied in this paper. Inspired by their work, Zhang and Qi [30] discussed the quantum eigenvalue problem, which arises from the geometric measure of entanglement of a multipartite symmetric pure state, in the complex tensor space. In fact, complex polynomial and complex tensor are interesting on their own. Eigenvalue and eigenvectors in the complex domain were already proposed and studied by Qi [25], whereas the name E-eigenvalue was coined. Very recently, Jiang et al. [17] discovered the necessary and sufficient condition for general conjugate complex polynomial functions that always take real values (see also Chapter 7.4 in Ph.D. thesis of Jiang [16]), based on which they extended the definitions of eigenvalues for conjugate type complex tensors.

Like its real-case counterpart, complex polynomial optimization is also NP-hard in general. Therefore, approximation algorithms for complex models are on high demand. However, in the literature approximation algorithms are mostly considered for quadratic models only. Ben-Tal et al. [5] first studied complex quadratic optimization whose objective function is restricted nonnegative by using complex matrix cube theorem. Zhang and Huang [29], So et al. [27] considered complex quadratic form maximization under the m -th roots of unity constraints and unity constraints. Later, Huang and Zhang [15] also considered bilinear form complex optimization models under similar constraints. For real valued polynomial optimization problems, Luo and Zhang [22]

first considered approximation algorithms for quartic optimization. At the same time, Ling et al. [21] considered a special quartic optimization model. Basically, the problem is to maximize a biquadratic form over two spherical constraints. Significant progresses have recently been made by He et al. [10, 11, 12], where the authors derived a series of approximation methods for optimization of any fixed degree polynomial function under various constrains. So [26] further considered spherically constrained homogeneous polynomial optimization and proposed a deterministic algorithm with an improved approximation ratio. For most recent development on approximation algorithms for homogeneous polynomial optimization, we refer the interested readers to [9, 14].

To the best of our knowledge, there is no result on approximation algorithms for general degree complex polynomial optimization model. A related work is due to Doherty and Wehner [8], where the authors studied spherical constrained homogeneous polynomial optimization both in the real and the complex domain, and gave an explicit approximation bound to estimate the optimal value of the problem using the SOS method, while no approximate solutions were generated there. On the other hand, it is always a possible practice of transforming a general high degree complex polynomial to the real case by doubling the problem dimension, and then resorting to the existing approximation algorithms for the real-valued polynomials [10, 11, 12, 26, 9, 14]. The latter approach, however, may lose the handle on the structure of the problem, hence misses nice properties of the complex polynomial functions. As a result, the computational costs may increase while the solution qualities may deteriorate. Exploiting the special structure of the complex model, it is often possible to get better approximation bounds, e.g. [29]. With this in mind, in this paper we shall study the complex polynomial optimization in its direct form. Let us start with some preparations next.

2 Models, notations, and organization

Throughout this paper, for any complex number $z = a + ib \in \mathbb{C}$ with $a, b \in \mathbb{R}$, its real part is denoted by $\text{Re } z = a$, and its modulus by $|z| = \sqrt{z^H z} = \sqrt{a^2 + b^2}$. For $x \in \mathbb{C}^n$, its norm is denoted by $\|x\| := (\sum_{i=1}^n |x_i|^2)^{\frac{1}{2}}$.

Given a d -th order complex tensor $\mathcal{F} = (\mathcal{F}_{i_1 i_2 \dots i_d}) \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_d}$, its associated multilinear form is defined as

$$L(x^1, x^2, \dots, x^d) := \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_d=1}^{n_d} \mathcal{F}_{i_1 i_2 \dots i_d} x_{i_1}^1 x_{i_2}^2 \dots x_{i_d}^d,$$

where the variables $x^k \in \mathbb{C}^{n_k}$ for $k = 1, 2, \dots, d$, with ‘ L ’ standing for ‘multilinearity’.

Closely related to multilinear form is homogeneous polynomial function, or, more explicitly

$$H(x) := \sum_{1 \leq i_1 \leq i_2 \leq \dots \leq i_d \leq n} a_{i_1 i_2 \dots i_d} x_{i_1} x_{i_2} \dots x_{i_d},$$

where the variable $x \in \mathbb{C}^n$, with ‘ H ’ standing for ‘homogeneous polynomial’. Associated with any homogeneous polynomial is a super-symmetric complex tensor $\mathcal{F} \in \mathbb{C}^{n^d}$; i.e., its entries $\mathcal{F}_{i_1 i_2 \dots i_d}$ ’s are invariant under permutations of its indices $\{i_1, i_2, \dots, i_d\}$. In this sense,

$$\mathcal{F}_{i_1 i_2 \dots i_d} = \frac{a_{i_1 i_2 \dots i_d}}{|\Pi(i_1 i_2 \dots i_d)|} \quad \forall 1 \leq i_1 \leq i_2 \leq \dots \leq i_d \leq n,$$

where $\Pi(i_1 i_2 \dots i_d)$ is the set of all distinct permutations of the indices $\{i_1, i_2, \dots, i_d\}$.

In light of multilinear form L associated with a super-symmetric tensor, homogeneous polynomial H is obtained by letting $x^1 = x^2 = \dots = x^d$; i.e., $H(x) = L(\underbrace{x, x, \dots, x}_d)$. Furthermore, He et

al. [10] established an essential linkage between multilinear forms and homogeneous polynomials in the real domain.

Lemma 2.1 *Suppose $x^1, x^2, \dots, x^d \in \mathbb{R}^n$, and $\xi_1, \xi_2, \dots, \xi_d$ are i.i.d. symmetric Bernoulli random variables (taking 1 and -1 with equal probability). For any super-symmetric tensor $\mathcal{F} \in \mathbb{R}^{n^d}$ with its associated multilinear form L and homogeneous polynomial H , it holds that*

$$\mathbb{E} \left[\prod_{i=1}^d \xi_i H \left(\sum_{k=1}^d \xi_k x^k \right) \right] = d! L(x^1, x^2, \dots, x^d).$$

With Lemma 2.1 in place, tensor relaxation [10] is proposed to solve homogeneous polynomial optimization problems, by relaxing the objective function to a multilinear form.

In terms of the optimization, the real part of the above functions (multilinear form and homogeneous polynomial) is usually considered. In [16, 17], the conjugate partial-symmetric complex tensors were introduced, which are extended from Hermitian matrices.

Definition 2.2 *An even order complex tensor $\mathcal{F} \in \mathbb{C}^{n^{2d}}$ is called conjugate partial-symmetric if*

- (1) $\mathcal{F}_{i_1 \dots i_d i_{d+1} \dots i_{2d}} = \overline{\mathcal{F}_{i_{d+1} \dots i_{2d} i_1 \dots i_d}}$ and
- (2) $\mathcal{F}_{i_1 \dots i_d i_{d+1} \dots i_{2d}} = \mathcal{F}_{j_1 \dots j_d j_{d+1} \dots j_{2d}} \forall (j_1 \dots j_d) \in \Pi(i_1 \dots i_d), (j_{d+1} \dots j_{2d}) \in \Pi(i_{d+1} \dots i_{2d})$

hold for all $1 \leq i_1, \dots, i_{2d} \leq n$.

Associated with any conjugate partial-symmetric tensor, the following symmetric conjugate form

$$C(\bar{x}, x) := L(\underbrace{\bar{x}, \dots, \bar{x}}_d, \underbrace{x, \dots, x}_d) = \sum_{1 \leq i_1, \dots, i_d, j_1, \dots, j_d \leq n} \mathcal{F}_{i_1 \dots i_d j_1 \dots j_d} \overline{x_{i_1} \dots x_{i_d}} x_{j_1} \dots x_{j_d}$$

always takes real value for any $x \in \mathbb{C}^n$. Besides, any symmetric conjugate form C uniquely determines a conjugate partial-symmetric tensor. For details, one is referred to [17] or Chapter 7.4 of [16]. In the above expression, ‘ C ’ signifies ‘conjugate’.

The following commonly encountered constraint sets for complex polynomial optimization are considered in this paper:

- The m -th roots of unity constraint: $\Omega_m = \{1, \omega_m, \dots, \omega_m^{m-1}\}$, where $\omega_m = e^{i\frac{2\pi}{m}} = \cos \frac{2\pi}{m} + i \sin \frac{2\pi}{m}$. Denote $\Omega_m^n = \{x \in \mathbb{C}^n \mid x_i \in \Omega_m, i = 1, 2, \dots, n\}$.
- The unity constraint: $\Omega_\infty = \{z \in \mathbb{C} \mid |z| = 1\}$. Denote $\Omega_\infty^n = \{x \in \mathbb{C}^n \mid x_i \in \Omega_\infty, i = 1, 2, \dots, n\}$.
- The complex spherical constraint: $\mathbf{S}^n = \{x \in \mathbb{C}^n \mid \|x\| = 1\}$.

Throughout this paper we assume $m \geq 3$, to ensure that the decision variables being considered are essentially complex.

In this paper, we shall discuss various complex polynomial optimization models. The objective function will be one of the three afore-mentioned complex polynomial functions (L , H , and C), or their real parts whenever is applicable; the constraint set is one of the three kinds as discussed above. The organization of the paper is as follows. Maximizing multilinear form over three types of constraint sets will be discussed in Section 3, i.e., models (L_m), (L_∞) and (L_S), with the subscript indicating the constraint for: the m -th roots of unity, the unity, and the complex sphere, respectively. Section 4 deals with maximization of homogeneous polynomial over three types of constraints, i.e., models (H_m), (H_∞) and (H_S). Section 5 discusses maximization of symmetric conjugate form over three types of constraints, i.e., models (C_m), (C_∞) and (C_S). Finally in Section 6, we conduct some numerical tests and report the performance of the algorithm studies in this paper.

Table 1: Organization of the paper and the approximation results

Section	Model	Theorem	Approximation performance ratio
3.1	(L_m)	3.4	$\tau_m^{d-2} (2\tau_m - 1) \left(\prod_{k=1}^{d-2} n_k \right)^{-\frac{1}{2}}$ where $\tau_m = \frac{m^2}{4\pi} \sin^2 \frac{\pi}{m}$
3.2	(L_∞)	3.6	$0.7118 \left(\frac{\pi}{4} \right)^{d-2} \left(\prod_{k=1}^{d-2} n_k \right)^{-\frac{1}{2}}$
3.3	(L_S)	3.7	$\left(\prod_{k=1}^{d-2} n_k \right)^{-\frac{1}{2}}$
4.1	(H_m)	4.3, 4.4	$\tau_m^{d-2} (2\tau_m - 1) d! d^{-d} n^{-\frac{d-2}{2}}$
4.2	(H_∞)	4.5	$0.7118 \left(\frac{\pi}{4} \right)^{d-2} d! d^{-d} n^{-\frac{d-2}{2}}$
4.3	(H_S)	4.6	$d! d^{-d} n^{-\frac{d-2}{2}}$
5.1	(C_m)	5.3, 5.4	$\tau_m^{2d-2} (2\tau_m - 1) (d!)^2 (2d)^{-2d} n^{-(d-1)}$
5.2	(C_∞)	5.5	$0.7118 \left(\frac{\pi}{4} \right)^{2d-2} (d!)^2 (2d)^{-2d} n^{-(d-1)}$
5.2	(C_S)	5.6	$(d!)^2 (2d)^{-2d} n^{-(d-1)}$

As a matter of notation, for any maximization problem $(P) : \max_{x \in X} p(x)$, we denote $v(P)$ to be the optimal value, and $\underline{v}(P)$ to be the optimal value of its minimization counterpart $(\min_{x \in X} p(x))$.

Definition 2.3 (1) A maximization problem $(P) : \max_{x \in X} p(x)$ admits a polynomial-time approximation algorithm with approximation ratio $\tau \in (0, 1]$, if $v(P) \geq 0$ and a feasible solution $\hat{x} \in X$ can be found in polynomial-time, such that $p(\hat{x}) \geq \tau v(P)$.

(2) A maximization problem $(P) : \max_{x \in X} p(x)$ admits a polynomial-time approximation algorithm with relative approximation ratio $\tau \in (0, 1]$, if a feasible solution $\hat{x} \in X$ can be found in polynomial-time, such that $p(\hat{x}) - \underline{v}(P) \geq \tau (v(P) - \underline{v}(P))$.

In this paper, we reserve τ to denote the approximation ratio. All the optimization models considered in this paper are NP-hard in general, even restricting the domain to be real. We shall propose polynomial-time approximation algorithms with worst-case performance ratios for the models concerned, when the degree of these polynomial functions, d or $2d$, is fixed. These approximation ratios depend only on the dimensions of the problems, or are data-independent. We shall start off by presenting Table 1 which summarizes the approximation results and the organization of the paper.

3 Complex multilinear form optimization

Let us consider optimization of complex multilinear forms, under three types of constraints described in Section 2. Specifically, the models under consideration are:

$$\begin{aligned}
(L_m) \quad & \max \quad \operatorname{Re} L(x^1, x^2, \dots, x^d) \\
& \text{s.t.} \quad x^k \in \Omega_m^{n_k}, k = 1, 2, \dots, d; \\
(L_\infty) \quad & \max \quad \operatorname{Re} L(x^1, x^2, \dots, x^d) \\
& \text{s.t.} \quad x^k \in \Omega_\infty^{n_k}, k = 1, 2, \dots, d; \\
(L_S) \quad & \max \quad \operatorname{Re} L(x^1, x^2, \dots, x^d) \\
& \text{s.t.} \quad x^k \in \mathbf{S}^{n_k}, k = 1, 2, \dots, d.
\end{aligned}$$

Associated with multilinear form objective is a d -th order complex tensor $\mathcal{F} \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_d}$. Without loss of generality, we assume that $n_1 \leq n_2 \leq \dots \leq n_d$ and $\mathcal{F} \neq 0$. The multilinear form

optimization models are interesting on their own. For example, typical optimization problem in quantum entanglement problem [13] is in the formulation of (L_S) .

3.1 Multilinear form in the m -th roots of unity

When $d = 2$, (L_m) is already NP-hard, even for $m = 2$. In this case, (L_m) is to compute $\infty \mapsto 1$ -norm of a matrix, and the best approximation ratio is $\frac{2\ln(1+\sqrt{2})}{\pi} \approx 0.56$ due to Alon and Naor [2]. Huang and Zhang [15] studied general m when $d = 2$, and proposed polynomial-time randomized approximation algorithm with constant worst-case performance ratio. Specifically the ratio is $\frac{m^2}{4\pi}(1 - \cos \frac{2\pi}{m}) - 1 = 2\tau_m - 1$ for $m \geq 3$, where $\tau_m := \frac{m^2}{8\pi}(1 - \cos \frac{2\pi}{m}) = \frac{m^2}{4\pi} \sin^2 \frac{\pi}{m}$ throughout this paper.

To proceed to the general degree d , let us start with the case $d = 3$.

$$(L_m^3) \quad \max \quad \operatorname{Re} L(x, y, z) \\ \text{s.t.} \quad x \in \Omega_m^{n_1}, y \in \Omega_m^{n_2}, z \in \Omega_m^{n_3}.$$

Denote $W = xy^T$. It is easy to observe that $W_{ij} = x_i y_j \in \Omega_m$ for all (i, j) , implying $W \in \Omega_m^{n_1 \times n_2}$. The above problem can be relaxed to

$$(L_m^2) \quad \max \quad \operatorname{Re} \hat{L}(W, z) := \operatorname{Re} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \mathcal{F}_{ijk} W_{ij} z_k \\ \text{s.t.} \quad W \in \Omega_m^{n_1 \times n_2}, z \in \Omega_m^{n_3}.$$

This is exactly (L_m) with $d = 2$, which admits a polynomial-time approximation algorithm with approximation ratio $2\tau_m - 1$ in [15]. Denote the approximate solution of (L_m^2) to be (\hat{W}, \hat{z}) , i.e.,

$$\operatorname{Re} \hat{L}(\hat{W}, \hat{z}) \geq (2\tau_m - 1)v(L_m^2) \geq (2\tau_m - 1)v(L_m^3). \quad (1)$$

The key step is to recover (x, y) from \hat{W} . For this purpose, we introduce the following decomposition routine (DR).

DR (Decomposition Routine) 3.1

- *Input:* $\hat{W} \in \Omega_m^{n_1 \times n_2}$.
- *Construct*

$$\tilde{W} = \begin{bmatrix} I & \hat{W}/\sqrt{n_1} \\ \hat{W}^H/\sqrt{n_1} & \hat{W}^H \hat{W}/n_1 \end{bmatrix} \succeq 0 \quad (\text{Hermitian positive semidefinite}).$$

- *Randomly generate*

$$\begin{pmatrix} \xi \\ \eta \end{pmatrix} \sim \mathcal{N}(0, \tilde{W}).$$

- *For* $i = 1, 2, \dots, n_1$, *let*

$$\hat{x}_i := \omega_m^\ell \text{ if } \arg \xi_i \in \left[\frac{\ell}{m} 2\pi, \frac{\ell+1}{m} 2\pi \right) \text{ for some } \ell \in \mathbb{Z};$$

and for $j = 1, 2, \dots, n_2$, *let*

$$\hat{y}_j := \omega_m^{-\ell} \text{ if } \arg \eta_j \in \left[\frac{\ell}{m} 2\pi, \frac{\ell+1}{m} 2\pi \right) \text{ for some } \ell \in \mathbb{Z}.$$

- *Output:* $(\hat{x}, \hat{y}) \in \Omega_m^{n_1+n_2}$.

It was shown in [29] that

$$\mathbb{E}[\hat{x}_i \hat{y}_j] = \frac{m(2 - \omega_m - \omega_m^{-1})}{8\pi^2} \sum_{\ell=0}^{m-1} \omega_m^\ell \left(\arccos \left(-\operatorname{Re} \omega_m^{-\ell} \tilde{W}_{i, n_1+j} \right) \right)^2. \quad (2)$$

There are some useful properties regarding (2) as shown below; the proofs can be found in the appendix.

Lemma 3.2 *Define $F_m : \mathbb{C} \mapsto \mathbb{C}$ with $F_m(x) := \frac{m(2 - \omega_m - \omega_m^{-1})}{8\pi^2} \sum_{\ell=0}^{m-1} \omega_m^\ell \left(\arccos \left(-\operatorname{Re} \omega_m^{-\ell} x \right) \right)^2$.*
(1) *If $a \in \mathbb{C}$ and $b \in \Omega_m$, then $F_m(ab) = bF_m(a)$.*
(2) *If $a \in \mathbb{R}$, then $F_m(a) \in \mathbb{R}$.*

As (\hat{W}, \hat{z}) is a feasible solution of (L_m^2) , $\hat{W}_{ij} \in \Omega_m$. By Lemma 3.2, we have for all (i, j)

$$\mathbb{E}[\hat{x}_i \hat{y}_j] = F_m(\tilde{W}_{i, n_1+j}) = F_m(\hat{W}_{ij}/\sqrt{n_1}) = \hat{W}_{ij} F_m(1/\sqrt{n_1}) \text{ and } F_m(1/\sqrt{n_1}) \in \mathbb{R}. \quad (3)$$

We are now able to evaluate the objective value of $(\hat{x}, \hat{y}, \hat{z})$:

$$\begin{aligned} \mathbb{E}[\operatorname{Re} L(\hat{x}, \hat{y}, \hat{z})] &= \mathbb{E} \left[\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \operatorname{Re} \mathcal{F}_{ijk} \hat{x}_i \hat{y}_j \hat{z}_k \right] \\ &= \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \operatorname{Re} \mathcal{F}_{ijk} \mathbb{E}[\hat{x}_i \hat{y}_j] \hat{z}_k \\ &= \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \operatorname{Re} \mathcal{F}_{ijk} \hat{W}_{ij} F_m(1/\sqrt{n_1}) \hat{z}_k \\ &= F_m(1/\sqrt{n_1}) \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \operatorname{Re} \mathcal{F}_{ijk} \hat{W}_{ij} \hat{z}_k \\ &= F_m(1/\sqrt{n_1}) \operatorname{Re} \hat{L}(\hat{W}, \hat{z}). \end{aligned}$$

Furthermore, according to the appendix of [29], we have

$$F_m(1/\sqrt{n_1}) \geq \frac{m^2(1 - \cos \frac{2\pi}{m})}{8\pi\sqrt{n_1}} = \frac{\tau_m}{\sqrt{n_1}}. \quad (4)$$

Combined with (1), we finally get

$$\mathbb{E}[\operatorname{Re} L(\hat{x}, \hat{y}, \hat{z})] = F_m(1/\sqrt{n_1}) \operatorname{Re} \hat{L}(\hat{W}, \hat{z}) \geq \frac{\tau_m}{\sqrt{n_1}} (2\tau_m - 1) v(L_m^3).$$

Theorem 3.3 *When $d = 3$, (L_m) admits a polynomial-time randomized approximation algorithm with approximation ratio $\frac{\tau_m(2\tau_m-1)}{\sqrt{n_1}}$.*

By a similar method and using induction, the above discussion is readily extended to any fixed degree d .

Theorem 3.4 (L_m) admits a polynomial-time randomized approximation algorithm with approximation ratio $\tau(L_m) := \tau_m^{d-2} (2\tau_m - 1) \left(\prod_{k=1}^{d-2} n_k \right)^{-\frac{1}{2}}$, i.e., a feasible solution $(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d)$ can be found in polynomial-time, such that

$$\mathbb{E} \left[\text{Re } L(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d) \right] \geq \tau(L_m) v(L_m).$$

Proof. The proof is based on induction on the degree d . The conclusion for the case where $d = 2$ or $d = 3$ is known to be true. The inductive step can be similarly derived from Theorem 3.3.

For general d , denote $W = x^1 (x^d)^\top$ and (L_m) is then relaxed to

$$\begin{aligned} (L_m^{d-1}) \quad & \max \quad \text{Re } \hat{L}(W, x^2, \dots, x^{d-1}) := \text{Re} \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} \mathcal{F}_{i_1 i_2 \dots i_d} W_{i_1 i_d} x_{i_2}^2 \cdots x_{i_{d-1}}^{d-1} \\ \text{s.t.} \quad & W \in \Omega_m^{n_1 \times n_d}, x^k \in \Omega_m^{n_k}, k = 2, 3, \dots, d-1. \end{aligned}$$

By induction we are able to find $(\hat{W}, \hat{x}^2, \dots, \hat{x}^{d-1})$, such that

$$\begin{aligned} \mathbb{E} \left[\text{Re } \hat{L}(\hat{W}, \hat{x}^2, \dots, \hat{x}^{d-1}) \right] & \geq \tau_m^{d-3} (2\tau_m - 1) \left(\prod_{k=2}^{d-2} n_k \right)^{-\frac{1}{2}} v(L_m^{d-1}) \\ & \geq \tau_m^{d-3} (2\tau_m - 1) \left(\prod_{k=2}^{d-2} n_k \right)^{-\frac{1}{2}} v(L_m). \end{aligned}$$

Applying DR 3.1 with input \hat{W} and output (\hat{x}^1, \hat{x}^d) , and using (3) and (4), we conclude that

$$\begin{aligned} \mathbb{E} \left[\text{Re } L(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d) \right] & = \mathbb{E} \left[\text{Re } \hat{L} \left(\hat{x}^1 (\hat{x}^d)^\top, \hat{x}^2, \dots, \hat{x}^{d-1} \right) \right] \\ & = \mathbb{E} \left[\text{Re } \hat{L} \left(\mathbb{E} \left[\hat{x}^1 (\hat{x}^d)^\top | \hat{W} \right], \hat{x}^2, \dots, \hat{x}^{d-1} \right) \right] \\ & = \mathbb{E} \left[\text{Re } \hat{L} \left(\hat{W} F_m (1/\sqrt{n_1}), \hat{x}^2, \dots, \hat{x}^{d-1} \right) \right] \\ & = F_m (1/\sqrt{n_1}) \mathbb{E} \left[\text{Re } \hat{L}(\hat{W}, \hat{x}^2, \dots, \hat{x}^{d-1}) \right] \\ & \geq \frac{\tau_m}{\sqrt{n_1}} \cdot \tau_m^{d-3} (2\tau_m - 1) \left(\prod_{k=2}^{d-2} n_k \right)^{-\frac{1}{2}} v(L_m) \\ & = \tau(L_m) v(L_m). \end{aligned}$$

□

3.2 Multilinear form with unity constraints

Let us now turn to the optimization model with unity constraint (L_∞) , which can be taken as the model (L_m) when $m \rightarrow \infty$:

$$\begin{aligned} (L_\infty) \quad & \max \quad \text{Re } L(x^1, x^2, \dots, x^d) \\ \text{s.t.} \quad & x^k \in \Omega_\infty^{n_k}, k = 1, 2, \dots, d. \end{aligned}$$

When $d = 2$, (L_∞) was studied in [15] and a polynomial-time approximation algorithm with approximation ratio 0.7118 was presented. To treat the high degree case, one may again apply induction in the proof of Theorem 3.4. However, DR 3.1 should be slightly modified in order to apply the decomposition procedure for Ω_∞ .

DR (Decomposition Routine) 3.5

- *Input:* $\hat{W} \in \Omega_\infty^{n_1 \times n_2}$.
 - *Construct* $\tilde{W} = \begin{bmatrix} I & \hat{W}/\sqrt{n_1} \\ \hat{W}^H/\sqrt{n_1} & \hat{W}^H\hat{W}/n_1 \end{bmatrix} \succeq 0$.
 - *Randomly generate* $(\xi_\eta) \sim \mathcal{N}(0, \tilde{W})$.
 - *Let* $\hat{x}_i = e^{i \arg \xi_i}$ for $i = 1, 2, \dots, n_1$, and let $\hat{y}_j = e^{-i \arg \eta_j}$ for $j = 1, 2, \dots, n_2$.
 - *Output:* $(\hat{x}, \hat{y}) \in \Omega_\infty^{n_1+n_2}$.
-

The estimation of (\hat{x}, \hat{y}) is then

$$\mathbb{E}[\hat{x}_i \hat{y}_j] = F_\infty(\tilde{W}_{i, n_1+j}) = F_\infty(\hat{W}_{ij}/\sqrt{n_1}) \quad \forall (i, j).$$

It was calculated in [29] that

$$F_\infty(a) := \lim_{m \rightarrow \infty} F_m(a) = \frac{\pi}{4}a + \frac{\pi}{2} \sum_{k=1}^{\infty} \frac{((2k)!)^2}{2^{4k+1}(k!)^4(k+1)} |a|^{2k} a.$$

Similar as in Lemma 3.2:

$$\begin{aligned} F_\infty(ab) &= bF_\infty(a) \quad \forall a \in \mathbb{C}, b \in \Omega_\infty, \\ F_\infty(a) &\in \mathbb{R} \quad \forall a \in \mathbb{R}, \\ F_\infty(a) &\geq \frac{\pi}{4}a \quad \forall a > 0. \end{aligned}$$

By applying the result in [15] for case $d = 2$ and using a similar argument as Theorem 3.4, we have the following main result of this subsection.

Theorem 3.6 (L_∞) *admits a polynomial-time randomized approximation algorithm with approximation ratio* $\tau(L_\infty) := 0.7118 \left(\frac{\pi}{4}\right)^{d-2} \left(\prod_{k=1}^{d-2} n_k\right)^{-\frac{1}{2}}$.

3.3 Multilinear form with spherical constraints

Let us turn to our last model for multilinear form optimization:

$$(L_S) \quad \begin{aligned} \max \quad & \operatorname{Re} L(x^1, x^2, \dots, x^d) \\ \text{s.t.} \quad & x^k \in \mathbf{S}^{n_k}, k = 1, 2, \dots, d. \end{aligned}$$

Model (L_S) is also known as computing the largest singular value (the real part) of a d -th order complex tensor \mathcal{F} . The case when \mathcal{F} is real was widely studied [10, 26, 6, 20]. In particular, He et al. [10] introduced the recursive procedure and eigen-decomposition based approximation algorithm with approximation ratio $\left(\prod_{k=1}^{d-2} n_k\right)^{-\frac{1}{2}}$. Using a similar argument, we have the following result.

Theorem 3.7 (L_S) *admits a deterministic polynomial-time approximation algorithm with approximation ratio* $\tau(L_S) := \left(\prod_{k=1}^{d-2} n_k\right)^{-\frac{1}{2}}$.

When $d = 2$, (L_S) is to compute the largest singular value of a complex matrix, and is therefore solvable in polynomial-time, which also follows as a consequence of Theorem 3.7. The proof of Theorem 3.7 is similar to that of [10] for the real case. The main ingredients include establishing the initial step for the case $d = 2$, and then establishing a decomposition routine, which is shown as follows, to enable the induction.

DR (Decomposition Routine) 3.8

-
- *Input:* $\hat{W} \in \mathbb{C}^{n_1 \times n_2}$.
 - *Find the left singular vector $\hat{x} \in \mathbf{S}^{n_1}$ and the right singular vector $\hat{y} \in \mathbf{S}^{n_2}$ corresponding to the largest singular value of \hat{W} .*
 - *Output:* $\hat{x} \in \mathbf{S}^{n_1}, \hat{y} \in \mathbf{S}^{n_2}$.
-

4 Complex homogeneous polynomial optimization

This section is concerned with the optimization of complex homogeneous polynomial $H(x)$, associated with super-symmetric complex tensor $\mathcal{F} \in \mathbb{C}^{n^d}$. Specifically, the models under considerations are:

$$\begin{aligned}
 (H_m) \quad & \max \quad \operatorname{Re} H(x) \\
 & \text{s.t.} \quad x \in \Omega_m^n; \\
 (H_\infty) \quad & \max \quad \operatorname{Re} H(x) \\
 & \text{s.t.} \quad x \in \Omega_\infty^n; \\
 (H_S) \quad & \max \quad \operatorname{Re} H(x) \\
 & \text{s.t.} \quad x \in \mathbf{S}^n.
 \end{aligned}$$

Denote L to be the multilinear form associated with \mathcal{F} , and then $H(x) = L(\underbrace{x, x, \dots, x}_d)$. By

applying the tensor relaxation method established in [10], the above models are then relaxed to the following multilinear form optimization models discussed in Section 3:

$$\begin{aligned}
 (LH_m) \quad & \max \quad \operatorname{Re} L(x^1, x^2, \dots, x^d) \\
 & \text{s.t.} \quad x^k \in \Omega_m^n, k = 1, 2, \dots, d; \\
 (LH_\infty) \quad & \max \quad \operatorname{Re} L(x^1, x^2, \dots, x^d) \\
 & \text{s.t.} \quad x^k \in \Omega_\infty^n, k = 1, 2, \dots, d; \\
 (LH_S) \quad & \max \quad \operatorname{Re} L(x^1, x^2, \dots, x^d) \\
 & \text{s.t.} \quad x^k \in \mathbf{S}^n, k = 1, 2, \dots, d.
 \end{aligned}$$

The approximation results in Section 3 can return good approximation solutions for these relaxed models. The key next step is to obtain good solutions for the original homogeneous polynomial optimizations. Similar to Lemma 2.1, we establish a linkage between functions L and H in the complex domain. The proof of Lemma 4.1 can be found in the appendix.

Lemma 4.1 *Let $m \in \{3, 4, \dots, \infty\}$. Suppose $x^1, x^2, \dots, x^d \in \mathbb{C}^n$, and $\mathcal{F} \in \mathbb{C}^{n^d}$ is a super-symmetric complex tensor with its associated multilinear form L and homogeneous polynomial H . If $\xi_1, \xi_2, \dots, \xi_d$ are i.i.d. uniform distribution on Ω_m , then*

$$\mathbb{E} \left[\prod_{i=1}^d \bar{\xi}_i H \left(\sum_{k=1}^d \xi_k x^k \right) \right] = d! L(x^1, x^2, \dots, x^d) \quad \text{and} \quad \mathbb{E} \left[\prod_{i=1}^d \xi_i H \left(\sum_{k=1}^d \xi_k x^k \right) \right] = 0.$$

4.1 Homogeneous polynomial in the m -th roots of unity

Let us now focus on the model $(H_m) : \max_{x \in \Omega_m^n} \operatorname{Re} H(x)$. By Lemma 4.1, for any fixed $\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d \in \mathbb{C}^n$, we can find $\beta_1, \beta_2, \dots, \beta_d \in \Omega_m$ in polynomial-time, such that

$$\operatorname{Re} \prod_{i=1}^d \overline{\beta_i} H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right) \geq \operatorname{Re} d^{-d} d! L(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d). \quad (5)$$

For any $1 \leq i \leq n$, if $\hat{x}_i^k \in \Omega_m$ for all $1 \leq k \leq d$, then $\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}_i^k \in \operatorname{conv}(\Omega_m)$. As shown below, we are able to get a solution from $\operatorname{conv}(\Omega_m)$ to one of its vertices (Ω_m) .

Lemma 4.2 *Suppose $m \in \{3, 4, \dots, \infty\}$, and $x \in \mathbb{C}^n$ with $x_i \in \operatorname{conv}(\Omega_m)$ for all $1 \leq i \leq n$.*

(1) *If $H(x)$ is a complex homogeneous polynomial associated with square-free (meaning that its entry is zero whenever two of its indices are identical) super-symmetric tensor $\mathcal{F} \in \mathbb{C}^{n^d}$, then $y, z \in \Omega_m^n$ can be found in polynomial-time, such that $\operatorname{Re} H(y) \leq \operatorname{Re} H(x) \leq \operatorname{Re} H(z)$.*

(2) *If $\operatorname{Re} H(x)$ is convex, then $z \in \Omega_m^n$ can be found in polynomial-time, such that $\operatorname{Re} H(x) \leq \operatorname{Re} H(z)$.*

Proof. If $H(x)$ is square-free, by fixing x_2, x_3, \dots, x_n as constants and taking x_1 as the only decision variable, we may write

$$\operatorname{Re} H(x) = \operatorname{Re} h_1(x_2, x_3, \dots, x_n) + \operatorname{Re} x_1 h_2(x_2, x_3, \dots, x_n) =: \operatorname{Re} h(x_1).$$

Since $\operatorname{Re} h(x_1)$ is a linear function of x_1 , its optimal value over $\operatorname{conv}(\Omega_m)$ is attained at one of its vertices. For instance, $z_1 \in \Omega_m$ can be found easily such that $\operatorname{Re} h(z_1) \geq \operatorname{Re} h(x_1)$. Now, repeat the same procedures for x_2, x_3, \dots, x_n , and let them be replaced by z_2, z_3, \dots, z_n respectively. Then $z \in \Omega_m^n$ satisfies $\operatorname{Re} H(z) \geq \operatorname{Re} H(x)$. Using the same argument, we may find $y \in \Omega_m^n$, such that $\operatorname{Re} H(y) \leq \operatorname{Re} H(x)$. The case that $\operatorname{Re} H(x)$ is convex can be proven similarly. \square

Now we are ready to prove the main results in this subsection.

Theorem 4.3 *Suppose $H(x)$ is square-free or $\operatorname{Re} H(x)$ is convex.*

(1) *If $m \mid (d-1)$, then (H_m) admits a polynomial-time randomized approximation algorithm with approximation ratio $\tau(H_m) := \tau_m^{d-2} (2\tau_m - 1) d! d^{-d} n^{-\frac{d-2}{2}}$.*

(2) *If $m \nmid 2d$, then (H_m) admits a polynomial-time randomized approximation algorithm with approximation ratio $\frac{1}{2}\tau(H_m)$.*

Proof. Relaxing (H_m) to (LH_m) , we find a feasible solution $(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d)$ of (LH_m) in polynomial-time with approximation ratio $\tau_m^{d-2} (2\tau_m - 1) n^{-\frac{d-2}{2}}$ by Theorem 3.4. Then by (5), we further find $\beta \in \Omega_m^d$, such that

$$\operatorname{Re} \prod_{i=1}^d \overline{\beta_i} H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right) \geq \operatorname{Re} d! d^{-d} L(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d) \geq \tau(H_m) v(LH_m) \geq \tau(H_m) v(H_m).$$

Let us denote $\hat{x} := \frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k$. Clearly we have $\hat{x}_i \in \operatorname{conv}(\Omega_m)$ for $i = 1, 2, \dots, n$.

(1) If $m \mid (d-1)$, then $d = 1 + mp$ for some $p \in \mathbb{Z}$. As $\beta_i \in \Omega_m$, we have

$$H \left(\hat{x} \prod_{i=1}^d \overline{\beta_i} \right) = \left(\prod_{i=1}^d \overline{\beta_i} \right)^d H(\hat{x}) = \prod_{i=1}^d \overline{\beta_i}^{1+mp} H(\hat{x}) = \prod_{i=1}^d \overline{\beta_i} H(\hat{x}).$$

Since $\hat{x}_j \prod_{i=1}^d \bar{\beta}_i \in \text{conv}(\mathbf{\Omega}_m)$ for $j = 1, 2, \dots, n$, noticing $H(x)$ is square-free or $\text{Re } H(x)$ is convex, and applying Lemma 4.2, we are able to find $y \in \mathbf{\Omega}_m^n$ in polynomial-time, such that

$$\text{Re } H(y) \geq \text{Re } H\left(\hat{x} \prod_{i=1}^d \bar{\beta}_i\right) = \text{Re} \prod_{i=1}^d \bar{\beta}_i H(\hat{x}) \geq \tau(H_m)v(H_m).$$

(2) Let $\Phi = \{H(\omega_m^\ell \hat{x}) \mid \ell = 0, 1, \dots, m-1\}$. As $H(\omega_m^\ell \hat{x}) = \omega_m^{d\ell} H(\hat{x})$ for $\ell = 0, 1, \dots, m-1$, the elements of Φ is evenly distributed on the unity circle with radius $|H(\hat{x})|$ in the complex plane. Since $\omega_m^{d\ell} = e^{i \frac{2d\ell\pi}{m}}$ and $m \nmid 2d$, it is easy to verify that $|\Phi| \geq 3$. Let ϕ be the minimum angle between Φ and the real axis, or equivalently $|H(\hat{x})| \cos \phi = \max_{x \in \Phi} \text{Re } x$. Clearly $0 \leq \phi \leq \frac{\pi}{3}$ by $|\Phi| \geq 3$. Let $H(\omega_m^t \hat{x}) = \arg \max_{x \in \Phi} \text{Re } x$. As $\omega_m^t \hat{x}_j \in \text{conv}(\mathbf{\Omega}_m)$ for $j = 1, 2, \dots, n$, again by Lemma 4.2, we are able to find $y \in \mathbf{\Omega}_m^n$ in polynomial-time, such that

$$\text{Re } H(y) \geq \text{Re } H(\omega_m^t \hat{x}) = |H(\hat{x})| \cos \phi \geq \frac{1}{2}|H(\hat{x})| \geq \frac{1}{2} \text{Re} \prod_{i=1}^d \bar{\beta}_i H(\hat{x}) \geq \frac{1}{2} \tau(H_m)v(H_m).$$

□

Remark that condition (1) in Theorem 4.3 is a special case of (2); however in that special case a better approximation ratio than (2) is obtained. When $d \geq 4$ is even, almost all of the optimization models of homogeneous polynomials in the real domain (e.g. [10, 12, 26, 20]) only admit *relative* approximation ratios. Even for quartic polynomial optimization over spherical constraints, there is no polynomial-time approximation algorithm with a usual approximation ratio [21]. Interestingly, in the complex domain, as Theorem 4.3 suggests, absolute approximation ratios are possible for some m when d is even.

When $m \mid 2d$, the approach in (2) of Theorem 4.3 may not work, since $|\Phi| \leq 2$. The worst case performance of the approximate solution cannot be guaranteed any more. However a relative approximation bound is possible for any m , as long as $H(x)$ is square-free.

Theorem 4.4 *If $H(x)$ is square-free, then (H_m) admits a polynomial-time randomized approximation algorithm with relative approximation ratio $\frac{1}{4}\tau(H_m)$.*

Proof. Relaxing (H_m) to (LH_m) , we may find a feasible solution $(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d)$ of (LH_m) in polynomial-time with approximation ratio $\tau_m^{d-2} (2\tau_m - 1) n^{-\frac{d-2}{2}}$ by Theorem 3.4, such that

$$d!d^{-d} \text{Re } L(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d) \geq d!d^{-d} \tau_m^{d-2} (2\tau_m - 1) n^{-\frac{d-2}{2}} v(LH_m) = \tau(H_m)v(LH_m) \geq \tau(H_m)v(H_m).$$

Let $\xi_1, \xi_2, \dots, \xi_d$ be i.i.d. uniform distribution on $\mathbf{\Omega}_m$, and we have $\frac{1}{d} \sum_{k=1}^d \xi_k \hat{x}_i^k \in \text{conv}(\mathbf{\Omega}_m)$ for $i = 1, 2, \dots, n$. As $H(x)$ is square-free, by Lemma 4.2, there exists $y \in \mathbf{\Omega}_m^n$, such that

$$\text{Re } H\left(\frac{1}{d} \sum_{k=1}^d \xi_k \hat{x}^k\right) \geq \text{Re } H(y) \geq \underline{v}(H_m). \quad (6)$$

According to Lemma 4.1, it follows that

$$\mathbb{E} \left[\text{Re} \prod_{i=1}^d \bar{\xi}_i H\left(\sum_{k=1}^d \xi_k \hat{x}^k\right) \right] = \text{Re } d!L(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d) \text{ and } \mathbb{E} \left[\text{Re} \prod_{i=1}^d \xi_i H\left(\sum_{k=1}^d \xi_k \hat{x}^k\right) \right] = 0.$$

Combining the above two identities leads to

$$\begin{aligned}
\operatorname{Re} d!d^{-d}L(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d) &= \mathbb{E} \left[\operatorname{Re} \prod_{i=1}^d \bar{\xi}_i H \left(\frac{1}{d} \sum_{k=1}^m \xi_k \hat{x}^k \right) \right] + \mathbb{E} \left[\operatorname{Re} \prod_{i=1}^d \xi_i H \left(\frac{1}{d} \sum_{k=1}^m \xi_k \hat{x}^k \right) \right] \\
&= \mathbb{E} \left[\operatorname{Re} \left(\prod_{i=1}^d \bar{\xi}_i + \prod_{i=1}^d \xi_i \right) H \left(\frac{1}{d} \sum_{k=1}^d \xi_k \hat{x}^k \right) \right] \\
&= \mathbb{E} \left[\left(\prod_{i=1}^d \bar{\xi}_i + \prod_{i=1}^d \xi_i \right) \operatorname{Re} H \left(\frac{1}{d} \sum_{k=1}^d \xi_k \hat{x}^k \right) \right] \\
&= \mathbb{E} \left[\left(\prod_{i=1}^d \bar{\xi}_i + \prod_{i=1}^d \xi_i \right) \left(\operatorname{Re} H \left(\frac{1}{d} \sum_{k=1}^d \xi_k \hat{x}^k \right) - \underline{v}(H_m) \right) \right] \\
&\leq \mathbb{E} \left[\left| \prod_{i=1}^d \bar{\xi}_i + \prod_{i=1}^d \xi_i \right| \cdot \left| \operatorname{Re} H \left(\frac{1}{d} \sum_{k=1}^d \xi_k \hat{x}^k \right) - \underline{v}(H_m) \right| \right] \\
&\leq 2 \mathbb{E} \left[\operatorname{Re} H \left(\frac{1}{d} \sum_{k=1}^d \xi_k \hat{x}^k \right) - \underline{v}(H_m) \right],
\end{aligned}$$

where the fourth step is due to the fact that $\xi_1, \xi_2, \dots, \xi_d$ are i.i.d. and $\mathbb{E}[\xi_i] = 0$ for $i = 1, 2, \dots, d$, and the last step is due to (6). By randomizing, we are able to find $\beta \in \Omega_m^d$, such that

$$\operatorname{Re} H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right) - \underline{v}(H_m) \geq \frac{1}{2} \operatorname{Re} d!d^{-d}L(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d) \geq \frac{1}{2} \tau(H_m) v(H_m).$$

Let us now separately discuss two cases. In the first case, if $v(H_m) \geq \frac{1}{2} (v(H_m) - \underline{v}(H_m))$, then the above further leads to

$$\operatorname{Re} H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right) - \underline{v}(H_m) \geq \frac{1}{2} \tau(H_m) v(H_m) \geq \frac{1}{4} \tau(H_m) (v(H_m) - \underline{v}(H_m)).$$

Otherwise, we have $v(H_m) < \frac{1}{2} (v(H_m) - \underline{v}(H_m))$, which implies $-\underline{v}(H_m) > \frac{1}{2} (v(H_m) - \underline{v}(H_m))$, and this leads to

$$\operatorname{Re} H(0) - \underline{v}(H_m) = 0 - \underline{v}(H_m) > \frac{1}{2} (v(H_m) - \underline{v}(H_m)) \geq \frac{1}{4} \tau(H_m) (v(H_m) - \underline{v}(H_m)).$$

Combing these two cases, we shall uniformly get $\hat{x} = \arg \max \left\{ \operatorname{Re} H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right), \operatorname{Re} H(0) \right\}$ satisfying $\operatorname{Re} H(\hat{x}) - \underline{v}(H_m) \geq \frac{1}{4} \tau(H_m) (v(H_m) - \underline{v}(H_m))$. Finally, by noticing $\hat{x}_i \in \operatorname{conv}(\Omega_m)$ for $i = 1, 2, \dots, n$ and $H(x)$ is square-free, and applying Lemma 4.2, we are able to find $z \in \Omega_m^n$ in polynomial-time, such that

$$\operatorname{Re} H(z) - \underline{v}(H_m) \geq \operatorname{Re} H(\hat{x}) - \underline{v}(H_m) \geq \frac{1}{4} \tau(H_m) (v(H_m) - \underline{v}(H_m)).$$

□

Before concluding this subsection, we remark that (H_m) can be equivalently transferred to polynomial optimization over discrete variables in the real case, which was discussed in [12]. Essentially,

by letting $x = y + iz$ with $y, z \in \mathbb{R}^n$, $\operatorname{Re} H(x)$ can be rewritten as a homogeneous polynomial of (y, z) , where for each $i = 1, 2, \dots, n$, $(y_i, z_i) = (\cos \frac{2k\pi}{m}, \sin \frac{2k\pi}{m})$ for some $k \in \mathbb{Z}$. By applying the Lagrange polynomial interpolation technique, the problem can then be transferred to an inhomogeneous polynomial optimization with binary constraints, which will yield a worst case relative approximation ratio as well. However, comparing to the bounds obtained in Theorem 4.4, the direct transformation to the real case is much worse and more costly to implement.

4.2 Homogeneous polynomial with unity constraints

Let us now turn to the case $m \rightarrow \infty$. In that case, (H_m) becomes

$$(H_\infty) \quad \max \quad \operatorname{Re} H(x) \\ \text{s.t.} \quad x \in \Omega_\infty^n.$$

It is not hard to verify (see the proof of Theorem 4.5) that (H_∞) is actually equivalent to

$$\max \quad |H(x)| \\ \text{s.t.} \quad x \in \Omega_\infty^n.$$

For the case $d = 2$, the above problem was studied by Tokar and Ozbay [28], and was termed complex programming. Unlike the case of the m -th roots of unity, where certain conditions on m and d are required to secure approximation ratios, model (H_∞) actually always admits a polynomial-time approximation ratio for any fixed d .

Theorem 4.5 *If $H(x)$ is square-free or $\operatorname{Re} H(x)$ is convex, then (H_∞) admits a polynomial-time randomized approximation algorithm with approximation ratio $\tau(H_\infty) := 0.7118(\frac{\pi}{4})^{d-2} d! d^{-d} n^{-\frac{d-2}{2}}$.*

Proof. Relaxing (H_∞) to (LH_∞) , we may find a feasible solution $(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d)$ of (LH_∞) in polynomial-time with approximation ratio $0.7118(\frac{\pi}{4})^{d-2} n^{-\frac{d-2}{2}}$ by Theorem 3.6, i.e.,

$$\operatorname{Re} L(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d) \geq 0.7118 \left(\frac{\pi}{4}\right)^{d-2} n^{-\frac{d-2}{2}} v(LH_\infty).$$

Then by Lemma 4.1, we further find $\beta \in \Omega_\infty^d$ by randomization, such that

$$\operatorname{Re} \prod_{i=1}^d \bar{\beta}_i H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right) \geq \operatorname{Re} d^{-d} d! L(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d) \geq \tau(H_\infty) v(LH_\infty) \geq \tau(H_\infty) v(H_\infty).$$

Let $\phi = \arg H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right)$, and we get

$$H \left(\frac{e^{-i\phi/d}}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right) = e^{-i\phi} H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right) = \left| H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right) \right| \geq \operatorname{Re} \prod_{i=1}^d \bar{\beta}_i H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right).$$

Finally, by noticing that each component of $\frac{e^{-i\phi/d}}{d} \sum_{k=1}^d \beta_k \hat{x}^k$ is in $\operatorname{conv}(\Omega_\infty)$, and applying Lemma 4.2, we are able to find $y \in \Omega_\infty^n$ in polynomial-time, such that

$$\operatorname{Re} H(y) \geq \operatorname{Re} H \left(\frac{e^{-i\phi/d}}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right) \geq \operatorname{Re} \prod_{i=1}^d \bar{\beta}_i H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right) \geq \tau(H_\infty) v(H_\infty).$$

□

4.3 Homogeneous polynomial with spherical constraint

Our last model in this section is spherical constrained homogeneous polynomial optimization in the complex domain

$$(H_S) \quad \max \quad \operatorname{Re} H(x) \\ \text{s.t.} \quad x \in \mathbf{S}^n.$$

The model is equivalent to $\max_{x \in \mathbf{S}^n} |H(x)|$, which is also equivalent to computing the largest eigenvalue of a super-symmetric complex tensor $\mathcal{F} \in \mathbb{C}^{n^d}$.

The real counterpart of (H_S) is studied in the literature; see [10, 26, 20]. The problem is related to computing the largest Z -eigenvalue of a super-symmetric tensor, or equivalently, finding the best rank-one approximation of a super-symmetric tensor [6, 30]. Again, in principle, the complex case can be transformed to the real case by letting $x = y + iz$ with $y, z \in \mathbb{R}^n$, which however increases the number of the variables as well as the dimension of the data tensor \mathcal{F} . As a result, this will cause a deterioration in the approximation quality. Moreover, in the real case, (H_S) only admits a *relative* approximation ratio when d is even. Interestingly, for any fixed d , an absolute approximation ratio is possible for the complex case.

Theorem 4.6 (H_S) admits a deterministic polynomial-time approximation algorithm with approximation ratio $\tau(H_S) := d!d^{-d}n^{-\frac{d-2}{2}}$.

Proof. Like in the proof of Theorem 4.5, by relaxing (H_S) to (LH_S) , we first find a feasible solution $(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d)$ of (LH_S) with approximation ratio $n^{-\frac{d-2}{2}}$ (Theorem 3.7). Then by Lemma 4.1, we further find $\beta \in \Omega_\infty^d$, such that

$$\operatorname{Re} \prod_{i=1}^d \bar{\beta}_i H \left(\frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k \right) \geq \operatorname{Re} d^{-d} d! L(\hat{x}^1, \hat{x}^2, \dots, \hat{x}^d) \geq \tau(H_S) v(LH_S) \geq \tau(H_S) v(H_S).$$

Let $\hat{x} = \frac{1}{d} \sum_{k=1}^d \beta_k \hat{x}^k$ and $\phi = \arg H(\hat{x})$. By triangle inequality we have $\|\hat{x}\| \leq \frac{1}{d} \sum_{k=1}^d \|\beta_k \hat{x}^k\| = 1$. Finally, $e^{-i\phi/d} \hat{x} / \|\hat{x}\|$ is a feasible solution of (H_S) , satisfying

$$H \left(e^{-i\phi/d} \frac{\hat{x}}{\|\hat{x}\|} \right) = e^{-i\phi} \|\hat{x}\|^{-d} H(\hat{x}) = \|\hat{x}\|^{-d} |H(\hat{x})| \geq |H(\hat{x})| \geq \operatorname{Re} \prod_{i=1}^d \bar{\beta}_i H(\hat{x}) \geq \tau(H_S) v(H_S).$$

□

We remark that the above result does not require $H(x)$ to be square-free or $\operatorname{Re} H(x)$ to be convex, which is a condition for Theorems 4.3 and 4.5.

5 Conjugate form optimization

Our last set of optimization models involve the so-called symmetric conjugate forms:

$$(C_m) \quad \max \quad C(\bar{x}, x) \\ \text{s.t.} \quad x \in \Omega_m^n; \\ (C_\infty) \quad \max \quad C(\bar{x}, x) \\ \text{s.t.} \quad x \in \Omega_\infty^n; \\ (C_S) \quad \max \quad C(\bar{x}, x) \\ \text{s.t.} \quad x \in \mathbf{S}^n.$$

Recall that the symmetric conjugate form $C(\bar{x}, x) = L(\underbrace{\bar{x}, \dots, \bar{x}}_d, \underbrace{x, \dots, x}_d)$ is associated with a conjugate partial-symmetric tensor $\mathcal{F} \in \mathbb{C}^{n^{2d}}$ (cf. Section 2 for details).

These models are known to have wide applications as well. For instance, (C_m) and (C_∞) with degree 4 are used in the design of radar waveforms [4] sharing an ambiguity function. (C_∞) includes (H_∞) as its special case, since (H_∞) is equivalent to $\max_{x \in \Omega_\infty^n} |H(x)|$, where $|H(x)|^2$ is a special class for $C(\bar{x}, x)$. Therefore, complex programming $((H_\infty)$ with $d = 2$) studied by Toker and Ozbay [28] also belongs to (C_∞) . Similarly, (C_S) also includes (H_S) as its special case.

Let us now focus on approximation algorithms. Observe that for any conjugate partial-symmetric tensor \mathcal{F} with its associated symmetric conjugate form $C(\bar{x}, x)$:

$$C(\bar{x}, x) = \operatorname{Re} L(x^1, \dots, x^d, x^{d+1}, \dots, x^{2d}) \text{ when } x^1 = \dots = x^d = \bar{x} \text{ and } x^{d+1} = \dots = x^{2d} = x.$$

Therefore, (C_m) , (C_∞) and (C_S) can be relaxed to the following multilinear optimization models:

$$\begin{aligned} (LC_m) \quad & \max \operatorname{Re} L(x^1, \dots, x^d, x^{d+1}, \dots, x^{2d}) \\ & \text{s.t. } x^k \in \Omega_m^n, k = 1, 2, \dots, 2d; \\ (LC_\infty) \quad & \max \operatorname{Re} L(x^1, \dots, x^d, x^{d+1}, \dots, x^{2d}) \\ & \text{s.t. } x^k \in \Omega_\infty^n, k = 1, 2, \dots, 2d; \\ (LC_S) \quad & \max \operatorname{Re} L(x^1, \dots, x^d, x^{d+1}, \dots, x^{2d}) \\ & \text{s.t. } x^k \in \mathbf{S}^n, k = 1, 2, \dots, 2d. \end{aligned}$$

By the approximation results established in Section 3, we are able to find good approximate solutions for these multilinear form optimization models. In order to generate good approximate solutions for the original conjugate form optimizations, we need the following new linkage between the symmetric conjugate form and the multilinear form.

Lemma 5.1 *Let $m \in \{3, 4, \dots, \infty\}$. Suppose $x^1, x^2, \dots, x^{2d} \in \mathbb{C}^n$, and $\mathcal{F} \in \mathbb{C}^{n^{2d}}$ is a conjugate partial-symmetric tensor with its associated multilinear form L and symmetric conjugate form C . If $\xi_1, \xi_2, \dots, \xi_{2d}$ are i.i.d. uniform distribution on Ω_m , then*

$$\mathbb{E} \left[\left(\prod_{i=1}^d \xi_i \right) \left(\prod_{i=d+1}^{2d} \bar{\xi}_i \right) C \left(\sum_{k=1}^d \bar{\xi}_k x^k + \sum_{k=d+1}^{2d} \bar{\xi}_k x^k, \sum_{k=1}^d \xi_k \bar{x}^k + \sum_{k=d+1}^{2d} \xi_k x^k \right) \right] = (d!)^2 L(x^1, x^2, \dots, x^{2d}).$$

The proof of Lemma 5.1 can be found in the appendix. By randomization we find $\beta \in \Omega_m^{2d}$ in polynomial-time, such that

$$\operatorname{Re} \left(\prod_{i=1}^d \beta_i \right) \left(\prod_{i=d+1}^{2d} \bar{\beta}_i \right) C(\bar{x}_\beta, x_\beta) \geq (d!)^2 (2d)^{-2d} \operatorname{Re} L(x^1, x^2, \dots, x^{2d}), \quad (7)$$

where

$$x_\beta := \frac{1}{2d} \sum_{k=1}^d \beta_k \bar{x}^k + \frac{1}{2d} \sum_{k=d+1}^{2d} \beta_k x^k. \quad (8)$$

5.1 Conjugate form in the m -th roots of unity

For (C_m) , by relaxing to (LC_m) and generating its approximate solution $(x^1, x^2, \dots, x^{2d})$ from Theorem 3.4, we know $x^k \in \Omega_m^n$ for $k = 1, 2, \dots, 2d$. Observe that each component of x_β defined by (8) is a convex combination of the elements in Ω_m , and is thus in $\operatorname{conv}(\Omega_m)$. Though x_β may not be feasible to (C_m) , a vertex solution (in Ω_m) can be found under certain conditions.

Lemma 5.2 Let $m \in \{3, 4, \dots, \infty\}$. Suppose $x \in \mathbb{C}^n$ with $x_i \in \text{conv}(\Omega_m)$ for all $1 \leq i \leq n$.

(1) If $C(\bar{x}, x)$ is a square-free symmetric conjugate form, then $y, z \in \Omega_m^n$ can be found in polynomial-time, such that $C(\bar{y}, y) \leq C(\bar{x}, x) \leq C(\bar{z}, z)$.

(2) If $C(\bar{x}, x)$ is convex, then $z \in \Omega_m^n$ can be found in polynomial-time, such that $C(\bar{x}, x) \leq C(\bar{z}, z)$.

The proof is similar to that of Lemma 4.2, and is thus omitted. Basically, the algorithm optimizes one variable x_i over Ω_m while fixing other $n - 1$ variables, alternatively for $i = 1, 2, \dots, n$. The condition of square-free or convexity guarantees that each step of optimization can be done in polynomial-time. With all these preparations in place, we are ready to present the first approximation result for symmetric conjugate form optimization.

Theorem 5.3 If $C(\bar{x}, x)$ is convex, then (C_m) admits a polynomial-time randomized approximation algorithm with approximation ratio $\tau(C_m) := \tau_m^{2d-2}(2\tau_m - 1)(d!)^2(2d)^{-2d}n^{-(d-1)}$.

Proof. By relaxing (C_m) to (LC_m) and getting its approximate solution $(x^1, x^2, \dots, x^{2d})$, we have

$$\text{Re } L(x^1, x^2, \dots, x^{2d}) \geq \tau_m^{2d-2}(2\tau_m - 1)n^{-(d-1)}v(LC_m) \geq \tau_m^{2d-2}(2\tau_m - 1)n^{-(d-1)}v(C_m). \quad (9)$$

Applying Lemma 5.1, we further get x_β defined by (8), satisfying (7), i.e.,

$$\text{Re} \left(\prod_{i=1}^d \beta_i \right) \left(\prod_{i=d+1}^{2d} \bar{\beta}_i \right) C(\bar{x}_\beta, x_\beta) \geq (d!)^2(2d)^{-2d} \text{Re } L(x^1, x^2, \dots, x^{2d}) \geq \tau(C_m)v(C_m).$$

Next it is easy to verify that any convex symmetric conjugate form is always nonnegative (see [4] for the proof in the quartic case), i.e., $C(\bar{x}, x) \geq 0$ for all $x \in \mathbb{C}^n$. This further leads to

$$C(\bar{x}_\beta, x_\beta) \geq \text{Re} \left(\prod_{i=1}^d \beta_i \right) \left(\prod_{i=d+1}^{2d} \bar{\beta}_i \right) C(\bar{x}_\beta, x_\beta) \geq \tau(C_m)v(C_m).$$

Finally, as each component of x_β belongs to $\text{conv}(\Omega_m)$, applying Lemma 5.2, we find $z \in \Omega_m^n$ with $C(\bar{z}, z) \geq C(\bar{x}_\beta, x_\beta) \geq \tau(C_m)v(C_m)$. \square

As seen from the proof in Theorem 5.3, the nonnegativity of convex symmetric conjugate form plays an essential role in preserving approximation guarantee. For the general case, this approximation is not possible, since a symmetric conjugate form may be negative definite. However under the square-free condition, relative approximation is doable.

Theorem 5.4 If $C(\bar{x}, x)$ is square-free, then (C_m) admits a polynomial-time randomized approximation algorithm with relative approximation ratio $\frac{1}{2}\tau(C_m)$.

Proof. The main structure of the proof is similar to that of Theorem 4.4, based on two complementary cases: $v(C_m) \geq \frac{1}{2}(v(C_m) - \underline{v}(C_m))$ and $-\underline{v}(C_m) > \frac{1}{2}(v(C_m) - \underline{v}(C_m))$. For the latter case, it is obvious that

$$C(\bar{0}, 0) - \underline{v}(C_m) = 0 - \underline{v}(C_m) \geq \frac{1}{2}(v(C_m) - \underline{v}(C_m)) \geq \frac{1}{2}\tau(C_m)(v(C_m) - \underline{v}(C_m)). \quad (10)$$

For the former case, we relax (C_m) to (LC_m) and get its approximate solution $(x^1, x^2, \dots, x^{2d})$. By (9) it follow that

$$\begin{aligned} (d!)^2(2d)^{-2d} \text{Re } L(x^1, x^2, \dots, x^{2d}) &\geq (d!)^2(2d)^{-2d} \tau_m^{2d-2}(2\tau_m - 1)n^{-(d-1)}v(C_m) \\ &\geq \frac{1}{2}\tau(C_m)(v(C_m) - \underline{v}(C_m)). \end{aligned} \quad (11)$$

Assume $\xi \in \Omega_m^{2d}$, whose components are i.i.d. uniform distribution on Ω_m . As each component of x_ξ defined by (8) belongs to $\text{conv}(\Omega_m)$, by Lemma 5.2, there exists $y \in \Omega_m^n$ such that

$$C(\bar{x}_\xi, x_\xi) \geq C(\bar{y}, y) \geq \underline{v}(C_m). \quad (12)$$

Applying Lemma 5.1, (11) further leads to

$$\begin{aligned} \frac{1}{2}\tau(C_m)(v(C_m) - \underline{v}(C_m)) &\leq (d!)^2(2d)^{-2d} \text{Re} L(x^1, x^2, \dots, x^{2d}) \\ &= \mathbb{E} \left[\text{Re} \left(\prod_{i=1}^d \xi_i \right) \left(\prod_{i=d+1}^{2d} \bar{\xi}_i \right) C(\bar{x}_\xi, x_\xi) \right] \\ &= \mathbb{E} \left[\text{Re} \left(\prod_{i=1}^d \xi_i \right) \left(\prod_{i=d+1}^{2d} \bar{\xi}_i \right) (C(\bar{x}_\xi, x_\xi) - \underline{v}(C_m)) \right] \\ &\leq \mathbb{E} \left[\left| \left(\prod_{i=1}^d \xi_i \right) \left(\prod_{i=d+1}^{2d} \bar{\xi}_i \right) \right| \cdot |C(\bar{x}_\xi, x_\xi) - \underline{v}(C_m)| \right] \\ &= \mathbb{E} [C(\bar{x}_\xi, x_\xi) - \underline{v}(C_m)], \end{aligned}$$

where the third step is due to $\mathbb{E} \left[\left(\prod_{i=1}^d \xi_i \right) \left(\prod_{i=d+1}^{2d} \bar{\xi}_i \right) \right] = 0$, and the last step is due to (12). Therefore by randomization, we are able to find $\beta \in \Omega_m^{2d}$, such that

$$C(\bar{x}_\beta, x_\beta) - \underline{v}(C_m) \geq \mathbb{E} [C(\bar{x}_\xi, x_\xi) - \underline{v}(C_m)] \geq \frac{1}{2}\tau(C_m)(v(C_m) - \underline{v}(C_m)).$$

Combining (10), if we let $x' = \arg \max \{C(\bar{0}, 0), C(\bar{x}_\beta, x_\beta)\}$, then we shall uniformly have $C(\bar{x}', x') - \underline{v}(C_m) \geq \frac{1}{2}\tau(C_m)(v(C_m) - \underline{v}(C_m))$. Finally, as each component of x' belongs to $\text{conv}(\Omega_m)$ and $C(\bar{x}, x)$ is square-free, by Lemma 5.2, we are able to find $z \in \Omega_m^n$ in polynomial-time, such that

$$C(\bar{z}, z) - \underline{v}(C_m) \geq C(\bar{x}', x') - \underline{v}(C_m) \geq \frac{1}{2}\tau(C_m)(v(C_m) - \underline{v}(C_m)).$$

□

5.2 Conjugate form with unity constraints or spherical constraint

The discussion in Section 5.1 can be extended to symmetric conjugate form optimization over unity constraints, and the complex spherical constraint: (C_∞) and (C_S) . Due to its similar nature, here we shall skip the details and only provide the main approximation results; the details can be easily supplemented by the interested reader. Essentially, the main steps are: (1) relax to multilinear form optimization models and find their approximate solutions as discussed in Section 3; (2) conduct randomization based on the link provided in Lemma 5.1; (3) search for the best vertex solution. For the complex unity constrained (C_∞) , a vertex solution is guaranteed by Lemma 5.2, and for the spherically constrained (C_S) , a vertex solution is obtained by scaling to \mathbf{S}^n : $x_\beta/\|x_\beta\|$.

Theorem 5.5 (1) If $C(\bar{x}, x)$ is convex, then (C_∞) admits a polynomial-time randomized approximation algorithm with approximation ratio $\tau(C_\infty) := 0.7118 \left(\frac{\pi}{4}\right)^{2d-2} (d!)^2 (2d)^{-2d} n^{-(d-1)}$.

(2) If $C(\bar{x}, x)$ is square-free, then (C_∞) admits a polynomial-time randomized approximation algorithm with relative approximation ratio $\frac{1}{2}\tau(C_\infty)$.

Theorem 5.6 (1) If $C(\bar{x}, x)$ is nonnegative (including convex as its special case), then (C_S) admits a deterministic polynomial-time approximation algorithm with approximation ratio $\tau(C_S) := (d!)^2(2d)^{-2d}n^{-(d-1)}$.

(2) For general $C(\bar{x}, x)$, (C_S) admits a deterministic polynomial-time approximation algorithm with relative approximation ratio $\frac{1}{2}\tau(C_S)$.

6 Numerical results

In the final section we are going to test the performance of the approximation algorithms proposed. In order to demonstrate the advantage of the tailor-made algorithms for complex polynomial optimization models, we compare our methods with the algorithms for the real case converted equivalently from the original complex model. All the numerical computations are conducted using an Intel Core i5-2520M 2.5GHz computer with 4GB of RAM. The supporting software is MATLAB R2012b.

For the m -th roots of unity constrained problem or unity constrained problem, no *practical* approximation algorithm is available for the equivalently converted model in the real domain. Therefore we focus on the following complex spherical constrained optimization problem:

$$(T_S) \quad \max \quad \sum_{r=1}^{R_1} (z^H A_r z)(z^H A_r^T z) - \sum_{r=R_1+1}^{R_1+R_2} (z^H A_r z)(z^H A_r^T z) \\ \text{s.t.} \quad z \in \mathbf{S}^n,$$

where $A_r \in \mathbb{R}^{n \times n}$ is a real symmetric matrix for $r = 1, 2, \dots, R_1 + R_2$. Obviously, the objective function of (T_S) is a symmetric conjugate form, which can be solved by the algorithm in Theorem 5.6.

To convert (T_S) into an optimization model in the real domain, we denote $x = \text{Re } z$, $y = \text{Im } z$ and $u = \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^{2n}$. Consequently we have $\|u\|^2 = \left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|^2 = \|x\|^2 + \|y\|^2 = \|z\|^2 = 1$. Moreover for any r it is easy to verify that

$$\begin{aligned} (z^H A_r z)(z^H A_r^T z) &= (x - iy)^T A_r (x + iy)(x - iy)^T A_r^T (x + iy) \\ &= (x^T A_r x + y^T A_r y)^2 + (y^T A_r x - x^T A_r y)^2 \\ &:= f_r(u), \end{aligned}$$

where $f_r(u)$ is a homogeneous quartic polynomial of u . Thus (T_S) is equivalent to the following quartic model in the real domain:

$$(R_S) \quad \max \quad \sum_{r=1}^{R_1} f_r(u) - \sum_{r=R_1+1}^{R_1+R_2} f_r(u) \\ \text{s.t.} \quad \|u\| = 1, u \in \mathbb{R}^{2n}.$$

The real approximation algorithm proposed in [10] is applied to solve (R_S) . In order to get an accurate estimation on the approximation performance ratio numerically, an approach in computing the true optimal value of (R_S) is required here. For this purpose, we adopt the recently proposed solution method by Jiang et al. [18] for solving tensor PCA problems. For this particular model (R_S) , the algorithm in [18] is very likely to return a global optimum, or else it provides a close upper bound. The other common approach in the literature to solve (R_S) is the sum of squares (SOS) method proposed by Lasserre [19] and Parrilo [24]. However for this model, the problem dimensions that can be solved by the SOS method are quite limited.

In our first set of tests, we randomly generate 20 instances of (T_S) , and compute their corresponding 20 instances of (R_S) . For the parameters $R_1 = 3$ and $R_2 = 6$, and n is chosen from 4 to

Table 2: Comparison of the approximation ratios

n	Complex approximation algorithm			Real approximation algorithm		
	Average ratio	Worst ratio	Average time	Average ratio	Worst ratio	Average time
4	0.8340	0.6428	0.0143	0.7228	0.5002	0.0039
5	0.7852	0.5605	0.0139	0.7691	0.4855	0.0055
6	0.7682	0.5776	0.0138	0.7350	0.4978	0.0080
7	0.7834	0.5524	0.0155	0.7737	0.5061	0.0117
8	0.7158	0.5549	0.0150	0.6930	0.5056	0.0168
9	0.6955	0.5516	0.0185	0.6794	0.5033	0.0222

Table 3: Comparison of the running time

n	Average time		
	Complex approximation algorithm	Real approximation algorithm	Tensor PCA method
5	0.0138	0.0068	2.6470
10	0.0152	0.0362	122.03
15	0.0239	0.1137	3044.9
20	0.0342	0.2504	∞
25	0.0596	0.5902	∞
30	0.1096	1.2987	∞
35	0.2227	2.5872	∞
40	0.2733	4.1698	∞

9. The complex and the real approximation algorithms are used to solve (T_S) and (R_S) respectively, and their performance ratios are reported in Table 2, where “average ratio” and “worst ratio” denote the average of the relative approximation ratios and the worst relative approximation ratio over all 20 instances, respectively. For each instance, the relative approximation ratio is computed as follows. Denote v to be the the objective value returned by an approximation algorithm, either the real one in [10] for (R_S) or the complex one in this paper for (T_S) . We then apply the algorithm in [18] to solve (R_S) and its minimization counterpart, and output \bar{v} and \underline{v} , which are taken as the optimal values. The relative ratio of this instance is then computed by $\frac{v-\underline{v}}{\bar{v}-\underline{v}}$.

Numerical results in Table 2 suggest that our method enjoys both better average ratio and better worst ratio, demonstrating the advantage of the complex approximation algorithm in terms of the solution quality. However from Table 2, we cannot conclude whether the tailor-made complex approximation algorithm is faster than the real approximation algorithm. Remark that when computing the average running time, we exclude the time spent on reformulating the complex model (T_S) to the real one (R_S) for a real approximation algorithm.

To get an affirmative answer in terms of average CPU time, in this set of tests, we run the algorithms for larger dimensions where n is up to 40. Again for each n , 20 random instances are generated and the average running time is presented in Table 3. It clearly shows that the complex approximation algorithm is faster than the real approximation algorithm for medium to large scale problems. In particular when $n = 40$, our approach is about 15 times faster than the algorithm applied to the reformulated real counterpart.

In our final set of tests, we combine our approach with some local search methods to further improve the solution quality, i.e., using the solution returned by our approach as the starting point of a local search method. The local search method used here is the so-called maximum block improvement (MBI) approach proposed in [6]. We considered problems with dimension ranging

Table 4: Combining the complex approximation algorithm with the MBI method

Inst.	Complex approximation algorithm (CAA) + MBI method					Tensor PCA method	
	CAA value	CAA time	MBI value	MBI time	Total time	PCA value	PCA time
Dimension $n = 4$							
1	15.04	0.0128	15.16	0.0207	0.03	15.16	0.6789
2	-2.22	0.0134	2.76	0.0784	0.09	4.30	0.5079
3	7.63	0.0113	7.91	0.0358	0.05	9.02	1.1915
4	9.04	0.0110	9.39	0.0233	0.03	9.39	0.6405
5	-1.54	0.0117	5.01	0.0587	0.07	6.74	0.5370
Dimension $n = 5$							
1	12.92	0.0549	13.15	0.5304	0.59	13.15	1.7920
2	3.23	0.0116	14.98	0.0906	0.10	14.98	1.7736
3	18.16	0.0114	18.79	0.0734	0.08	18.79	1.9854
4	-0.71	0.0110	5.74	0.0825	0.09	6.21	2.2382
5	10.62	0.0126	11.81	0.0378	0.05	11.81	1.8650
Dimension $n = 6$							
1	-0.52	0.0128	4.84	0.1116	0.12	18.01	5.2123
2	7.23	0.0112	8.38	0.0518	0.06	8.38	5.0184
3	1.51	0.0128	11.10	0.1489	0.16	11.10	76.918
4	-1.82	0.0170	7.81	0.6414	0.66	7.81	6.3776
5	4.91	0.0241	9.50	0.0675	0.09	10.97	10.913
Dimension $n = 7$							
1	-4.09	0.0226	12.16	0.4387	0.46	12.17	113.44
2	9.84	0.0114	15.47	0.1217	0.13	15.47	13.483
3	0.50	0.0121	12.15	0.1179	0.13	12.15	13.077
4	19.91	0.0133	20.42	0.0597	0.07	20.42	12.603
5	-2.15	0.0115	9.84	0.2795	0.29	11.45	11.942
Dimension $n = 8$							
1	1.55	0.0133	14.01	0.1349	0.15	16.40	22.152
2	29.53	0.0115	29.71	0.0393	0.05	29.71	26.362
3	-0.44	0.0123	14.94	0.1700	0.18	17.67	32.473
4	2.60	0.0119	20.73	0.0623	0.07	20.73	31.969
5	19.18	0.0152	21.16	0.0600	0.08	21.16	30.554
Dimension $n = 9$							
1	-0.57	0.0266	21.70	0.3001	0.33	21.70	143.65
2	0.82	0.0128	14.85	0.1727	0.19	14.85	50.436
3	9.34	0.0135	19.40	0.2099	0.22	19.40	65.186
4	21.95	0.0120	25.94	0.1032	0.12	25.94	55.706
5	-2.38	0.0121	22.92	0.4821	0.49	22.92	65.539

from 4 to 9, and for each dimension 5 random instances are tested with their detail results shown in Table 4. The term “total time” stands for the total running time for our approach and the MBI method. According to Table 4, our complex approximation algorithm with the MBI method generates solution whose objective value is very close to the optimal value for most instances.

Acknowledgments

This research was partially supported by National Science Foundation of USA [Grant CMMI-1161242], Natural Science Foundation of China [Grant 11371242], Natural Science Foundation of Shanghai [Grant 12ZR1410100], and Ph.D. Programs Foundation of Chinese Ministry of Education [Grant 20123108120002].

References

- [1] T. Aittomaki and V. Koivunen, *Beampattern Optimization by Minimization of Quartic Polynomial*, Proceedings of 2009 IEEE/SP 15th Workshop on Statistical Signal Processing, 437–440, 2009.
- [2] N. Alon and A. Naor, *Approximating the Cut-Norm via Grothendieck’s Inequality*, SIAM Journal on Computing, 35, 787–803, 2006.
- [3] M. F. Anjos and J. B. Lasserre, *Handbook on Semidefinite, Conic and Polynomial Optimization*, Springer-Verlag, New York, 2011.
- [4] A. Aubry, A. De Maio, B. Jiang, and S. Zhang, *Ambiguity Function Shaping for Cognitive Radar Via Complex Quartic Optimization*, IEEE Transactions on Signal Processing, 61, 5603–5619, 2013.
- [5] A. Ben-Tal, A. Nemirovski, and C. Roos, *Extended Matrix Cube Theorems with Applications to μ -Theory in Control*, Mathematics of Operations Research, 28, 497–523, 2003.
- [6] B. Chen, S. He, Z. Li, and S. Zhang, *Maximum Block Improvement and Polynomial Optimization*, SIAM Journal on Optimization, 22, 87–107, 2012.
- [7] C. Chen and P. P. Vaidyanathan, *MIMO Radar Waveform Optimization With Prior Information of the Extended Target and Clutter*, IEEE Transactions on Signal Processing, 57, 3533–3544, 2009.
- [8] A. C. Doherty and S. Wehner, *Convergence of SDP Hierarchies for Polynomial Optimization on the Hypersphere*, Technical Report, arXiv:1210.5048, 2012.
- [9] S. He, B. Jiang, Z. Li, and S. Zhang, *Probability Bounds for Polynomial Functions in Random Variables*, Mathematics of Operations Research, to appear.
- [10] S. He, Z. Li, and S. Zhang, *Approximation Algorithms for Homogeneous Polynomial Optimization with Quadratic Constraints*, Mathematical Programming, Series B, 125, 353–383, 2010.
- [11] S. He, Z. Li, and S. Zhang, *Inhomogeneous Polynomial Optimization Over Convex Set: An Approximation Approach*, Mathematics of Computation, to appear.

- [12] S. He, Z. Li, and S. Zhang, *Approximation Algorithms for Discrete Polynomial Optimization*, Journal of Operations Research Society of China, 1, 3–36, 2013.
- [13] J. J. Hilling and A. Sudbery, *The Geometric Measure of Multipartite Entanglement and the Singular Values of a Hypermatrix*, Journal of Mathematical Physics, 51, 072102, 2010.
- [14] K. Hou and A. M.-C. So, *Hardness and Approximation Results for L_p -Ball Constrained Homogeneous Polynomial Optimization Problems*, Mathematics of Operations Research, to appear.
- [15] Y. Huang and S. Zhang, *Approximation Algorithms for Indefinite Complex Quadratic Maximization Problems*, Science China Mathematics, 53, 2697–2708, 2010.
- [16] B. Jiang, *Polynomial Optimization: Structures, Algorithms, and Engineering Applications*, Ph.D. Thesis, University of Minnesota, Minneapolis, MN, 2013.
- [17] B. Jiang, Z. Li, and S. Zhang, *Real-valued Conjugate Complex Polynomials and Eigenvalues of Complex Tensors*, Working Paper, 2013.
- [18] B. Jiang, S. Ma, and S. Zhang, *Tensor Principal Component Analysis via Convex Optimization*, Technical Report, Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, MN, 2012.
- [19] J. B. Lasserre, *Global Optimization with Polynomials and the Problem of Moments*, SIAM Journal on Optimization, 11, 796–817, 2001.
- [20] Z. Li, S. He, and S. Zhang, *Approximation Methods for Polynomial Optimization: Models, Algorithms, and Applications*, SpringerBriefs in Optimization, Springer, New York, NY, 2012.
- [21] C. Ling, J. Nie, L. Qi, and Y. Ye, *Biquadratic Optimization over Unit Spheres and Semidefinite Programming Relaxations*, SIAM Journal on Optimization, 20, 1286–1310, 2009.
- [22] Z.-Q. Luo and S. Zhang, *A Semidefinite Relaxation Scheme for Multivariate Quartic Polynomial Optimization with Quadratic Constraints*, SIAM Journal on Optimization, 20, 1716–1736, 2010.
- [23] B. Maricic, Z.-Q. Luo, and T. N. Davidson, *Blind Constant Modulus Equalization via Convex Optimization*, IEEE Transactions on Signal Processing, 51, 805–818, 2003.
- [24] P. A. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. Dissertation, California Institute of Technology, Pasadena, CA, 2000.
- [25] L. Qi, *Eigenvalues and Invariants of Tensors*, Journal of Mathematical Analysis and Applications, 325, 1363–1377, 2007.
- [26] A. M.-C. So, *Deterministic Approximation Algorithms for Sphere Constrained Homogeneous Polynomial Optimization Problems*, Mathematical Programming, Series B, 129, 357–382, 2011.
- [27] A. M.-C. So, J. Zhang, Y. Ye. *On Approximating Complex Quadratic Optimization Problems via Semidefinite Programming Relaxations*, Mathematical Programming, Series B, 110, 93–110, 2007.

- [28] O. Toker and H. Ozbay, *On the Complexity of Purely Complex μ Computation and Related Problems in Multidimensional Systems*, IEEE Transactions on Automatic Control, 43, 409–414, 1998.
- [29] S. Zhang and Y. Huang, *Complex Quadratic Optimization and Semidefinite Programming*, SIAM Journal on Optimization, 16, 871–890, 2006.
- [30] X. Zhang and L. Qi, *The Quantum Eigenvalue Problem and Z-Eigenvalues of Tensors*, Technical Peport, arXiv:1205.1342, 2012.

A Proofs of the lemmas

Lemma 3.2 Define $F_m : \mathbb{C} \mapsto \mathbb{C}$ with $F_m(x) := \frac{m(2-\omega_m-\omega_m^{-1})}{8\pi^2} \sum_{\ell=0}^{m-1} \omega_m^\ell (\arccos(-\operatorname{Re} \omega_m^{-\ell} x))^2$.

(1) If $a \in \mathbb{C}$ and $b \in \Omega_m$, then $F_m(ab) = bF_m(a)$.

(2) If $a \in \mathbb{R}$, then $F_m(a) \in \mathbb{R}$.

Proof. (1) If $b \in \Omega_m$, let $b = \omega_m^k$ for some $k \in \mathbb{Z}$. It holds that

$$\begin{aligned} F_m(ab) &= F_m(\omega_m^k a) = \frac{m(2-\omega_m-\omega_m^{-1})}{8\pi^2} \sum_{\ell=0}^{m-1} \omega_m^\ell \left(\arccos \left(-\operatorname{Re} \omega_m^{-\ell} \omega_m^k a \right) \right)^2 \\ &= \omega_m^k \frac{m(2-\omega_m-\omega_m^{-1})}{8\pi^2} \sum_{\ell=0}^{m-1} \omega_m^{\ell-k} \left(\arccos \left(-\operatorname{Re} \omega_m^{-(\ell-k)} a \right) \right)^2 \\ &= b \frac{m(2-\omega_m-\omega_m^{-1})}{8\pi^2} \sum_{j=-k}^{m-1-k} \omega_m^j \left(\arccos \left(-\operatorname{Re} \omega_m^{-j} a \right) \right)^2 \\ &= bF_m(a). \end{aligned}$$

(2) If $a \in \mathbb{R}$, then $\operatorname{Re} \omega_m^{-k} a = a \operatorname{Re} \omega_m^{-k} = a \operatorname{Re} \omega_m^k = \operatorname{Re} \omega_m^k a$ for any $k \in \mathbb{Z}$. Therefore,

$$\begin{aligned} \overline{F_m(a)} &= \frac{m(2-\omega_m^{-1}-\omega_m)}{8\pi^2} \sum_{\ell=0}^{m-1} \omega_m^{-\ell} \left(\arccos \left(-\operatorname{Re} \omega_m^{-\ell} a \right) \right)^2 \\ &= \frac{m(2-\omega_m-\omega_m^{-1})}{8\pi^2} \sum_{\ell=0}^{m-1} \omega_m^{-\ell} \left(\arccos \left(-\operatorname{Re} \omega_m^\ell a \right) \right)^2 \\ &= \frac{m(2-\omega_m-\omega_m^{-1})}{8\pi^2} \sum_{j=1-m}^0 \omega_m^j \left(\arccos \left(-\operatorname{Re} \omega_m^{-j} a \right) \right)^2 \\ &= F_m(a), \end{aligned}$$

implying that $F_m(a) \in \mathbb{R}$. □

Lemma 4.1 Let $m \in \{3, 4, \dots, \infty\}$. Suppose $x^1, x^2, \dots, x^d \in \mathbb{C}^n$, and $\mathcal{F} \in \mathbb{C}^{n^d}$ is a supersymmetric complex tensor with its associated multilinear form L and homogeneous polynomial H . If $\xi_1, \xi_2, \dots, \xi_d$ are i.i.d. uniform distribution on Ω_m , then

$$\mathbb{E} \left[\prod_{i=1}^d \overline{\xi_i} H \left(\sum_{k=1}^d \xi_k x^k \right) \right] = d! L(x^1, x^2, \dots, x^d) \text{ and } \mathbb{E} \left[\prod_{i=1}^d \xi_i H \left(\sum_{k=1}^d \xi_k x^k \right) \right] = 0.$$

Proof. First we observe that

$$\begin{aligned} \mathbb{E} \left[\prod_{i=1}^d \bar{\xi}_i H \left(\sum_{k=1}^d \xi_k x^k \right) \right] &= \mathbb{E} \left[\prod_{i=1}^d \bar{\xi}_i \sum_{1 \leq k_1, k_2, \dots, k_d \leq d} L \left(\xi_{k_1} x^{k_1}, \xi_{k_2} x^{k_2}, \dots, \xi_{k_d} x^{k_d} \right) \right] \\ &= \sum_{1 \leq k_1, k_2, \dots, k_d \leq d} \mathbb{E} \left[\left(\prod_{i=1}^d \bar{\xi}_i \right) \left(\prod_{j=1}^d \xi_{k_j} \right) L \left(x^{k_1}, x^{k_2}, \dots, x^{k_d} \right) \right]. \end{aligned}$$

If $(k_1, k_2, \dots, k_d) \in \Pi(1, 2, \dots, d)$, i.e., a permutation of $\{1, 2, \dots, d\}$, then

$$\mathbb{E} \left[\left(\prod_{i=1}^d \bar{\xi}_i \right) \left(\prod_{j=1}^d \xi_{k_j} \right) \right] = \mathbb{E} \left[\prod_{i=1}^d \bar{\xi}_i \xi_i \right] = 1;$$

otherwise, there is k_0 ($1 \leq k_0 \leq d$) such that $k_0 \neq k_j$ for all $j = 1, 2, \dots, d$. In the latter case,

$$\mathbb{E} \left[\left(\prod_{i=1}^d \bar{\xi}_i \right) \left(\prod_{j=1}^d \xi_{k_j} \right) \right] = \mathbb{E} [\bar{\xi}_{k_0}] \mathbb{E} \left[\left(\prod_{1 \leq i \leq d, i \neq k_0} \bar{\xi}_i \right) \left(\prod_{j=1}^d \xi_{k_j} \right) \right] = 0.$$

Since the number of different permutations of $\{1, 2, \dots, d\}$ is $d!$, by taking into account the super-symmetric property of L , the first identity follows.

For the second identity, similarly we have

$$\mathbb{E} \left[\prod_{i=1}^d \xi_i H \left(\sum_{k=1}^d \xi_k x^k \right) \right] = \sum_{1 \leq k_1, k_2, \dots, k_d \leq d} \mathbb{E} \left[\left(\prod_{i=1}^d \xi_i \right) \left(\prod_{j=1}^d \xi_{k_j} \right) L \left(x^{k_1}, x^{k_2}, \dots, x^{k_d} \right) \right].$$

There exists k_0 ($1 \leq k_0 \leq d$) such that ξ_{k_0} appears once or twice in $\left(\prod_{i=1}^d \xi_i \right) \left(\prod_{j=1}^d \xi_{k_j} \right)$. For $m \in \{3, 4, \dots, \infty\}$, we notice that $\mathbb{E}[\xi_i] = 0$ and $\mathbb{E}[\xi_i^2] = 0$ for $i = 1, 2, \dots, d$. By independence of ξ_i 's, $\mathbb{E} \left[\left(\prod_{i=1}^d \xi_i \right) \left(\prod_{j=1}^d \xi_{k_j} \right) \right]$ is always zero, leading to the second identity. \square

Lemma 5.1 *Let $m \in \{3, 4, \dots, \infty\}$. Suppose $x^1, x^2, \dots, x^{2d} \in \mathbb{C}^n$, and $\mathcal{F} \in \mathbb{C}^{n^{2d}}$ is a conjugate partial-symmetric tensor with its associated multilinear form L and symmetric conjugate form C . If $\xi_1, \xi_2, \dots, \xi_{2d}$ are i.i.d. uniform distribution on Ω_m , then*

$$\mathbb{E} \left[\left(\prod_{i=1}^d \xi_i \right) \left(\prod_{i=d+1}^{2d} \bar{\xi}_i \right) C \left(\sum_{k=1}^d \bar{\xi}_k x^k + \sum_{k=d+1}^{2d} \overline{\xi_k x^k}, \sum_{k=1}^d \xi_k \overline{x^k} + \sum_{k=d+1}^{2d} \xi_k x^k \right) \right] = (d!)^2 L(x^1, x^2, \dots, x^{2d}).$$

Proof. We first consider the following

$$\begin{aligned} &\mathbb{E} \left[\left(\prod_{i=1}^d \xi_i \right) \left(\prod_{i=d+1}^{2d} \bar{\xi}_i \right) C \left(\sum_{k=1}^{2d} \overline{\xi_k x^k}, \sum_{k=1}^{2d} \xi_k x^k \right) \right] \\ &= \mathbb{E} \left[\left(\prod_{i=1}^d \xi_i \right) \left(\prod_{i=d+1}^{2d} \bar{\xi}_i \right) \sum_{1 \leq k_1, \dots, k_{2d} \leq 2d} L \left(\overline{\xi_{k_1} x^{k_1}}, \dots, \overline{\xi_{k_d} x^{k_d}}, \xi_{k_{d+1}} x^{k_{d+1}}, \dots, \xi_{k_{2d}} x^{k_{2d}} \right) \right] \\ &= \sum_{1 \leq k_1, \dots, k_{2d} \leq 2d} \mathbb{E} \left[\left(\prod_{i=1}^d \xi_i \right) \left(\prod_{i=d+1}^{2d} \bar{\xi}_i \right) \left(\prod_{j=1}^d \bar{\xi}_{k_j} \right) \left(\prod_{j=d+1}^{2d} \xi_{k_j} \right) L \left(\overline{x^{k_1}}, \dots, \overline{x^{k_d}}, x^{k_{d+1}}, \dots, x^{k_{2d}} \right) \right]. \end{aligned}$$

For $m \in \{3, 4, \dots, \infty\}$, we observe that $\mathbb{E}[\xi_i] = 0$ and $\mathbb{E}[\xi_i^2] = 0$ for $i = 1, 2, \dots, 2d$. Using a similar argument in the proof of Lemma 4.1, we have

$$\mathbb{E} \left[\left(\prod_{i=1}^d \xi_i \right) \left(\prod_{i=d+1}^{2d} \bar{\xi}_i \right) \left(\prod_{j=1}^d \bar{\xi}_{k_j} \right) \left(\prod_{j=d+1}^{2d} \xi_{k_j} \right) \right] = \begin{cases} 1 & (k_1, \dots, k_d) \in \Pi(1, \dots, d) \text{ and} \\ & (k_{d+1}, \dots, k_{2d}) \in \Pi(d+1, \dots, 2d); \\ 0 & \text{otherwise.} \end{cases}$$

By noticing that \mathcal{F} is conjugate partial-symmetric (see Definition 2.2), and considering numbers of permutations, it follows that

$$\mathbb{E} \left[\left(\prod_{i=1}^d \xi_i \right) \left(\prod_{i=d+1}^{2d} \bar{\xi}_i \right) C \left(\sum_{k=1}^{2d} \bar{\xi}_k x^k, \sum_{k=1}^{2d} \xi_k x^k \right) \right] = (d!)^2 L \left(\bar{x}^1, \dots, \bar{x}^d, x^{d+1}, \dots, x^{2d} \right).$$

Finally, replacing \bar{x}^k by x^k for $k = 1, 2, \dots, d$ in the above identity leads to the desired result. \square