Approximation Algorithms for Discrete Polynomial Optimization

Simai HE [*]   Zhening LI [†]   Shuzhong ZHANG [‡]

December 22, 2012

Abstract

In this paper, we consider approximation algorithms for optimizing a generic multivariate polynomial function in discrete (typically binary) variables. Such models have natural applications in graph theory, neural networks, error-correcting codes, among many others. In particular, we focus on three types of optimization models: (1) maximizing a homogeneous polynomial function in binary variables; (2) maximizing a homogeneous polynomial function in binary variables, mixed with variables under spherical constraints; (3) maximizing an inhomogeneous polynomial function in binary variables. We propose polynomial-time randomized approximation algorithms for such polynomial optimization models, and establish the approximation ratios (or relative approximation ratios whenever appropriate) for the proposed algorithms. Some examples of applications for these models and algorithms are discussed as well.

Keywords: polynomial optimization problem; binary integer programming; mixed integer programming; approximation algorithm; approximation ratio.

Mathematics Subject Classification: 90C10, 90C11, 90C59, 15A69, 90C26.

---

[*]Department of Management Sciences, City University of Hong Kong, Hong Kong. Email: simaihe@cityu.edu.hk. The research of this author was supported in part by Hong Kong General Research Fund (Grant CityU143711).

[†]Department of Mathematics, Shanghai University, Shanghai 200444, China. Email: zheningli@gmail.com. The research of this author was supported in part by Natural Science Foundation of Shanghai (Grant 12ZR1410100) and Ph.D. Programs Foundation of Chinese Ministry of Education (Grant 20123108120002).

[‡]Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, MN 55455, USA. Email: zhangs@umn.edu. The research of this author was supported in part by U.S. National Science Foundation (Grant CMMI-1161242).

1 Introduction

This paper is concerned with optimizing a (high degree) multivariate polynomial function in (mixed) binary variables. Our basic model is to maximize a d-th degree polynomial function $p(\boldsymbol{x})$ where $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ is chosen such that $x_i \in \{1, -1\}$ for $i = 1, 2, \ldots, n$. For ease of referencing, let us call this basic model to be $(P) : \max_{\boldsymbol{x} \in \{1,-1\}^n} p(\boldsymbol{x})$. This type of problem can be found in a great variety of application domains. For example, the following hypergraph max-covering problem is well studied in the literature, which is precisely (P). Given a hypergraph $H = (V, E)$ with V being the set of vertices and E the set of hyperedges (or subsets of V), and each hyperedge $e \in E$ is associated with a real-valued weight $w(e)$. The problem is to find a subset S of the vertices set V, such that the total weight of the hyperedges covered by S is maximized. Denoting $x_i \in \{0, 1\}$ $(i = 1, 2, \ldots, n)$ to indicate whether or not vertex i is selected in S. The problem thus is $\max_{\boldsymbol{x} \in \{0,1\}^n} \sum_{e \in E} w(e) \prod_{i \in e} x_i$. By a simple variable transformation $x_i \rightarrow (x_i + 1)/2$, the problem is transformed to (P), and vice versa.

Note that (P) is a fundamental problem in integer programming. As such it has received attention in the literature; see [17, 18]. It is also known as *Fourier support graph* problem. Mathematically, a polynomial function $p : \{-1, 1\}^n \rightarrow \mathbb{R}$ has Fourier expansion $p(\boldsymbol{x}) = \sum_{S \subseteq \{1,2,\ldots,n\}} \hat{p}(S) \prod_{i \in S} x_i$, which is also called Fourier support graph. Assume that p has only succinct (polynomially many) non-zero Fourier coefficient $\hat{p}(S)$. The question is: Can we compute the maximum value of p over the discrete cube $\{1, -1\}^n$, or alternatively can we find a good approximate solution in polynomial-time? The latter question actually motivates this paper. Indeed, (P) has been investigated extensively in the quadratic case, due to its connections to various graph partitioning problems, e.g., the maximum cut problem [16]. In general, (P) is closely related to finding the *maximum weighted independent set* in a graph. In particular, let $G = (V, E)$ be a graph with V the set of vertices V and E the set of edges, and each vertex is assigned a positive weight. We call S to be an independent set of vertices if and only if $S \subseteq V$ and no two vertices in S share an edge. The problem is to find an independent set of vertices such that the sum of its weights is maximum over all possible independent sets.

In fact, any unconstrained binary polynomial maximization problem can be transformed into the maximum weighted independent set problem, which is also commonly used technique in the literature for solving (P) (see e.g., [5, 30]). The transformation uses the concept of a *conflict graph* of a 0-1 polynomial function. The idea is illustrated in the following example. Let us consider

$$f(\boldsymbol{x}) = -2x_1 - 2x_2 + 5x_1x_2 - 4x_1x_2x_3, \ (x_1, x_2, x_3) \in \{0,1\}^3.$$

Note that $f(\boldsymbol{x})$ can be transformed to an equivalent polynomial so that all terms (except the constant term) have positive coefficients. The new polynomial involves both the variables and their complements, i.e., $\bar{x}_i := 1 - x_i$ for $i = 1, 2, 3$. In our example, such polynomial can be

$$f(\boldsymbol{x}) = -4 + 2\bar{x}_1 + 2\bar{x}_2 + x_1x_2 + 4x_1x_2\bar{x}_3.$$

The conflict graph $G(f)$ associated with a polynomial $f(\boldsymbol{x})$ has vertices corresponding to the terms of $f(\boldsymbol{x})$, and each vertex is associated with a term in the polynomial except for the constant term. Two vertices in $G(f)$ are connected by an edge if and only if one of the corresponding terms contains a variable and the other corresponding term contains its complement variable. The weight of a vertex in $G(f)$ is the coefficient of the corresponding term in f. The conflict graph of $f(\boldsymbol{x})$ is shown in Figure 1. Maximizing the weighted independent set of the conflict graph also solves the binary polynomial optimization problem. Beyond its connection to the graph problems, (P) also has applications in *neural networks* [21, 8, 4], *error-correcting codes* [8, 29], etc. For instance,
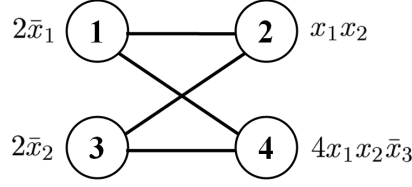
Figure 1: Conflict graph associated with $-2x_1 - 2x_2 + 5x_1x_2 - 4x_1x_2x_3$

recently Khot and Naor [24] show that it has applications in the problem of *refutation of random k-CNF formulas* [12, 13].

One important subclass of polynomial function is homogeneous polynomials. Likewise, the homogeneous quadratic case of (P) has been studied extensively; see e.g. [16, 27, 28, 2]. Homogeneous cubic polynomial is also studies by Khot and Naor [24]. Another interesting problem of this class is the $\infty \mapsto 1$-norm of a matrix $\boldsymbol{M} = (a_{ij})_{n_1 \times n_2}$ (see e.g., [2]), i.e.,

$$\|\boldsymbol{M}\|_{\infty \mapsto 1} = \max_{\boldsymbol{x} \in \{1,-1\}^{n_1}, \boldsymbol{y} \in \{1,-1\}^{n_2}} \boldsymbol{x}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{y} := \sum_{1 \leq i \leq n_1, 1 \leq j \leq n_2} a_{ij} x_i y_j.$$

It is quite natural to extend the problem of $\infty \mapsto 1$-norm to higher order tensors. In particular, the $\|\boldsymbol{F}\|_{\infty \mapsto 1}$ of a d-th order tensor $\boldsymbol{F} = (a_{i_1 i_2 \cdots i_d})$ can be defined as

$$\max_{\boldsymbol{x}^k \in \{1,-1\}^{n_k}, \, k=1,2,\ldots,d} F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d) := \sum_{1 \leq i_1 \leq n_1, 1 \leq i_2 \leq n_2, \cdots, 1 \leq i_d \leq n_d} a_{i_1 i_2 \cdots i_d} x_{i_1}^1 x_{i_2}^2 \cdots x_{i_d}^d.$$

The other generalization of the matrix $\infty \mapsto 1$-norm is to extend the entry a_{ij} of the matrix $\boldsymbol{M}$ to symmetric matrix $\boldsymbol{A}_{ij}$, i.e., the problem of

$$\max_{\boldsymbol{x} \in \{1,-1\}^{n_1}, \boldsymbol{y} \in \{1,-1\}^{n_2}} \lambda_{\max} \left(\sum_{1 \leq i \leq n_1, 1 \leq j \leq n_2} x_i y_j \boldsymbol{A}_{ij} \right),$$

where $\lambda_{\max}(\cdot)$ indicates the largest eigenvalue of a matrix. If the matrix $\boldsymbol{A}_{ij}$ is not restricted to be symmetric, we may instead maximize the largest singular value, i.e.,

$$\max_{\boldsymbol{x} \in \{1,-1\}^{n_1}, \boldsymbol{y} \in \{1,-1\}^{n_2}} \sigma_{\max} \left(\sum_{1 \leq i \leq n_1, 1 \leq j \leq n_2} x_i y_j \boldsymbol{A}_{ij} \right).$$

These two problems are actually equivalent to

$$\max_{\boldsymbol{x} \in \{1,-1\}^{n_1}, \boldsymbol{y} \in \{1,-1\}^{n_2}, \|\boldsymbol{z}\|_2 = 1} F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{z}) \quad \text{and} \quad \max_{\boldsymbol{x} \in \{1,-1\}^{n_1}, \boldsymbol{y} \in \{1,-1\}^{n_2}, \|\boldsymbol{z}\|_2 = \|w\|_2 = 1} F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{w})$$

respectively, where F is a multilinear function induced by the tensor $\boldsymbol{F}$, whose (i, j, k, ℓ)-th entry is (k, ℓ)-th entry of the matrix $\boldsymbol{A}_{ij}$.

In fact, a very interesting and succinct matrix combinatorial problem is: Given n matrices $\boldsymbol{A}_i$ $(i = 1, 2, \ldots, n)$, find a binary combination of the matrices so as to maximize the spectral norm of the combined matrix:

$$\max_{\boldsymbol{x} \in \{1,-1\}^n} \sigma_{\max} \left(\sum_{i=1}^{n} x_i \boldsymbol{A}_i \right).$$

3

This is indeed equivalent to

$$\max_{\boldsymbol{x}\in\{1,-1\}^n, \|\boldsymbol{y}\|_2=\|\boldsymbol{z}\|_2=1} F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}).$$

All the problems studied in this paper are NP-hard in general, and our focus will be polynomial-time approximation algorithms. In the case that the objective polynomial is quadratic, a well known example is the semidefinite programming relaxation and randomization approach for the max-cut problem due to Goemans and Williamson [16], where essentially a 0.878-approximation ratio of the model $\max_{\boldsymbol{x}\in\{1,-1\}^n} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{x}$ is shown with $\boldsymbol{M}$ being the Laplacian of a given graph. In the case $\boldsymbol{M}$ is only known to be positive semidefinite, Nesterov [27] derived a 0.636-approximation bound. Charikar and Wirth [9] considered a more general model; they proposed an $\Omega\left(\frac{1}{\log n}\right)$-approximate algorithm for diagonal-free $\boldsymbol{M}$. For the matrix $\infty \mapsto 1$-norm problem

$$\max_{\boldsymbol{x}\in\{1,-1\}^{n_1}, \boldsymbol{y}\in\{1,-1\}^{n_2}} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{y},$$

Alon and Naor [2] derived a 0.56-approximation bound. Remark that all these approximation bounds remain hitherto the best available ones. When the degree of the polynomial function is greater than 2, to the best of our knowledge, the only known approximation result in the literature is due to Khot and Naor [24], where they showed how to estimate the optimal value of the problem $\max_{\boldsymbol{x}\in\{1,-1\}^n} \sum_{1\le i,j,k\le n} a_{ijk}x_i x_j x_k$ with $(a_{ijk})_{n\times n\times n}$ being square-free ($a_{ijk} = 0$ whenever two of the indices are equal). Specifically, they presented a polynomial-time procedure to get an estimated value that is no less than $\Omega\left(\sqrt{\frac{\ln n}{n}}\right)$ times the optimal value. No solution, however, can be derived from the process. Moreover, the process is highly complex and is mainly of theoretical interest.

In this paper we consider the optimization models for a general polynomial function of any fixed degree d in (mixed) binary variables, and present polynomial-time randomized approximation algorithms. The algorithms proposed are fairly simple to implement. This study is motivated by our previous investigations on polynomial optimization under quadratic constraints [19, 20], as well as recent developments on homogeneous polynomial optimization under spherical constraints, e.g., So [31] and Chen et al. [10]. However, the discrete models studied in this paper have novel features, and the analysis is therefore entirely different from previous works. This paper is organized as follows. First, we introduce the notations and models in Section 2. In Section 3, we present the new approximation results, and also sketch the main ideas, while leaving the technical details to the appendix (Appendix A). In Section 4 we shall discuss a few more specific problems where the models introduced can be directly applied.

2 Notations and Model Descriptions

In this paper we shall use the boldface letters to denote vectors, matrices, and tensors in general (e.g., the decision variable $\boldsymbol{x}$, the data matrix $\boldsymbol{Q}$, and the tensor form $\boldsymbol{F}$), while the usual lowercase letters are reserved for scalars (e.g., x_1 being the first component of the vector $\boldsymbol{x}$).

2.1 Objective Functions

The objective functions of the optimization models studied in this paper are all multivariate polynomial functions. The following multilinear tensor function plays a major role in our discussion:

$$\text{Function T} \qquad F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d) = \sum_{1\le i_1\le n_1, 1\le i_2\le n_2, \cdots, 1\le i_d\le n_d} a_{i_1 i_2\cdots i_d} x_{i_1}^1 x_{i_2}^2 \cdots x_{i_d}^d,$$

4

where $\boldsymbol{x}^k \in \mathbb{R}^{n_k}$ for $k = 1, 2, \ldots, d$; and the letter 'T' signifies the notion of *tensor*. In the shorthand notation we shall denote $\boldsymbol{F} = (a_{i_1 i_2 \cdots i_d}) \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ to be a d-th order tensor, and F to be its corresponding multilinear form. Closely related with the tensor $\boldsymbol{F}$ is a general d-th degree homogeneous polynomial function $f(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbb{R}^n$. We call the tensor $\boldsymbol{F} = (a_{i_1 i_2 \cdots i_d})$ *super-symmetric* (see [25]) if $a_{i_1 i_2 \cdots i_d}$ is invariant under all permutations of $\{i_1, i_2, \cdots, i_d\}$. As any homogeneous quadratic function uniquely determines a symmetric matrix, a given d-th degree homogeneous polynomial function $f(\boldsymbol{x})$ also uniquely determines a super-symmetric tensor. In particular, if we denote a d-th degree homogeneous polynomial function:

$$\text{Function H} \qquad f(x) = \sum_{1 \le i_1 \le i_2 \le \cdots \le i_d \le n} a_{i_1 i_2 \cdots i_d} x_{i_1} x_{i_2} \cdots x_{i_d},$$

then its corresponding super-symmetric tensor form can be written as $\boldsymbol{F} = (b_{i_1 i_2 \cdots i_d}) \in \mathbb{R}^{n^d}$, with $b_{i_1 i_2 \cdots i_d} \equiv a_{i_1 i_2 \cdots i_d} / |\Pi(i_1, i_2, \cdots, i_d)|$, where $|\Pi(i_1, i_2, \cdots, i_d)|$ is the number of distinctive permutations of the indices $\{i_1, i_2, \cdots, i_d\}$. This super-symmetric tensor representation is indeed unique. Let F be its corresponding multilinear function defined by the super-symmetric tensor $\boldsymbol{F}$, then we have $f(\boldsymbol{x}) = F(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{d})$. The letter 'H' here is used to emphasize that the polynomial function in question is *homogeneous*.

We shall also consider in this paper the following:

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

where $\boldsymbol{x}^k \in \mathbb{R}^{n_k}$ for $k = 1, 2, \ldots, s$, $d_1 + d_2 + \cdots + d_s = d$, and d-th order tensor form $\boldsymbol{F} \in \mathbb{R}^{n_1^{d_1} \times n_2^{d_2} \times \cdots \times n_s^{d_s}}$; the letter 'M' signifies the notion of *mixed polynomial forms*. We may without loss of generality assume that $\boldsymbol{F}$ has partial symmetric property, namely for any fixed $(\boldsymbol{x}^2, \boldsymbol{x}^3, \cdots, \boldsymbol{x}^s)$, $F(\underbrace{\cdot, \cdot, \cdots, \cdot}_{d_1}, \underbrace{\boldsymbol{x}^2, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^2}_{d_2}, \cdots, \underbrace{\boldsymbol{x}^s, \boldsymbol{x}^s, \cdots, \boldsymbol{x}^s}_{d_s})$ is a super-symmetric d_1-th order tensor, and so on.

Beyond the homogeneous polynomial functions described above, a generic multivariate inhomogeneous polynomial function of degree d, $p(\boldsymbol{x})$, can be explicitly written as a summation of homogenous polynomial functions in decreasing degrees, namely

$$\text{Function P} \qquad p(\boldsymbol{x}) := \sum_{k=1}^{d} F_k(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{k}) + f_0 = \sum_{k=1}^{d} f_k(\boldsymbol{x}) + f_0,$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $f_0 \in \mathbb{R}$, and $f_k(\boldsymbol{x}) = F_k(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{k})$ is a homogenous polynomial function of degree k for $k = 1, 2, \ldots, d$; the letter 'P' signifies the notion of *polynomial*.

Throughout we shall adhere to the notation F for a multilinear form defined by a tensor form $\boldsymbol{F}$, and f for a homogenous polynomial function, and p for an inhomogeneous polynomial function. Without loss of generality we assume that $n_1 \le n_2 \le \cdots \le n_d$ in the tensor form $\boldsymbol{F} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, and $n_1 \le n_2 \le \cdots \le n_s$ in the tensor form $\boldsymbol{F} \in \mathbb{R}^{n_1^{d_1} \times n_2^{d_2} \times \cdots \times n_s^{d_s}}$. We also assume at lease one component of the tensor form, $\boldsymbol{F}$ in Functions T, H, M, and $\boldsymbol{F}_d$ in Function P is nonzero to avoid triviality. Finally, without loss of generality we assume the inhomogeneous polynomial function $p(\boldsymbol{x})$ has no constant term, i.e., $f_0 = 0$ in Function P.

2.2 Decision Variables

This paper is focused on integer and mixed integer programming with polynomial functions. In particular, two types of decision variables are considered in this paper: discrete binary variables

$$\boldsymbol{x} \in \mathbb{B}^n := \left\{ \boldsymbol{z} \in \mathbb{R}^n \,\middle|\, z_i{}^2 = 1, i = 1, 2, \ldots, n \right\},$$

and continuous variables on the unit sphere:

$$\boldsymbol{y} \in \mathbb{S}^m := \left\{ \boldsymbol{z} \in \mathbb{R}^m \,\middle|\, \|\boldsymbol{z}\| := \left(z_1{}^2 + z_2{}^2 + \cdots + z_m{}^2 \right)^{1/2} = 1 \right\}.$$

Note that in this paper we shall by default use the Euclidean norm for vectors, matrices and tensors. The decision variables in our models range from the pure binary vector $\boldsymbol{x}$, to a mixed one including both $\boldsymbol{x}$ $(\in \mathbb{B}^n)$ and $\boldsymbol{y}$ $(\in \mathbb{S}^m)$.

2.3 Model Descriptions

In this paper we consider the following binary integer optimization models with objection functions as specified in Section 2.1:

(T) max $F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d)$
  s.t. $\boldsymbol{x}^k \in \mathbb{B}^{n_k}, \ k = 1, 2, \ldots, d;$

(H) max $f(\boldsymbol{x}) = F(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{d})$
  s.t. $\boldsymbol{x} \in \mathbb{B}^n;$

(M) max $f(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^s) = F(\underbrace{\boldsymbol{x}^1, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^1}_{d_1}, \underbrace{\boldsymbol{x}^2, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^2}_{d_2}, \cdots, \underbrace{\boldsymbol{x}^s, \boldsymbol{x}^s, \cdots, \boldsymbol{x}^s}_{d_s})$
  s.t. $\boldsymbol{x}^k \in \mathbb{B}^{n_k}, \ k = 1, 2, \ldots, s;$

(P) max $p(\boldsymbol{x}) = \sum_{k=1}^{d} F_k(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{k}) + f_0$
  s.t. $\boldsymbol{x} \in \mathbb{B}^n;$

and their *mixed* models:

$(T)'$ max $F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d, \boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^{d'})$
  s.t. $\boldsymbol{x}^k \in \mathbb{B}^{n_k}, \ k = 1, 2, \ldots, d,$
      $\boldsymbol{y}^\ell \in \mathbb{S}^{m_\ell}, \ \ell = 1, 2, \ldots, d';$

$(H)'$ max $f(\boldsymbol{x}, \boldsymbol{y}) = F(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{d}, \underbrace{\boldsymbol{y}, \boldsymbol{y}, \cdots, \boldsymbol{y}}_{d'})$
  s.t. $\boldsymbol{x} \in \mathbb{B}^n,$
      $\boldsymbol{y} \in \mathbb{S}^m;$

$(M)'$ max $f(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^s, \boldsymbol{y}^1, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^t)$
      $= F(\underbrace{\boldsymbol{x}^1, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^1}_{d_1}, \cdots, \underbrace{\boldsymbol{x}^s, \boldsymbol{x}^s, \cdots, \boldsymbol{x}^s}_{d_s}, \underbrace{\boldsymbol{y}^1, \boldsymbol{y}^1, \cdots, \boldsymbol{y}^1}_{d'_1}, \cdots, \underbrace{\boldsymbol{y}^t, \boldsymbol{y}^t, \cdots, \boldsymbol{y}^t}_{d'_t})$
  s.t. $\boldsymbol{x}^k \in \mathbb{B}^{n_k}, \ k = 1, 2, \ldots, s,$
      $\boldsymbol{y}^\ell \in \mathbb{S}^{m_\ell}, \ \ell = 1, 2, \ldots, t.$

Let $d_1 + d_2 + \cdots + d_s = d$ and $d'_1 + d'_2 + \cdots + d'_t = d'$ in the above mentioned models. The degrees of the polynomial functions in these models, d for the pure binary models and $d + d'$ for the mixed models, are understood as fixed constants in our subsequent discussions. As before, we also assume that the tensor forms of the objective functions in $(H)'$ and $(M)'$ to have partial symmetric property, $m_1 \leq m_2 \leq \cdots \leq m_{d'}$ in $(T)'$, and $m_1 \leq m_2 \leq \cdots \leq m_t$ in $(M)'$.

2.4 Approximation Ratios

All the optimization problems mentioned in the previous subsection are in general NP-hard when the degree of the objective polynomial function is larger than or equal to 2. This is because each one includes computing the matrix $\infty \mapsto 1$-norm as a subclass, i.e.,

$$\|Q\|_{\infty \mapsto 1} = \begin{array}{ll} \max & (x^1)^{\mathrm{T}} Q x^2 \\ \text{s.t.} & x^1 \in \mathbb{B}^{n_1}, \\ & x^2 \in \mathbb{B}^{n_2}. \end{array}$$

Thus, in this paper we shall focus on polynomial-time approximation algorithms with provable worst-case performance ratios. For any maximization problem (P) defined as $\max_{x \in S} f(x)$, we use $v_{\max}(P)$ to denote its optimal value, and $v_{\min}(P)$ to denote the optimal value of its minimization counterpart, i.e.,

$$v_{\max}(P) := \max_{x \in S} f(x) \text{ and } v_{\min}(P) := \min_{x \in S} f(x).$$

Definition 2.1 *We call the maximization model (P) to admit a polynomial-time approximation algorithm with approximation ratio $\tau \in (0, 1]$, if $v_{\max}(P) \geq 0$ and a feasible solution $z \in S$ can be found in polynomial-time such that $f(z) \geq \tau \, v_{\max}(P)$.*

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

Regarding to the relative approximation ratios (Definition 2.2), in some cases it is convenient to use the equivalent form: $v_{\max}(P) - f(z) \leq (1 - \tau) \, (v_{\max}(P) - v_{\min}(P))$.

3 Bounds on the Approximation Ratios

In this section we shall present our main results, viz. the approximation ratios for the discrete polynomial optimization models considered in this paper. In order not to distract reading the main results, the proofs will be postponed and placed in the appendix (Appendix A). To simplify, we use the notion $\Omega(f(n))$ to signify that there are positive universal constants α and n_0 such that $\Omega(f(n)) \geq \alpha f(n)$ for all $n \geq n_0$. Throughout our discussion, we shall fix the degree of the objective polynomial function (denoted by d or $d + d'$ in the paper) to be a constant.

3.1 Homogeneous Polynomials in Binary Variables

Theorem 3.1 (T) : $\max_{x^k \in \mathbb{B}^{n_k}} F(x^1, x^2, \cdots, x^d)$ *admits a polynomial-time approximation algorithm with approximation ratio τ_T, where*

$$\tau_T := (n_1 n_2 \cdots n_{d-2})^{-\frac{1}{2}} (2/\pi)^{d-1} \ln(1 + \sqrt{2}) = \Omega\left((n_1 n_2 \cdots n_{d-2})^{-\frac{1}{2}}\right).$$

We remark that when $d = 2$, (T) is to compute $\|\boldsymbol{F}\|_{\infty \mapsto 1}$. The current best polynomial-time approximation ratio for that problem is $\frac{2\ln(1+\sqrt{2})}{\pi} \approx 0.56$ due to Alon and Naor [2]. Huang and Zhang [22] considered similar problems for the complex variables and derived constant approximation ratios.

When $d = 3$, (T) is a slight generalization of the model considered by Khot and Naor [24], where $\boldsymbol{F}$ was assumed to be super-symmetric (implying $n_1 = n_2 = n_3$) and square-free (i.e., $a_{ijk} = 0$ whenever two of the three indices are equal). In our case, we discard the assumptions on the symmetry and the square-free property altogether. The approximation bound of the optimal value given in [24] is $\Omega\left(\sqrt{\frac{\ln n_1}{n_1}}\right)$; however, no polynomial-time procedure is provided to find a corresponding approximate solution.

Our approximation algorithm works for general degree d based on recursion, and is fairly simple. We may take any approximation algorithm for the $d = 2$ case, say the algorithm by Alon and Naor [2], as a basis. When $d = 3$, noticing that any $n_1 \times n_2 \times n_3$ third order tensor can be written as an $(n_1 n_2) \times n_3$ matrix by combining its first and second modes, (T) can be *relaxed* to

$$\begin{array}{ll} \max & F(\boldsymbol{X}, \boldsymbol{x}^3) := \sum_{1 \leq i \leq n_1, 1 \leq j \leq n_2, 1 \leq k \leq n_3} a_{ijk} X_{ij} x_k^3 \\ \text{s.t.} & \boldsymbol{X} \in \mathbb{B}^{n_1 n_2}, \boldsymbol{x}^3 \in \mathbb{B}^{n_3}. \end{array}$$

This problem is the exact form of (T) when $d = 2$, which can be solved approximately with approximation ratio $\frac{2\ln(1+\sqrt{2})}{\pi}$. Denote its approximate solution to be $(\hat{\boldsymbol{X}}, \hat{\boldsymbol{x}}^3)$. The next key step is to recover $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2)$ from $\hat{\boldsymbol{X}}$. For this purpose, we introduce the following decomposition routine, which plays a fundamental role in our algorithms.

DR (Decomposition Routine) 3.1

- *Input: matrices $\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2}$ and $\hat{\boldsymbol{X}} \in \mathbb{B}^{n_1 \times n_2}$.*

- *Construct*

$$\tilde{\boldsymbol{X}} = \left[\begin{array}{cc} \boldsymbol{I}_{n_1 \times n_1} & \hat{\boldsymbol{X}}/\sqrt{n_1} \\ \hat{\boldsymbol{X}}^{\mathrm{T}}/\sqrt{n_1} & \hat{\boldsymbol{X}}^{\mathrm{T}}\hat{\boldsymbol{X}}/n_1 \end{array} \right] \succeq 0.$$

- *Randomly generate*

$$\begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{pmatrix} \sim \mathcal{N}(\boldsymbol{0}_{n_1+n_2}, \tilde{\boldsymbol{X}})$$

  and compute

$$\hat{\boldsymbol{x}}^1 = \operatorname{sign}(\boldsymbol{\xi}), \hat{\boldsymbol{x}}^2 = \operatorname{sign}(\boldsymbol{\eta});$$

  repeat if necessary, until $(\hat{\boldsymbol{x}}^1)^{\mathrm{T}} \boldsymbol{M} \hat{\boldsymbol{x}}^2 \geq \frac{2}{\pi\sqrt{n_1}} \boldsymbol{M} \bullet \hat{\boldsymbol{X}}$.

- *Output: binary vectors $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2)$.*

If we let $\boldsymbol{M} = F(\cdot, \cdot, \hat{\boldsymbol{x}}^3)$ and apply DR 3.1, then we can prove the output $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2)$ satisfies

$$\mathsf{E}[F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \hat{\boldsymbol{x}}^3)] = \mathsf{E}[(\hat{\boldsymbol{x}}^1)^{\mathrm{T}} \boldsymbol{M} \hat{\boldsymbol{x}}^2] \geq \frac{2}{\pi\sqrt{n_1}} \boldsymbol{M} \bullet \hat{\boldsymbol{X}} = \frac{2}{\pi\sqrt{n_1}} F(\hat{\boldsymbol{X}}, \hat{\boldsymbol{x}}^3) \geq \frac{4\ln(1+\sqrt{2})}{\pi^2\sqrt{n_1}} v_{\max}(T),$$

which yields an approximation ratio for $d = 3$. By a recursive procedure, the approximation algorithm is readily extended to solve (T) with any fixed degree d.

8

Theorem 3.2 *If* $F(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{d})$ *is square-free and d is odd, then $(H) : \max_{\boldsymbol{x} \in \mathbb{B}^n} f(\boldsymbol{x})$ admits a polynomial-time approximation algorithm with approximation ratio τ_H, where*

$$\tau_H := d! d^{-d} n^{-\frac{d-2}{2}} (2/\pi)^{d-1} \ln(1 + \sqrt{2}) = \Omega\left(n^{-\frac{d-2}{2}}\right).$$

Theorem 3.3 *If* $F(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{d})$ *is square-free and d is even, then $(H) : \max_{\boldsymbol{x} \in \mathbb{B}^n} f(\boldsymbol{x})$ admits a polynomial-time approximation algorithm with relative approximation ratio τ_H.*

The key linkage from multilinear tensor function $F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d)$ to the homogeneous polynomial function $f(\boldsymbol{x})$ is the following lemma. Essentially it makes the tensor relaxation method applicable for (H).

Lemma 3.4 (He, Li, and Zhang [19]) *Suppose $\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d \in \mathbb{R}^n$, and $\xi_1, \xi_2, \cdots, \xi_d$ are i.i.d. random variables, each taking values 1 and -1 with equal probability. For any super-symmetric d-th order tensor form $\boldsymbol{F}$ and function $f(\boldsymbol{x}) = F(\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x})$, it holds that*

$$\mathsf{E}\left[\prod_{i=1}^{d} \xi_i f\left(\sum_{k=1}^{d} \xi_k \boldsymbol{x}^k\right)\right] = d! F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d).$$

Remark that the approximation ratios for (H) hold under the square-free condition. This is because in this case the decision variables are actually in the multilinear form. Hence, one can replace any point in the box ($[-1, 1]^n$) by one of its vertices ($\{-1, 1\}^n$) without decreasing its objective function value, due to the linearity. Besides, in the case when d is odd, one may first relax (H) to $\max_{\boldsymbol{x} \in [-1,1]^n} f(\boldsymbol{x})$, and then directly apply the approximation result for homogeneous polynomial maximization over intersection of n co-centered ellipsoids (see [19]). Under the square-free condition, this procedure is able to generate a feasible solution for (H) with approximation ratio $\Omega\left(n^{-\frac{d-2}{2}} \log^{-(d-1)} n\right)$, which is worse than τ_H in Theorem 3.2. Therefore, we may treat Theorem 3.2 an improvement of the approximation ratio.

We move on to consider the mixed form of discrete polynomial optimization model (M). It is a generalization of (T) and (H), making the model applicable to a wider range of practical problems.

Theorem 3.5 *If $F(\underbrace{\boldsymbol{x}^1, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^1}_{d_1}, \underbrace{\boldsymbol{x}^2, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^2}_{d_2}, \cdots, \underbrace{\boldsymbol{x}^s, \boldsymbol{x}^s, \cdots, \boldsymbol{x}^s}_{d_s})$ is square-free in each $\boldsymbol{x}^k$ ($k = 1, 2, \ldots, s$), and one of d_k ($k = 1, 2, \ldots, s$) is odd, then $(M) : \max_{\boldsymbol{x}^k \in \mathbb{B}^{n_k}} f(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^s)$ admits a polynomial-time approximation algorithm with approximation ratio τ_M, where*

$$\tau_M := \begin{cases} \left(\frac{2}{\pi}\right)^{d-1} \ln\left(1 + \sqrt{2}\right) \prod_{k=1}^{s} d_k! d_k^{-d_k} \left(n_1^{d_1} n_2^{d_2} \cdots n_{s-2}^{d_{s-2}} n_{s-1}^{d_{s-1}-1}\right)^{-\frac{1}{2}} & d_s = 1, \\[2mm] \left(\frac{2}{\pi}\right)^{d-1} \ln\left(1 + \sqrt{2}\right) \prod_{k=1}^{s} d_k! d_k^{-d_k} \left(n_1^{d_1} n_2^{d_2} \cdots n_{s-1}^{d_{s-1}} n_s^{d_s-2}\right)^{-\frac{1}{2}} & d_s \geq 2. \end{cases}$$

Theorem 3.6 *If $F(\underbrace{\boldsymbol{x}^1, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^1}_{d_1}, \underbrace{\boldsymbol{x}^2, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^2}_{d_2}, \cdots, \underbrace{\boldsymbol{x}^s, \boldsymbol{x}^s, \cdots, \boldsymbol{x}^s}_{d_s})$ is square-free in each $\boldsymbol{x}^k$ ($k = 1, 2, \ldots, s$), and all d_k ($k = 1, 2, \ldots, s$) are even, then $(M) : \max_{\boldsymbol{x}^k \in \mathbb{B}^{n_k}} f(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^s)$ admits a polynomial-time approximation algorithm with relative approximation ratio τ_M.*

The main idea in the proof is tensor relaxation (to relax its objective function $f(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^s)$ to a multilinear tensor function), which leads to (T). After solving (T) approximately by Theorem 3.1, we are able to adjust the solutions one by one, using Lemma 3.4.

3.2 Homogeneous Polynomials in Mixed Variables

Proposition 3.7 *When $d = d' = 1$, $(T)' : \max_{\boldsymbol{x}^1 \in \mathbb{B}^{n_1}, \boldsymbol{y}^1 \in \mathbb{S}^{m_1}} F(\boldsymbol{x}^1, \boldsymbol{y}^1)$ admits a polynomial-time approximation algorithm with approximation ratio $\sqrt{2/\pi}$.*

Proposition 3.7 serves as the basis for $(T)'$ of general d and d'. In this particular case, $(T)'$ can be equivalently transformed into $\max_{\boldsymbol{x} \in \mathbb{B}^{n_1}} \boldsymbol{x}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{x}$ with $\boldsymbol{Q} \succeq 0$. The later problem admits a polynomial-time approximation algorithm (SDP relaxation and randomization) with approximation ratio $2/\pi$ by Nesterov [27].

Recursion is again the tool to handle the high degree case. For the recursion on d, with discrete variables $\boldsymbol{x}^k$, DR 3.1 is applied in each recursive step. For the recursion on d', with continuous variables $\boldsymbol{y}^k$, two decomposition routines in He, Li, and Zhang [19] are readily available, namely the eigenvalue decomposition approach (DR 2 of [19]) and the randomized decomposition approach (DR 1 of [19]), either one of them serves the purpose here.

Theorem 3.8 $(T)' : \max_{\boldsymbol{x}^k \in \mathbb{B}^{n_k}, \boldsymbol{y}^\ell \in \mathbb{S}^{m_\ell}} F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d, \boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^{d'})$ *admits a polynomial-time approximation algorithm with approximation ratio τ'_T, where*

$$\tau'_T := (2/\pi)^{\frac{2d-1}{2}} (n_1 n_2 \cdots n_{d-1} m_1 m_2 \cdots m_{d'-1})^{-\frac{1}{2}} = \Omega\left((n_1 n_2 \cdots n_{d-1} m_1 m_2 \cdots m_{d'-1})^{-\frac{1}{2}}\right).$$

From Theorem 3.8, by applying Lemma 3.4 as a linkage, together with the square-free property, we are led to the following two theorems regarding $(H)'$.

Theorem 3.9 *If $F(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{d}, \underbrace{\boldsymbol{y}, \boldsymbol{y}, \cdots, \boldsymbol{y}}_{d'})$ is square-free in $\boldsymbol{x}$, and either d or d' is odd, then $(H)' : \max_{\boldsymbol{x} \in \mathbb{B}^n, \boldsymbol{y} \in \mathbb{S}^m} f(\boldsymbol{x}, \boldsymbol{y})$ admits a polynomial-time approximation algorithm with approximation ratio τ'_H, where*

$$\tau'_H := d! d^{-d} d'! d'^{-d'} (2/\pi)^{\frac{2d-1}{2}} n^{-\frac{d-1}{2}} m^{-\frac{d'-1}{2}} = \Omega\left(n^{-\frac{d-1}{2}} m^{-\frac{d'-1}{2}}\right).$$

Theorem 3.10 *If $F(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{d}, \underbrace{\boldsymbol{y}, \boldsymbol{y}, \cdots, \boldsymbol{y}}_{d'})$ is square-free in $\boldsymbol{x}$, and both d and d' are even, then $(H)' : \max_{\boldsymbol{x} \in \mathbb{B}^n, \boldsymbol{y} \in \mathbb{S}^m} f(\boldsymbol{x}, \boldsymbol{y})$ admits a polynomial-time approximation algorithm with relative approximation ratio τ'_H.*

By relaxing $(M)'$ to the multilinear tensor function optimization $(T)'$ and solving it approximately using Theorem 3.8, we may further adjust its solution one by one using Lemma 3.4, leading to the following general result.

Theorem 3.11 *If $F(\underbrace{\boldsymbol{x}^1, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^1}_{d_1}, \cdots, \underbrace{\boldsymbol{x}^s, \boldsymbol{x}^s, \cdots, \boldsymbol{x}^s}_{d_s}, \underbrace{\boldsymbol{y}^1, \boldsymbol{y}^1, \cdots, \boldsymbol{y}^1}_{d'_1}, \cdots, \underbrace{\boldsymbol{y}^t, \boldsymbol{y}^t, \cdots, \boldsymbol{y}^t}_{d'_t})$ is square-free in each $\boldsymbol{x}^k$ $(k = 1, 2, \ldots, s)$, and one of d_k $(k = 1, 2, \ldots, s)$ or one of d'_ℓ $(\ell = 1, 2, \ldots, t)$ is odd, then $(M)' : \max_{\boldsymbol{x}^k \in \mathbb{B}^{n_k}, \boldsymbol{y}^\ell \in \mathbb{S}^{m_\ell}} f(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^s, \boldsymbol{y}^1, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^t)$ admits a polynomial-time approximation algorithm with approximation ratio τ'_M, where*

$$
\begin{aligned}
\tau'_M &:= \left(\tfrac{2}{\pi}\right)^{\frac{2d-1}{2}} \prod_{k=1}^{s} d_k! d_k^{-d_k} \prod_{\ell=1}^{t} d'_\ell! d'^{-d'_\ell}_\ell \left(n_1^{d_1} \cdots n_{s-1}^{d_{s-1}} n_s^{d_s-1} m_1^{d'_1} \cdots m_{t-1}^{d'_{t-1}} m_t^{d'_t-1}\right)^{-\frac{1}{2}} \\
&= \Omega\left(\left(n_1^{d_1} \cdots n_{s-1}^{d_{s-1}} n_s^{d_s-1} m_1^{d'_1} \cdots m_{t-1}^{d'_{t-1}} m_t^{d'_t-1}\right)^{-\frac{1}{2}}\right).
\end{aligned}
$$

Theorem 3.12 *If* $F(\underbrace{\boldsymbol{x}^1, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^1}_{d_1}, \cdots, \underbrace{\boldsymbol{x}^s, \boldsymbol{x}^s, \cdots, \boldsymbol{x}^s}_{d_s}, \underbrace{\boldsymbol{y}^1, \boldsymbol{y}^1, \cdots, \boldsymbol{y}^1}_{d_1'}, \cdots, \underbrace{\boldsymbol{y}^t, \boldsymbol{y}^t, \cdots, \boldsymbol{y}^t}_{d_t'})$ *is square-free in each* $\boldsymbol{x}^k$ $(k = 1, 2, \ldots, s)$, *and all* d_k $(k = 1, 2, \ldots, s)$ *and all* d_ℓ' $(\ell = 1, 2, \ldots, t)$ *are even, then* $(M)'$: $\max_{\boldsymbol{x}^k \in \mathbb{B}^{n_k}, \boldsymbol{y}^\ell \in \mathbb{S}^{m_\ell}} f(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^s, \boldsymbol{y}^1, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^t)$ *admits a polynomial-time approximation algorithm with relative approximation ratio* τ_M'.

3.3 Inhomogeneous Polynomials in Binary Variables

Extending the approximation algorithms and the corresponding analysis for *homogeneous* polynomial optimization to the general *inhomogeneous* polynomials is not straightforward. Technically it is also a way to get around the square-free property, which is a requirement for all the homogeneous polynomials mentioned in the previous subsections. The analysis here, like the analysis in our previous paper [20], is to directly deal with *homogenization*.

It is quite natural to introduce a new variable, say x_h, which is actually set to be 1, to yield a homogeneous form for Function P:

$$p(\boldsymbol{x}) = \sum_{k=1}^d F_k(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{k}) x_h^{d-k} + f_0 x_h^d := F\left(\underbrace{\begin{pmatrix} \boldsymbol{x} \\ x_h \end{pmatrix}, \begin{pmatrix} \boldsymbol{x} \\ x_h \end{pmatrix}, \cdots, \begin{pmatrix} \boldsymbol{x} \\ x_h \end{pmatrix}}_{d} \right) = F(\underbrace{\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}, \cdots, \bar{\boldsymbol{x}}}_{d}) = f(\bar{\boldsymbol{x}}),$$

where $f(\bar{\boldsymbol{x}})$ is an $(n+1)$-dimensional homogeneous polynomial function of degree d, with variable $\bar{\boldsymbol{x}}$, i.e., $\boldsymbol{F} \in \mathbb{R}^{(n+1)^d}$ and $\bar{\boldsymbol{x}} \in \mathbb{R}^{n+1}$. Optimization of this homogeneous form can be done due to our previous results, but in general we do not have any control on the solution of x_h, which has to be 1 as required by the feasibility. The following lemma ensures that construction of a feasible solution is possible.

Lemma 3.13 (He, Li, and Zhang [20]) *Suppose* $\bar{\boldsymbol{x}}^k = \begin{pmatrix} \boldsymbol{x}^k \\ x_h^k \end{pmatrix} \in \mathbb{R}^{n+1}$ *with* $|x_h^k| \leq 1$ *for* $k = 1, 2, \ldots, d$. *Let* $\eta_1, \eta_2, \cdots, \eta_d$ *be independent random variables, each taking values* 1 *and* -1 *with* $\mathsf{E}[\eta_k] = x_h^k$ *for* $k = 1, 2, \ldots, d$, *and let* $\xi_1, \xi_2, \cdots, \xi_d$ *be i.i.d. random variables, each taking values* 1 *and* -1 *with equal probability (thus the mean is 0). If the last component of the tensor* $\boldsymbol{F}$ *is* 0, *then we have*

$$\mathsf{E}\left[\prod_{k=1}^d \eta_k F\left(\begin{pmatrix} \eta_1 \boldsymbol{x}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \eta_2 \boldsymbol{x}^2 \\ 1 \end{pmatrix}, \cdots, \begin{pmatrix} \eta_d \boldsymbol{x}^d \\ 1 \end{pmatrix} \right) \right] = F(\bar{\boldsymbol{x}}^1, \bar{\boldsymbol{x}}^2, \cdots, \bar{\boldsymbol{x}}^d),$$

and

$$\mathsf{E}\left[F\left(\begin{pmatrix} \xi_1 \boldsymbol{x}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \xi_2 \boldsymbol{x}^2 \\ 1 \end{pmatrix}, \cdots, \begin{pmatrix} \xi_d \boldsymbol{x}^d \\ 1 \end{pmatrix} \right) \right] = 0.$$

Our last result is the following theorem.

Theorem 3.14 (P) *admits a polynomial-time approximation algorithm with relative approximation ratio* τ_P, *where*

$$\tau_P := \frac{\ln(1 + \sqrt{2})}{2(1 + e)\,\pi^{d-1}} (d+1)!\, d^{-2d} (n+1)^{-\frac{d-2}{2}} = \Omega\left(n^{-\frac{d-2}{2}} \right).$$

We remark that (P) is indeed a very general discrete optimization model. For example, it can be used to model the following general polynomial optimization problem in discrete values:

$$(D) \quad \max \quad p(\boldsymbol{x})$$
$$\text{s.t.} \quad x_i \in \{a_1^i, a_2^i, \cdots, a_{m_i}^i\}, \, i = 1, 2, \ldots, n.$$

To see this, we observe that by adopting the Lagrange interpolation technique and letting

$$x_i = \sum_{j=1}^{m_i} a_j^i \prod_{1 \le k \le m_i, \, k \ne j} \frac{u_i - k}{j - k}, \, i = 1, 2, \ldots, n,$$

the original decision variables can be equivalently transformed into

$$u_i = j \implies x_i = a_j^i, \, i = 1, 2, \ldots, n, \, j = 1, 2, \ldots, m_i,$$

where $u_i \in \{1, 2, \ldots, m_i\}$, which can be further represented by $\lceil \log_2 m_i \rceil$ independent binary variables. Combining these two steps of substitution, (D) is then reformulated as (P), with the degree of its objective polynomial function no larger than $\max_{1 \le i \le n} \{d(m_i - 1)\}$, and the dimension of its decision variables being $\sum_{i=1}^n \lceil \log_2 m_i \rceil$.

In many real world applications, the data $\{a_1^i, a_2^i, \cdots, a_{m_i}^i\}$ $(i = 1, 2, \ldots, n)$ in (D) are arithmetic sequences. Then it is much easier to transform (D) to (P), without going through the Lagrange interpolation. It keeps the same degree of the objective polynomial function, and the dimension of its decision variables is $\sum_{i=1}^n \lceil \log_2 m_i \rceil$.

The proofs of all the theorems presented in this section are delegated to Appendix A.

4   Examples of Application

As we discussed in Section 1, the models studied in this paper have versatile applications. Given the generic nature of the discrete polynomial optimization models (viz. (T), (H), (M), (P), $(T)'$, $(H)'$ and $(M)'$), this point is perhaps self-evident. However, we believe it is helpful to present a few examples at this point with more details, to illustrate the potential modeling opportunities with the new optimization models. We present four problems in this section and show that they are readily formulated by the discrete polynomial optimization models in this paper.

4.1   The Tensor Cut-Norm Problem

The concept of *cut-norm* is initially defined on a real matrix $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n_1 \times n_2}$, denoted by $\|\boldsymbol{A}\|_C$, the maximum over all $I \subseteq \{1, 2, \ldots, n_1\}$ and $J \subseteq \{1, 2, \ldots, n_2\}$, of the quantity $|\sum_{i \in I, j \in J} a_{ij}|$. This concept plays a major role in the design of efficient approximation algorithms for dense graph and matrix problems (see e.g., [14, 3]). Alon and Naor [2] proposed a randomized polynomial-time approximation algorithm that approximates the cut-norm with a factor at least 0.56, which is currently the best available approximation ratio. Since a matrix is a second order tensor, it is natural to extend the cut-norm to general higher order tensors, e.g., a recent paper by Kannan [23]. Specifically, given a d-th order tensor $\boldsymbol{F} = (a_{i_1 i_2 \cdots i_d}) \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, its cut-norm is defined by

$$\|\boldsymbol{F}\|_C := \max_{I_k \subseteq \{1,2,\ldots,n_k\}, \, k=1,2,\ldots,d} \left| \sum_{i_k \in I_k, \, k=1,2,\ldots,d} a_{i_1 i_2 \cdots i_d} \right|.$$

In fact, the cut-norm $\|\boldsymbol{F}\|_C$ is closely related to $\|\boldsymbol{F}\|_{\infty \mapsto 1}$, which is exactly in the form of (T). By Theorem 3.1, there is a polynomial-time approximation algorithm which computes $\|\boldsymbol{F}\|_{\infty \mapsto 1}$

with a factor at least $\Omega\left((n_1 n_2 \cdots n_{d-2})^{-\frac{1}{2}}\right)$. The following result, asserts that the cut-norm of a general d-th order tensor can also be approximated by a factor of $\Omega\left((n_1 n_2 \cdots n_{d-2})^{-\frac{1}{2}}\right)$.

Proposition 4.1 *For any d-th order tensor $\boldsymbol{F} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, $\|\boldsymbol{F}\|_C \leq \|\boldsymbol{F}\|_{\infty \mapsto 1} \leq 2^d \|\boldsymbol{F}\|_C$.*

Proof. Let $\boldsymbol{F} = (a_{i_1 i_2 \cdots i_d}) \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$. Recall that $\|\boldsymbol{F}\|_{\infty \mapsto 1} = \max_{\boldsymbol{x}^k \in \mathbb{B}^{n_k}, k=1,2,\ldots,d} F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d)$. Given any $\boldsymbol{x}^k \in \mathbb{B}^{n_k}$ for $k = 1, 2, \ldots, d$, it follows that

$$
\begin{aligned}
F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d) &= \sum_{1 \leq i_k \leq n_k,\, k=1,2,\ldots,d} a_{i_1 i_2 \cdots i_d} x_{i_1}^1 x_{i_2}^2 \cdots x_{i_d}^d \\
&= \sum_{\boldsymbol{\beta} \in \mathbb{B}^d} \sum_{i_k \in \{j | x_j^k = \beta_k, 1 \leq j \leq n_k\},\, k=1,2,\ldots,d} a_{i_1 i_2 \cdots i_d} x_{i_1}^1 x_{i_2}^2 \cdots x_{i_d}^d \\
&= \sum_{\boldsymbol{\beta} \in \mathbb{B}^d} \left(\prod_{1 \leq k \leq d} \beta_k \sum_{i_k \in \{j | x_j^k = \beta_k, 1 \leq j \leq n_k\},\, k=1,2,\ldots,d} a_{i_1 i_2 \cdots i_d} \right) \\
&\leq \sum_{\boldsymbol{\beta} \in \mathbb{B}^d} \left| \sum_{i_k \in \{j | x_j^k = \beta_k, 1 \leq j \leq n_k\},\, k=1,2,\ldots,d} a_{i_1 i_2 \cdots i_d} \right| \\
&\leq \sum_{\boldsymbol{\beta} \in \mathbb{B}^d} \|\boldsymbol{F}\|_C = 2^d \|\boldsymbol{F}\|_C,
\end{aligned}
$$

which implies $\|\boldsymbol{F}\|_{\infty \mapsto 1} \leq 2^d \|\boldsymbol{F}\|_C$.

Observe that $\|\boldsymbol{F}\|_C = \max_{\boldsymbol{z}^k \in \{0,1\}^{n_k}, k=1,2,\ldots,d} |F(\boldsymbol{z}^1, \boldsymbol{z}^2, \cdots, \boldsymbol{z}^d)|$. Given any $\boldsymbol{z}^k \in \{0,1\}^{n_k}$ for $k = 1, 2, \ldots, d$, let $\boldsymbol{z}^k = (\boldsymbol{e} + \boldsymbol{x}^k)/2$, where $\boldsymbol{e}$ is the all one vector. Clearly $\boldsymbol{x}^k \in \mathbb{B}^{n_k}$ for $k = 1, 2, \ldots, d$, and thus

$$
\begin{aligned}
F(\boldsymbol{z}^1, \boldsymbol{z}^2, \cdots, \boldsymbol{z}^d) &= F\left(\frac{\boldsymbol{e} + \boldsymbol{x}^1}{2}, \frac{\boldsymbol{e} + \boldsymbol{x}^2}{2}, \cdots, \frac{\boldsymbol{e} + \boldsymbol{x}^d}{2} \right) \\
&= \frac{F(\boldsymbol{e}, \boldsymbol{e}, \cdots, \boldsymbol{e}) + F(\boldsymbol{x}^1, \boldsymbol{e}, \cdots, \boldsymbol{e}) + \cdots + F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d)}{2^d} \\
&\leq \frac{1}{2^d} \cdot \|\boldsymbol{F}\|_{\infty \mapsto 1} \cdot 2^d = \|\boldsymbol{F}\|_{\infty \mapsto 1},
\end{aligned}
$$

which implies $\|\boldsymbol{F}\|_C \leq \|\boldsymbol{F}\|_{\infty \mapsto 1}$. $\qquad\square$

4.2 The Vector-Valued Maximum Cut Problem

Consider an undirected graph $G = (V, E)$ where $V = \{v_1, v_2, \cdots, v_n\}$ is the set of the vertices, and $E \subseteq V \times V$ is the set of the edges. On each edge $e \in E$ there is an associated weight, which is a nonnegative vector in this case, $\boldsymbol{w}_e \in \mathbb{R}_+^m$. The problem now is to find a cut in such a way that the total sum of the weights, which is a vector in this case, has a maximum norm. More formally, this problem can be formulated as

$$
\max_{C \text{ is a cut of } G} \left\| \sum_{e \in C} \boldsymbol{w}_e \right\|.
$$

13

Note that the usual max-cut problem is a special case of the above model where each weight $w_e \geq 0$ is a scalar. Similar to the scalar case (see [16]), we may reformulate the above problem in binary variables as

$$\max \quad \left\| \sum_{1 \leq i,j \leq n} x_i x_j \boldsymbol{w}'_{ij} \right\|$$
$$\text{s.t.} \quad \boldsymbol{x} \in \mathbb{B}^n,$$

where

$$\boldsymbol{w}'_{ij} = \begin{cases} -\boldsymbol{w}_{ij} & i \neq j, \\ \sum_{1 \leq k \leq n, k \neq i} \boldsymbol{w}_{ik} & i = j. \end{cases} \tag{1}$$

Observing the Cauchy-Schwartz inequality, we further formulate the above problem as

$$\max \quad \left(\sum_{1 \leq i,j \leq n} x_i x_j \boldsymbol{w}'_{ij} \right)^{\mathrm{T}} \boldsymbol{y} = F(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{y})$$
$$\text{s.t.} \quad \boldsymbol{x} \in \mathbb{B}^n, \, \boldsymbol{y} \in \mathbb{S}^m.$$

This is the exact form of $(H)'$ with $d = 2$ and $d' = 1$. Although the square-free property in terms of $\boldsymbol{x}$ does not hold in this model (which is a condition of Theorem 3.9), one can still replace any point in the box $([-1,1]^n)$ by one of its vertices $(\{-1,1\}^n)$ without decreasing its objective function value, since the matrix $F(\cdot, \cdot, \boldsymbol{e}_k) = \left((w'_{ij})_k \right)_{n \times n}$ is diagonal dominant for $k = 1, 2, \ldots, m$. Thus, the vector-valued max-cut problem admits an approximation ratio of $\frac{1}{2} \left(\frac{2}{\pi} \right)^{3/2} n^{-1/2}$ by Theorem 3.9.

If the weights on edges are positive semidefinite matrices (i.e., $\boldsymbol{W}_{ij} \in \mathbb{R}^{m \times m}$, $\boldsymbol{W}_{ij} \succeq \boldsymbol{0}$), then the matrix-valued max-cut problem can also be formulated as

$$\max \quad \lambda_{\max} \left(\sum_{1 \leq i,j \leq n} x_i x_j \boldsymbol{W}'_{ij} \right)$$
$$\text{s.t.} \quad \boldsymbol{x} \in \mathbb{B}^n,$$

where $\boldsymbol{W}'_{ij}$ is defined similarly to (1); or equivalently,

$$\max \quad \boldsymbol{y}^{\mathrm{T}} \left(\sum_{1 \leq i,j \leq n} x_i x_j \boldsymbol{W}'_{ij} \right) \boldsymbol{y}$$
$$\text{s.t.} \quad \boldsymbol{x} \in \mathbb{B}^n, \, \boldsymbol{y} \in \mathbb{S}^m.$$

Similar to the vector-valued case, by the diagonal dominant property and Theorem 3.10, the above problem admits an approximation ratio of $\frac{1}{4} \left(\frac{2}{\pi} \right)^{3/2} (mn)^{-1/2}$. Notice that Theorem 3.10 only asserts a relative approximation ratio; however for this problem the optimal value of its minimization counterpart is obviously nonnegative, and thus a relative approximation ratio implies a usual approximation ratio.

4.3 The Maximum Complete Satisfiability Problem

The usual maximum satisfiability problem (see e.g., [15]) is to find the boolean values of the literals, so as to maximize the total weighted sum of the satisfied clauses. The key point of the problem is that each clause is in the *disjunctive* form, namely if one of the literals is assigned the *ture* value, then the clause is called satisfied. If the literals are also *conjunctive*, then this form of satisfiability problem is easy to solve. However, if not all the clauses can be satisfied, and we alternatively look for an assignment that maximizes the weighted sum of the satisfied clauses, then the problem is quite different. To make a distinction from the usual Max-SAT problem, let us call the new problem to be *maximum complete satisfiability* problem, to be abbreviated as Max-C-SAT. It is immediately clear that Max-C-SAT is NP-hard, since we can easily reduce the max-cut problem to

14

it. The reduction can be done as follows. For each edge (v_i, v_j) we consider two clauses $\{x_i, \bar{x}_j\}$ and $\{\bar{x}_i, x_j\}$, both having weight w_{ij}. Then a Max-C-SAT solution leads to a solution to the max-cut problem.

Now consider an instance of the Max-C-SAT problem with m clauses, each clause containing no more than d literals. Suppose that clause k $(k = 1, 2, \ldots, m)$ has the following form

$$\{x_{k_1}, x_{k_2}, \cdots, x_{k_{s_k}}, \bar{x}_{\bar{k}_1}, \bar{x}_{\bar{k}_2}, \ldots, \bar{x}_{\bar{k}_{t_k}}\},$$

where $s_k + t_k \leq d$, associated with a weight $w_k \geq 0$ for $k = 1, 2, \ldots, m$. Then, the Max-C-SAT problem can be formulated in the form of (P) as

$$\begin{array}{ll} \max & \sum_{k=1}^m w_k \prod_{i=1}^{s_k} \frac{1+x_{k_i}}{2} \cdot \prod_{j=1}^{t_k} \frac{1-x_{\bar{k}_j}}{2} \\ \text{s.t.} & \boldsymbol{x} \in \mathbb{B}^n. \end{array}$$

According to Theorem 3.14 and the nonnegativity of the objective function, the above problem admits a polynomial-time approximation algorithm with approximation ratio $\Omega\left(n^{-\frac{d-2}{2}}\right)$, which is independent of the number of clauses m.

4.4 The Box Constrained Diophantine Equation

Solving a system of linear equations where the variables are integers and constrained to a box is an important problem in discrete optimization and linear algebra. Examples of application include the classical Frobenius problem (see e.g., [6]), and a "market split problem" [11], other from engineering applications in integrated circuits design and video signal processing. For more details, one is referred to Aardal et al. [1]. Essentially, the problem is to find an integer-valued $\boldsymbol{x} \in \mathbb{Z}^n$ and $\boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{u}$, such that $\boldsymbol{Ax} = \boldsymbol{b}$. The problem can be formulated by the least square method as

$$\begin{array}{lll} (L) & \max & -(\boldsymbol{Ax} - \boldsymbol{b})^{\mathrm{T}}(\boldsymbol{Ax} - \boldsymbol{b}) \\ & \text{s.t.} & \boldsymbol{x} \in \mathbb{Z}^n, \, \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{u}. \end{array}$$

According to the discussion at the end of Section 3.3, the above problem can be reformulated as a form of (P), whose objective function is quadratic polynomial and number of decision variables is $\sum_{i=1}^n \lceil \log_2(u_i + 1) \rceil$. By applying Theorem 3.14, (L) admits a polynomial-time approximation algorithm with a constant relative approximation ratio.

In general, the Diophantine equations are polynomial equations. The box constrained polynomial equations can also be formulated by the least square method as of (L). Suppose the highest degree of the polynomial equations is d. Then, this least square problem can be reformulated as a form of (P), with the degree of the objective polynomial being $2d$ and number of decision variables being $\sum_{i=1}^n \lceil \log_2(u_i + 1) \rceil$. By applying Theorem 3.14, this problem admits a polynomial-time approximation algorithm with a relative approximation ratio $\Omega\left(\left(\sum_{i=1}^n \log u_i\right)^{-(d-1)}\right)$.

We have tested extensively the numerical performance of the algorithms proposed in this paper, based on simulated data. In general the results show that the algorithms are not only efficient in the theoretical sense as we prove in this paper, but also effective in practice. The numerical results and the discussion of these results under various circumstances, however, are too lengthy to be included in the current paper. Instead, we refer the interested readers to the recent Ph.D. thesis of one of the authors, Li [26].

References

[1] K. Aardal, C. A. J. Hurkens, and A. K. Lenstra, *Solving A System of Linear Diophantine Equations with Lower and Upper Bounds on the Variables*, Mathematics of Operations Research, 25, 427–442, 2000.

[2] N. Alon and A. Naor, *Approximating the Cut-Norm via Grothendieck's Inequality*, SIAM Journal on Computing, 35, 787–803, 2006.

[3] N. Alon, W. F. de la Vega, R. Kannan, and M. Karpinski, *Random Sampling and Approximation of MAX-CSP Problems*, Proceedings of the 34th Annuals ACM Symposium on Theory of Computing, 232–239, 2002.

[4] N. Ansari and E. Hou, *Computational Intelligence for Optimization*, Kluwer Academic Publishing Group, Norwell, MA, 1997.

[5] M. L. Balinski, *On A Selection Problem*, Management Sciences, 17, 230–231, 1970.

[6] D. Beihoffer, J. Hendry, A. Nijenhuis, and S. Wagon, *Faster Algorithms for Frobenius Numbers*, The Electronic Journal of Combinatorics, 12, R27, 2005.

[7] D. Bertsimas and Y. Ye, *Semidefinite Relaxations, Multivariate Normal Distributions, and Order Statistics*, in D.-Z. Du and P. M. Pardalos (Eds.), Handbook of Combinatorial Optimization, Kluwer Academic Publishers, Norwell, MA, Volumn 3, 1–19, 1998.

[8] J. Bruck and M. Blaum, *Neural Networks, Error-Correcting Codes, and Polynomials over the Binary n-Cube*, IEEE Transactions on Information Theory, 35, 976–987, 1989.

[9] M. Charikar and A. Wirth, *Maximizing Quadratic Programs: Extending Grothendieck's Inequality*, Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, 54–60, 2004.

[10] B. Chen, S. He, Z. Li, and S. Zhang, *Maximum Block Improvement and Polynomial Optimization*, SIAM Journal on Optimization, 22, 87–107, 2012.

[11] G. Cornuéjols and M. Dawande, *A Class of Hard Small 0-1 Programs*, Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science, 1412, 284–293, 1998.

[12] U. Feige and E. Ofek, *Easily Refutable Subformulas of Large Random 3CNF Forumlas*, Automata, Languages and Programming, Lecture Notes in Compuer Science, 3142, 519–530, 2004.

[13] J. Friedman, A. Goerdt, and M. Krivelevich, *Recognizing More Unsatisfiable Random k-SAT Instances Efficiently*, SIAM Journal on Computing, 35, 408–430, 2005.

[14] A. M. Frieze and R. Kannan, *Quick Approximation to Matrices and Applications*, Combinatorica, 19, 175–200, 1999.

[15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman & Co., New York, NY, 1979.

[16] M. X. Goemans and D. P. Williamson, *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems using Semidefinite Programming*, Journal of the ACM, 42, 1115–1145, 1995.

[17] P. L. Hammer and S. Rudeanu, *Boolean Methods in Operations Research*, Springer-Verlag, New York, NY, 1968.

[18] P. Hansen, *Methods of Nonlinear 0-1 Programming*, Annals of Discrete Mathematics, 5, 53–70, 1979.

[19] S. He, Z. Li, and S. Zhang, *Approximation Algorithms for Homogeneous Polynomial Optimization with Quadratic Constraints*, Mathematical Programming, 125, 353–383, 2010.

[20] S. He, Z. Li, and S. Zhang, *General Constrained Polynomial Optimization: An Approximation Approach*, Technical Report, Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, 2010.

[21] J. J. Hopfield and D. W. Tank, *"Neural" Computation of Decisions in Optimization Problem*, Biological Cybernetics, 52, 141–152, 1985.

[22] Y. Huang and S. Zhang, *Approximation Algorithms for Indefinite Complex Quadratic Maximization Problems*, Science China Mathematics, 53, 2697–2708, 2010.

[23] R. Kannan, *Spectral Methods for Matrices and Tensors*, Proceedings of the 42nd Annual ACM Symposium on Theory of Computing, 1–12, 2010.

[24] S. Khot and A. Naor, *Linear Equations Modulo 2 and the L_1 Diameter of Convex Bodies*, Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, 318–328, 2007.

[25] E. Kofidis and Ph. Regalia, *On the Best Rank-1 Approximation of Higher Order Supersymmetric Tensors*, SIAM Journal on Matrix Analysis and Applications, 23, 863–884, 2002.

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

[27] Yu. Nesterov, *Semidefinite Relaxation and Nonconvex Quadratic Optimization*, Optimization Methods and Softwares, 9, 141–160, 1998.

[28] Yu. Nesterov, *Random Walk in a Simplex and Quadratic Optimization over Convex Polytopes*, CORE Discussion Paper, UCL, Louvain-la-Neuve, Belgium, 2003.

[29] M. Purser, *Introduction to Error-Correcting Codes*, Artech House Inc., Norwood, MA, 1995.

[30] J. M. W. Rhys, *A Selection Problem of Shared Fixed Costs and Network Flows*, Management Science, 17, 200–207, 1970.

[31] A. M.-C. So, *Deterministic Approximation Algorithms for Sphere Constrained Homogeneous Polynomial Optimization Problems*, Mathematical Programming, 129, 357–382, 2011.

A   Proofs of the Theorems

A.1   Proof of Theorem 3.1

Proof. The proof is based on mathematical induction on the degree d. For the case of $d = 2$, it is exactly the algorithm by Alon and Naor [2]. For general $d \geq 3$, let $\boldsymbol{X} = \boldsymbol{x}^1(\boldsymbol{x}^d)^{\mathrm{T}}$ and (T) is then

relaxed to

$$
\begin{aligned}
(\hat{T}) \quad \max \quad & F(\boldsymbol{X}, \boldsymbol{x}^2, \boldsymbol{x}^3 \cdots, \boldsymbol{x}^{d-1}) \\
\text{s.t.} \quad & \boldsymbol{X} \in \mathbb{B}^{n_1 n_d}, \\
& \boldsymbol{x}^k \in \mathbb{B}^{n_k}, \ k = 2, 3, \ldots, d-1,
\end{aligned}
$$

where we treat $\boldsymbol{X}$ as an $n_1 n_d$-dimensional vector, and $\boldsymbol{F} \in \mathbb{R}^{n_1 n_d \times n_2 \times \cdots \times n_{d-1}}$ as a $(d-1)$-th order tensor. Observe that $(\hat{T})$ is the exact form of (T) in degree $d-1$, and so by induction we can find $\hat{\boldsymbol{X}} \in \mathbb{B}^{n_1 n_d}$ and $\hat{\boldsymbol{x}}^k \in \mathbb{B}^{n_k}$ $(k = 2, 3, \ldots, d-1)$ in polynomial-time, such that

$$
F\left(\hat{\boldsymbol{X}}, \hat{\boldsymbol{x}}^2, \hat{\boldsymbol{x}}^3, \ldots, \hat{\boldsymbol{x}}^{d-1}\right) \geq (2/\pi)^{d-2} \ln(1 + \sqrt{2}) \, (n_2 n_3 \cdots n_{d-2})^{-\frac{1}{2}} \, v_{\max}(\hat{T})
$$

$$
\geq (2/\pi)^{d-2} \ln(1 + \sqrt{2}) \, (n_2 n_3 \cdots n_{d-2})^{-\frac{1}{2}} \, v_{\max}(T).
$$

Rewrite $\hat{\boldsymbol{X}}$ as an $n_1 \times n_d$ matrix, and construct

$$
\tilde{\boldsymbol{X}} = \left[\begin{array}{cc} \boldsymbol{I}_{n_1 \times n_1} & \hat{\boldsymbol{X}}/\sqrt{n_1} \\ \hat{\boldsymbol{X}}^{\mathrm{T}}/\sqrt{n_1} & \hat{\boldsymbol{X}}^{\mathrm{T}} \hat{\boldsymbol{X}}/n_1 \end{array} \right] \succeq 0 \tag{2}
$$

as in DR 3.1, and then randomly generate

$$
\binom{\boldsymbol{\xi}}{\boldsymbol{\eta}} \sim \mathcal{N}(\boldsymbol{0}_{n_1 + n_d}, \tilde{\boldsymbol{X}}).
$$

Let $\hat{\boldsymbol{x}}^1 := \operatorname{sign}(\boldsymbol{\xi})$ and $\hat{\boldsymbol{x}}^d := \operatorname{sign}(\boldsymbol{\eta})$. Noticing that the diagonal components of $\tilde{\boldsymbol{X}}$ are all ones, it follows from Bertsimas and Ye [7] that for all $1 \leq i \leq n_1$ and $1 \leq j \leq n_d$,

$$
\mathsf{E}\left[\hat{x}_i^1 \hat{x}_j^d\right] = \frac{2}{\pi} \arcsin \frac{\hat{X}_{ij}}{\sqrt{n_1}} = \frac{2}{\pi} \hat{X}_{ij} \arcsin \frac{1}{\sqrt{n_1}},
$$

where the last equality is due to $|\hat{X}_{ij}| = 1$. Denote matrix $\hat{\boldsymbol{Q}} = F(\cdot, \hat{\boldsymbol{x}}^2, \hat{\boldsymbol{x}}^3, \cdots, \hat{\boldsymbol{x}}^{d-1}, \cdot)$, then

$$
\begin{aligned}
\mathsf{E}\left[F\left(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d\right)\right] &= \mathsf{E}\left[\sum_{1 \leq i \leq n_1, 1 \leq j \leq n_d} \hat{x}_i^1 \hat{Q}_{ij} \hat{x}_j^d\right] \\
&= \sum_{1 \leq i \leq n_1, 1 \leq j \leq n_d} \hat{Q}_{ij} \, \mathsf{E}\left[\hat{x}_i^1 \hat{x}_j^d\right] \\
&= \sum_{1 \leq i \leq n_1, 1 \leq j \leq n_d} \hat{Q}_{ij} \frac{2}{\pi} \hat{X}_{ij} \arcsin \frac{1}{\sqrt{n_1}} \\
&= \frac{2}{\pi} \arcsin \frac{1}{\sqrt{n_1}} \sum_{1 \leq i \leq n_1, 1 \leq j \leq n_d} \hat{Q}_{ij} \hat{X}_{ij} \\
&= \frac{2}{\pi} \arcsin \frac{1}{\sqrt{n_1}} F\left(\hat{\boldsymbol{X}}, \hat{\boldsymbol{x}}^2, \hat{\boldsymbol{x}}^3, \cdots, \hat{\boldsymbol{x}}^{d-1}\right) \tag{3} \\
&\geq \frac{2}{\pi \sqrt{n_1}} (2/\pi)^{d-2} \ln(1 + \sqrt{2}) \, (n_2 n_3 \cdots n_{d-2})^{-\frac{1}{2}} \, v_{\max}(T) \\
&= (2/\pi)^{d-1} \ln(1 + \sqrt{2}) \, (n_1 n_2 \cdots n_{d-2})^{-\frac{1}{2}} \, v_{\max}(T).
\end{aligned}
$$

Therefore $\hat{\boldsymbol{x}}^1$ and $\hat{\boldsymbol{x}}^d$ can be found by randomization, which concludes the induction step. $\qquad \square$

Lemma A.1 *If a polynomial function $p(\boldsymbol{x})$ is square-free and $\boldsymbol{z} \in [-1,1]^n$, then $\boldsymbol{x}' \in \mathbb{B}^n$ and $\boldsymbol{x}'' \in \mathbb{B}^n$ can be found in polynomial-time, such that $p(\boldsymbol{x}') \leq p(\boldsymbol{z}) \leq p(\boldsymbol{x}'')$.*

Proof. Since $p(\boldsymbol{x})$ is square-free, by fixing $x_2, x_3, \cdots, x_n$ as constants and taking x_1 as the variable, we may rewrite

$$p(\boldsymbol{x}) = g_1(x_2, x_3, \cdots, x_n) + x_1 g_2(x_2, x_3, \cdots, x_n).$$

Let

$$x_1' = \begin{cases} -1 & g_2(z_2, z_3, \cdots, z_n) \geq 0, \\ 1 & g_2(z_2, z_3, \cdots, z_n) < 0. \end{cases}$$

Then

$$p\left((x_1', z_2, z_3, \cdots, z_n)^{\mathrm{T}}\right) \leq p(\boldsymbol{z}).$$

Repeat the same procedures for $z_2, z_3, \cdots, z_n$, and let them be replaced by $x_2', x_3', \cdots, x_n'$ respectively. Then $\boldsymbol{x}' = (x_1', x_2', \cdots, x_n')^{\mathrm{T}} \in \mathbb{B}^n$ satisfies $p(\boldsymbol{x}') \leq p(\boldsymbol{z})$. Using a similar argument, we may find $\boldsymbol{x}'' \in \mathbb{B}^n$ with $p(\boldsymbol{x}'') \geq p(\boldsymbol{z})$. $\qquad\square$

A.2 Proof of Theorem 3.2

Proof. Let $f(\boldsymbol{x}) = F(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{d})$ with $\boldsymbol{F}$ being super-symmetric. (H) can be relaxed to

$$(\tilde{T}) \quad \max \quad F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d)$$
$$\text{s.t.} \quad \boldsymbol{x}^k \in \mathbb{B}^n, \ k = 1, 2, \ldots, d.$$

By Theorem 3.1 we are able to find a set of binary vectors $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d)$ in polynomial-time, such that

$$F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d) \geq (2/\pi)^{d-1} \ln(1 + \sqrt{2}) \, n^{-\frac{d-2}{2}} v_{\max}(\tilde{T}) \geq (2/\pi)^{d-1} \ln(1 + \sqrt{2}) \, n^{-\frac{d-2}{2}} v_{\max}(H).$$

When d is odd, let $\xi_1, \xi_2, \cdots, \xi_d$ be i.i.d. random variables, each taking values 1 and -1 with equal probability. Then by Lemma 3.4 it follows that

$$d! F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d) = \mathsf{E}\left[\prod_{i=1}^d \xi_i f\left(\sum_{k=1}^d \xi_k \hat{\boldsymbol{x}}^k\right)\right] = \mathsf{E}\left[f\left(\sum_{k=1}^d \left(\prod_{i \neq k} \xi_i\right)\hat{\boldsymbol{x}}^k\right)\right].$$

Thus we may find a binary vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_d)^{\mathrm{T}} \in \mathbb{B}^d$, such that

$$f\left(\sum_{k=1}^d \left(\prod_{i \neq k} \beta_i\right)\hat{\boldsymbol{x}}^k\right) \geq d! F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \ldots, \hat{\boldsymbol{x}}^d) \geq d!(2/\pi)^{d-1} \ln(1 + \sqrt{2}) \, n^{-\frac{d-2}{2}} v_{\max}(H).$$

Now we notice that $\frac{1}{d} \sum_{k=1}^d \left(\prod_{i \neq k} \beta_i\right)\hat{\boldsymbol{x}}^k \in [-1,1]^n$, because for all $1 \leq j \leq n$,

$$\left|\left(\frac{1}{d} \sum_{k=1}^d \left(\prod_{i \neq k} \beta_i\right)\hat{\boldsymbol{x}}^k\right)_j\right| = \frac{1}{d}\left|\sum_{k=1}^d \left(\prod_{i \neq k} \beta_i\right)\hat{x}_j^k\right| \leq \frac{1}{d}\sum_{k=1}^d \left|\left(\prod_{i \neq k} \beta_i\right)\hat{x}_j^k\right| = 1. \qquad (4)$$

Since $f(\boldsymbol{x})$ is square-free, by Lemma A.1 we are able to find $\tilde{\boldsymbol{x}} \in \mathbb{B}^n$ in polynomial-time, such that

$$f(\tilde{\boldsymbol{x}}) \geq f\left(\frac{1}{d} \sum_{k=1}^d \left(\prod_{i \neq k} \beta_i\right)\hat{\boldsymbol{x}}^k\right) \geq d^{-d} d!(2/\pi)^{d-1} \ln(1 + \sqrt{2}) \, n^{-\frac{d-2}{2}} v_{\max}(H).$$

$\qquad\square$

Lemma A.2 *Suppose in (P): $\max_{\boldsymbol{x}\in\mathbb{B}^n} p(\boldsymbol{x})$, the objective polynomial function $p(\boldsymbol{x})$ is square-free and has no constant term. Then $v_{\min}(P) \leq 0 \leq v_{\max}(P)$, and a binary vector $\boldsymbol{x}' \in \mathbb{B}^n$ can be found in polynomial-time with $p(\boldsymbol{x}') \geq 0$.*

Proof. Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)^{\mathrm{T}}$, whose components are i.i.d. random variables and take values 1 and -1 with equal probability. Then for any term $a_{i_1 i_2 \ldots i_k} x_{i_1} x_{i_2} \cdots x_{i_k}$ with degree k $(1 \leq k \leq d)$ of $p(\boldsymbol{x})$, by the square-free property, it follows

$$\mathsf{E}[a_{i_1 i_2 \ldots i_k} \xi_{i_1} \xi_{i_2} \cdots \xi_{i_k}] = a_{i_1 i_2 \cdots i_k} \mathsf{E}[\xi_{i_1}] \mathsf{E}[\xi_{i_2}] \cdots \mathsf{E}[\xi_{i_k}] = 0.$$

This implies $\mathsf{E}[p(\boldsymbol{\xi})] = 0$, and consequently $v_{\min}(P) \leq 0 \leq v_{\max}(P)$. By a randomization process, a binary vector $\boldsymbol{x}' \in \mathbb{B}^n$ can be found in polynomial-time with $p(\boldsymbol{x}') \geq 0$. $\qquad\square$

We remark that the second part of Lemma A.2 can also be proven by conducting the procedure in Lemma A.1 with the input vector $\mathbf{0} \in [-1, 1]^n$, since $p(\mathbf{0}) = 0$. Therefore, finding a binary vector $\boldsymbol{x}' \in \mathbb{B}^n$ with $p(\boldsymbol{x}') \geq 0$ can be done by either a randomized process (Lemma A.2) or a deterministic process (Lemma A.1).

A.3 Proof of Theorem 3.3

Proof. Like in the proof of Theorem 3.2, by relaxing (H) to $(\tilde{T})$, we are able to find a set of binary vectors $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d)$ with

$$F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d) \geq (2/\pi)^{d-1} \ln(1 + \sqrt{2})\, n^{-\frac{d-2}{2}} v_{\max}(\tilde{T}).$$

Besides, we observe that $v_{\max}(H) \leq v_{\max}(\tilde{T})$ and $v_{\min}(H) \geq v_{\min}(\tilde{T}) = -v_{\max}(\tilde{T})$. Therefore

$$2\, v_{\max}(\tilde{T}) \geq v_{\max}(H) - v_{\min}(H).$$

Let $\xi_1, \xi_2, \cdots, \xi_d$ be i.i.d. random variables, each taking values 1 and -1 with equal probability. Use a similar argument of (4), we have $\frac{1}{d} \sum_{k=1}^{d} \xi_k \hat{\boldsymbol{x}}^k \in [-1, 1]^n$. Then by Lemma A.1, there exists $\hat{\boldsymbol{x}} \in \mathbb{B}^n$ such that

$$f\left(\frac{1}{d} \sum_{k=1}^{d} \xi_k \hat{\boldsymbol{x}}^k\right) \geq f(\hat{\boldsymbol{x}}) \geq v_{\min}(H).$$

Applying Lemma 3.4 and we have

$$\frac{1}{2}\, \mathsf{E}\left[f\left(\frac{1}{d} \sum_{k=1}^{d} \xi_k \hat{\boldsymbol{x}}^k\right) - v_{\min}(H) \,\middle|\, \prod_{i=1}^{d} \xi_i = 1\right]$$

$$\geq \frac{1}{2}\, \mathsf{E}\left[f\left(\frac{1}{d} \sum_{k=1}^{d} \xi_k \hat{\boldsymbol{x}}^k\right) - v_{\min}(H) \,\middle|\, \prod_{i=1}^{d} \xi_i = 1\right] - \frac{1}{2}\, \mathsf{E}\left[f\left(\frac{1}{d} \sum_{k=1}^{d} \xi_k \hat{\boldsymbol{x}}^k\right) - v_{\min}(H) \,\middle|\, \prod_{i=1}^{d} \xi_i = -1\right]$$

$$= \mathsf{E}\left[\prod_{i=1}^{d} \xi_i \left(f\left(\frac{1}{d} \sum_{k=1}^{d} \xi_k \hat{\boldsymbol{x}}^k\right) - v_{\min}(H)\right)\right]$$

$$= d^{-d} \mathsf{E}\left[\prod_{i=1}^{d} \xi_i f\left(\sum_{k=1}^{d} \xi_k \hat{\boldsymbol{x}}^k\right)\right] - v_{\min}(H)\, \mathsf{E}\left[\prod_{i=1}^{d} \xi_i\right]$$

$$= d^{-d} d!\, F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \ldots, \hat{\boldsymbol{x}}^d) \geq \tau_H\, v_{\max}(\tilde{T}) \geq (\tau_H/2)\,(v_{\max}(H) - v_{\min}(H)),$$

20

where the last inequality is due to $2\, v_{\max}(\tilde{T}) \geq v_{\max}(H) - v_{\min}(H)$. Thus we may find a binary vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_d)^{\mathrm{T}} \in \mathbb{B}^d$ with $\prod_{i=1}^d \beta_i = 1$, such that

$$f\left(\frac{1}{d}\sum_{k=1}^d \beta_k \hat{\boldsymbol{x}}^k\right) - v_{\min}(H) \geq \tau_H\left(v_{\max}(H) - v_{\min}(H)\right).$$

Noticing $\frac{1}{d}\sum_{k=1}^d \beta_k \hat{\boldsymbol{x}}^k \in [-1, 1]^n$ and applying Lemma A.1, by the square-free property of $f(\boldsymbol{x})$, we are able to find $\tilde{\boldsymbol{x}} \in \mathbb{B}^n$ with

$$f(\tilde{\boldsymbol{x}}) - v_{\min}(H) \geq f\left(\frac{1}{d}\sum_{k=1}^d \beta_k \hat{\boldsymbol{x}}^k\right) - v_{\min}(H) \geq \tau_H\left(v_{\max}(H) - v_{\min}(H)\right).$$

$\square$

A.4 Proof of Theorem 3.5

Proof. Like in the proof of Theorem 3.2, by relaxing model (M) to (T), we are able to find a set of binary vectors $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d)$ with

$$F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d) \geq \tau_M \left(\prod_{k=1}^s \frac{d_k^{\,d_k}}{d_k!}\right) v_{\max}(M).$$

Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_d)^{\mathrm{T}}$, whose components are i.i.d. random variables taking values 1 and -1 with equal probability. Denote

$$\hat{\boldsymbol{x}}^1_\xi := \sum_{k=1}^{d_1} \xi_k \hat{\boldsymbol{x}}^k,\ \hat{\boldsymbol{x}}^2_\xi := \sum_{k=d_1+1}^{d_1+d_2} \xi_k \hat{\boldsymbol{x}}^k, \cdots, \hat{\boldsymbol{x}}^s_\xi := \sum_{k=d_1+d_2+\cdots+d_{s-1}+1}^{d} \xi_k \hat{\boldsymbol{x}}^k. \tag{5}$$

Without loss of generality, we assume d_1 to be odd. By applying Lemma 3.4 d times, it is easy to verify that

$$d_1! d_2! \cdots d_s! F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d) = \mathsf{E}\left[\prod_{i=1}^d \xi_i f\left(\hat{\boldsymbol{x}}^1_\xi, \hat{\boldsymbol{x}}^2_\xi, \cdots, \hat{\boldsymbol{x}}^s_\xi\right)\right] = \mathsf{E}\left[f\left(\prod_{i=1}^d \xi_i \hat{\boldsymbol{x}}^1_\xi, \hat{\boldsymbol{x}}^2_\xi, \cdots, \hat{\boldsymbol{x}}^s_\xi\right)\right].$$

Thus we are able to find $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_d)^{\mathrm{T}} \in \mathbb{B}^d$, such that

$$f\left(\prod_{i=1}^d \beta_i \frac{\hat{\boldsymbol{x}}^1_\beta}{d_1}, \frac{\hat{\boldsymbol{x}}^2_\beta}{d_2}, \cdots, \frac{\hat{\boldsymbol{x}}^s_\beta}{d_s}\right) \geq \prod_{k=1}^s d_k! d_k^{\,-d_k} F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d) \geq \tau_M\, v_{\max}(M).$$

It is easy to verify that $\prod_{i=1}^d \beta_i \hat{\boldsymbol{x}}^1_\beta / d_1 \in [-1, 1]^n$, and $\hat{\boldsymbol{x}}^k_\beta / d_k \in [-1, 1]^n$ for $k = 2, 3, \ldots s$. By the square-free property of the function f and applying Lemma A.1, we are able to find a set of binary vectors $(\tilde{\boldsymbol{x}}^1, \tilde{\boldsymbol{x}}^2, \cdots, \tilde{\boldsymbol{x}}^s)$ in polynomial-time, such that

$$f(\tilde{\boldsymbol{x}}^1, \tilde{\boldsymbol{x}}^2, \cdots, \tilde{\boldsymbol{x}}^s) \geq f\left(\prod_{i=1}^d \beta_i \frac{\hat{\boldsymbol{x}}^1_\beta}{d_1}, \frac{\hat{\boldsymbol{x}}^2_\beta}{d_2}, \cdots, \frac{\hat{\boldsymbol{x}}^s_\beta}{d_s}\right) \geq \tau_M\, v_{\max}(M).$$

$\square$

A.5 Proof of Theorem 3.6

Proof. The proof is analogous to the proof of Theorem 3.3. The main differences are: (i) we use

$$d_1! d_2! \cdots d_s! F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d) = \mathsf{E}\left[\prod_{i=1}^d \xi_i f\left(\hat{\boldsymbol{x}}_\xi^1, \hat{\boldsymbol{x}}_\xi^2, \cdots, \hat{\boldsymbol{x}}_\xi^s\right)\right]$$

instead of invoking Lemma 3.4 directly, where $\hat{\boldsymbol{x}}_\xi^k$ ($k = 1, 2, \ldots, s$) is defined by (5); and (ii) we use $f\left(\frac{1}{d_1}\hat{\boldsymbol{x}}_\xi^1, \frac{1}{d_2}\hat{\boldsymbol{x}}_\xi^2, \cdots, \frac{1}{d_s}\hat{\boldsymbol{x}}_\xi^s\right)$ instead of $f\left(\frac{1}{d}\sum_{k=1}^d \xi_k \hat{\boldsymbol{x}}^k\right)$ during the randomization process. □

A.6 Proof of Proposition 3.7

Proof. When $d = d' = 1$, $(T)'$ can be written as

$$\begin{aligned}\max \quad & \boldsymbol{x}^\mathrm{T} \boldsymbol{Q} \boldsymbol{y} \\ \text{s.t.} \quad & \boldsymbol{x} \in \mathbb{B}^{n_1}, \, \boldsymbol{y} \in \mathbb{S}^{m_1}.\end{aligned}$$

For any fixed $\boldsymbol{x}$, the corresponding optimal $\boldsymbol{y}$ must be $\boldsymbol{Q}^\mathrm{T}\boldsymbol{x}/\|\boldsymbol{Q}^\mathrm{T}\boldsymbol{x}\|$ due to the Cauchy-Schwartz inequality, and accordingly,

$$\boldsymbol{x}^\mathrm{T} \boldsymbol{Q} \boldsymbol{y} = \boldsymbol{x}^\mathrm{T} \boldsymbol{Q} \frac{\boldsymbol{Q}^\mathrm{T}\boldsymbol{x}}{\|\boldsymbol{Q}^\mathrm{T}\boldsymbol{x}\|} = \|\boldsymbol{Q}^\mathrm{T}\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^\mathrm{T} \boldsymbol{Q} \boldsymbol{Q}^\mathrm{T} \boldsymbol{x}}.$$

Thus the problem is equivalent to $\max_{\boldsymbol{x} \in \mathbb{B}^{n_1}} \boldsymbol{x}^\mathrm{T} \boldsymbol{Q} \boldsymbol{Q}^\mathrm{T} \boldsymbol{x}$. Noticing that $\boldsymbol{Q} \boldsymbol{Q}^\mathrm{T}$ is positive semidefinite, by the result of Nesterov [27], it admits an approximation ratio $2/\pi$. Thus the original problem admits a polynomial-time approximation algorithm with approximation ratio $\sqrt{2/\pi}$. □

Proposition A.3 (He, Li, and Zhang [19]) (S) : $\max_{\boldsymbol{y}^1 \in \mathbb{S}^{m_1}, \boldsymbol{y}^2 \in \mathbb{S}^{m_2}} (\boldsymbol{y}^1)^\mathrm{T} \boldsymbol{Q} \boldsymbol{y}^2$ *can be solved in polynomial-time, with* $v_{\max}(S) \geq \|\boldsymbol{Q}\|/\sqrt{m_1}$.

A.7 Proof of Theorem 3.8

Proof. The proof is based on mathematical induction on the degree $d + d'$. Proposition 3.7 can be used as the base for the induction process when $d + d' = 2$.

For general $d + d' \geq 3$, if $d' \geq 2$, let $\boldsymbol{Y} = \boldsymbol{y}^1 (\boldsymbol{y}^{d'})^\mathrm{T}$. Noticing that $\|\boldsymbol{Y}\|^2 = \|\boldsymbol{y}^1\|^2 \|\boldsymbol{y}^{d'}\|^2 = 1$, $(T)'$ can be relaxed to a case with degree $d + d' - 1$, i.e.,

$$\begin{aligned}\max \quad & F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d, \boldsymbol{Y}, \boldsymbol{y}^2, \boldsymbol{y}^3, \cdots, \boldsymbol{y}^{d'-1}) \\ \text{s.t.} \quad & \boldsymbol{x}^k \in \mathbb{B}^{n_k}, \, k = 1, 2, \ldots, d, \\ & \boldsymbol{Y} \in \mathbb{S}^{m_1 m_{d'}}, \, \boldsymbol{y}^\ell \in \mathbb{S}^{m_\ell}, \, \ell = 2, 3, \ldots, d' - 1.\end{aligned}$$

By induction, a feasible solution $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{Y}}, \hat{\boldsymbol{y}}^2, \hat{\boldsymbol{y}}^3, \cdots, \hat{\boldsymbol{y}}^{d'-1})$ can be found in polynomial-time, such that

$$F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{Y}}, \hat{\boldsymbol{y}}^2, \hat{\boldsymbol{y}}^3, \cdots, \hat{\boldsymbol{y}}^{d'-1}) \geq (2/\pi)^{\frac{2d-1}{2}} (n_1 n_2 \cdots n_{d-1} m_2 m_3 \cdots m_{d'-1})^{-\frac{1}{2}} v_{\max}(T').$$

Let us denote $\boldsymbol{Q} = F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \cdot, \hat{\boldsymbol{y}}^2, \hat{\boldsymbol{y}}^3, \cdots, \hat{\boldsymbol{y}}^{d'-1}, \cdot) \in \mathbb{R}^{m_1 \times m_{d'}}$. Then by Proposition A.3 (used in DR 2 of [19]), the problem $\max_{\boldsymbol{y}^1 \in \mathbb{S}^{m_1}, \boldsymbol{y}^{d'} \in \mathbb{S}^{m_{d'}}} (\boldsymbol{y}^1)^\mathrm{T} \boldsymbol{Q} \boldsymbol{y}^{d'}$ can be solved in polynomial-time, with its optimal solution $(\hat{\boldsymbol{y}}^1, \hat{\boldsymbol{y}}^{d'})$ satisfying

$$F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{y}}^1, \hat{\boldsymbol{y}}^2, \cdots, \hat{\boldsymbol{y}}^{d'}) = (\hat{\boldsymbol{y}}^1)^\mathrm{T} \boldsymbol{Q} \hat{\boldsymbol{y}}^{d'} \geq \|\boldsymbol{Q}\|/\sqrt{m_1}.$$

By the Cauchy-Schwartz inequality, it follows that

$$\|\boldsymbol{Q}\| = \max_{\boldsymbol{Y} \in \mathbb{S}^{m_1 m_{d'}}} \boldsymbol{Q} \bullet Y \geq \boldsymbol{Q} \bullet \hat{\boldsymbol{Y}} = F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{Y}}, \hat{\boldsymbol{y}}^2, \hat{\boldsymbol{y}}^3, \cdots, \hat{\boldsymbol{y}}^{d'-1}).$$

Thus we concludes that

$$\begin{aligned}
F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{y}}^1, \hat{\boldsymbol{y}}^2, \cdots, \hat{\boldsymbol{y}}^{d'}) &\geq \|\boldsymbol{Q}\|/\sqrt{m_1} \\
&\geq F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{Y}}, \hat{\boldsymbol{y}}^2, \hat{\boldsymbol{y}}^3, \ldots, \hat{\boldsymbol{y}}^{d'-1})/\sqrt{m_1} \\
&\geq \tau_T' \, v_{\max}(T').
\end{aligned}$$

For $d + d' \geq 3$ and $d' = 1$, let $\boldsymbol{X} = \boldsymbol{x}^1(\boldsymbol{x}^d)^{\mathrm{T}}$. $(T)'$ can be relaxed to the other case with degree $d - 1 + d'$, i.e.,

$$\begin{aligned}
\max \quad & F(\boldsymbol{X}, \boldsymbol{x}^2, \boldsymbol{x}^3, \cdots, \boldsymbol{x}^{d-1}, \boldsymbol{y}^1, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^{d'}) \\
\text{s.t.} \quad & \boldsymbol{X} \in \mathbb{B}^{n_1 n_d}, \; \boldsymbol{x}^k \in \mathbb{B}^{n_k}, \; k = 2, 3, \ldots, d-1, \\
& \boldsymbol{y}^\ell \in \mathbb{S}^{m_\ell}, \; \ell = 1, 2, \ldots, d'.
\end{aligned}$$

By induction, the problem admits a polynomial-time approximation algorithm with approximation ratio $(2/\pi)^{\frac{2d-3}{2}}(n_2 n_3 \cdots n_{d-1} m_1 m_2 \cdots m_{d'-1})^{-\frac{1}{2}}$. In order to decompose $\boldsymbol{X}$ into $\boldsymbol{x}^1$ and $\boldsymbol{x}^d$, we shall first conduct the randomization procedure as (2) (DR 3.1) in the proof of Theorem 3.1, which will further deteriorate by an additional factor of $\frac{2}{\pi\sqrt{n_1}}$ as shown in (3). Combining these two factors, we are led to the ratio τ_T'. $\qquad \square$

A.8 Proof of Theorem 3.9

Proof. Like in the proof of Theorem 3.2, by relaxing model $(H)'$ to $(T)'$, we are able to find $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{y}}^1, \hat{\boldsymbol{y}}^2, \cdots, \hat{\boldsymbol{y}}^{d'})$ with $\hat{\boldsymbol{x}}^k \in \mathbb{B}^n$ for all $1 \leq k \leq d$ and $\hat{\boldsymbol{y}}^\ell \in \mathbb{S}^m$ for all $1 \leq \ell \leq d'$, such that

$$F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{y}}^1, \hat{\boldsymbol{y}}^2, \cdots, \hat{\boldsymbol{y}}^{d'}) \geq (2/\pi)^{\frac{2d-1}{2}} n^{-\frac{d-1}{2}} m^{-\frac{d'-1}{2}} v_{\max}(H').$$

Let $\xi_1, \xi_2, \cdots, \xi_d, \eta_1, \eta_2, \cdots, \eta_{d'}$ be i.i.d. random variables, each taking values 1 and -1 with equal probability. By applying Lemma 3.4 twice, we have

$$\mathsf{E}\left[\prod_{i=1}^d \xi_i \prod_{j=1}^{d'} \eta_j \, f\left(\sum_{k=1}^d \xi_k \hat{\boldsymbol{x}}^k, \sum_{\ell=1}^{d'} \eta_\ell \hat{\boldsymbol{y}}^\ell\right)\right] = d! d'! F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{y}}^1, \hat{\boldsymbol{y}}^2, \cdots, \hat{\boldsymbol{y}}^{d'}). \tag{6}$$

Thus we are able to find $\boldsymbol{\beta} \in \mathbb{B}^d$ and $\boldsymbol{\beta}' \in \mathbb{B}^{d'}$, such that

$$\prod_{i=1}^d \beta_i \prod_{j=1}^{d'} \beta_j' \, f\left(\sum_{k=1}^d \beta_k \hat{\boldsymbol{x}}^k, \sum_{\ell=1}^{d'} \beta_\ell' \hat{\boldsymbol{y}}^\ell\right) \geq d! d'! F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{y}}^1, \hat{\boldsymbol{y}}^2, \cdots, \hat{\boldsymbol{y}}^{d'}).$$

If d is odd, let $\hat{\boldsymbol{x}} = \prod_{i=1}^d \beta_i \prod_{j=1}^{d'} \beta_j' \sum_{k=1}^d \beta_k \hat{\boldsymbol{x}}^k$ and $\hat{\boldsymbol{y}} = \sum_{\ell=1}^{d'} \beta_\ell' \hat{\boldsymbol{y}}^\ell$; otherwise let $\hat{\boldsymbol{x}} = \sum_{k=1}^d \beta_k \hat{\boldsymbol{x}}^k$ and $\hat{\boldsymbol{y}} = \prod_{i=1}^d \beta_i \prod_{j=1}^{d'} \beta_j' \sum_{\ell=1}^{d'} \beta_\ell' \hat{\boldsymbol{y}}^\ell$. Noticing $\|\hat{\boldsymbol{y}}\| \leq d'$ and combining the previous two inequalities, it follows that

$$f\left(\frac{\hat{\boldsymbol{x}}}{d}, \frac{\hat{\boldsymbol{y}}}{\|\hat{\boldsymbol{y}}\|}\right) \geq d^{-d} d'^{-d'} \prod_{i=1}^d \beta_i \prod_{j=1}^{d'} \beta_j' f\left(\sum_{k=1}^d \beta_k \hat{\boldsymbol{x}}^k, \sum_{\ell=1}^{d'} \beta_\ell' \hat{\boldsymbol{y}}^\ell\right) \geq \tau_H' \, v_{\max}(H').$$

Denote $\tilde{\boldsymbol{y}} = \hat{\boldsymbol{y}}/\|\hat{\boldsymbol{y}}\| \in \mathbb{S}^m$. Since $\hat{\boldsymbol{x}}/d \in [-1, 1]^n$ and $f(\boldsymbol{x}, \boldsymbol{y})$ is square-free in $\boldsymbol{x}$, by applying Lemma A.1, $\tilde{\boldsymbol{x}} \in \mathbb{B}^n$ can be found in polynomial-time, with

$$f(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \geq f(\hat{\boldsymbol{x}}/d, \tilde{\boldsymbol{y}}) \geq \tau_H' \, v_{\max}(H').$$

$\qquad \square$

A.9 Proof of Theorem 3.10

Proof. Following the same argument as in the proof of Theorem 3.9, we shall get (6), which implies

$$\mathsf{E}\left[\prod_{i=1}^{d}\xi_i\prod_{j=1}^{d'}\eta_j\,f\left(\sum_{k=1}^{d}\xi_k\hat{\boldsymbol{x}}^k,\sum_{\ell=1}^{d'}\eta_\ell\hat{\boldsymbol{y}}^\ell\right)\right]\geq d!d'!(2/\pi)^{\frac{2d-1}{2}}n^{-\frac{d-1}{2}}m^{-\frac{d'-1}{2}}v_{\max}(H').$$

Denote $\hat{\boldsymbol{x}}_\xi:=\frac{1}{d}\sum_{k=1}^{d}\xi_k\hat{\boldsymbol{x}}^k$ and $\hat{\boldsymbol{y}}_\eta:=\frac{1}{d'}\sum_{\ell=1}^{d'}\eta_\ell\hat{\boldsymbol{y}}^\ell$. Clearly we have

$$\mathsf{E}\left[\prod_{i=1}^{d}\xi_i\prod_{j=1}^{d'}\eta_j\,f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta\right)\right]\geq\tau'_H\,v_{\max}(H').$$

Pick any fixed $\boldsymbol{y}'\in\mathbb{S}^m$ and consider the following problem

$$(\hat{H})\quad\max\quad f(\boldsymbol{x},\boldsymbol{y}')$$
$$\text{s.t.}\quad\boldsymbol{x}\in\mathbb{B}^n.$$

Since $f(\boldsymbol{x},\boldsymbol{y}')$ is square-free in $\boldsymbol{x}$ and has no constant term, by Lemma A.2, a binary vector $\boldsymbol{x}'\in\mathbb{B}^n$ can be found in polynomial-time with $f(\boldsymbol{x}',\boldsymbol{y}')\geq 0\geq v_{\min}(\hat{H})\geq v_{\min}(H')$.

Next we shall argue $f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta\right)\geq v_{\min}(H')$. If this were not the case, then $f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta\right)<v_{\min}(H')\leq 0$. By noticing $\|\hat{\boldsymbol{y}}_\eta\|\leq 1$, this leads to

$$f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta/\|\hat{\boldsymbol{y}}_\eta\|\right)=\|\hat{\boldsymbol{y}}_\eta\|^{-d'}f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta\right)\leq f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta\right)<v_{\min}(H').$$

Also noticing $\hat{\boldsymbol{x}}_\xi\in[-1,1]^n$, by applying Lemma A.1, a binary vector $\hat{\boldsymbol{x}}\in\mathbb{B}^n$ can be found with

$$v_{\min}(H')\leq f(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}_\eta/\|\hat{\boldsymbol{y}}_\eta\|)\leq f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta/\|\hat{\boldsymbol{y}}_\eta\|\right)<v_{\min}(H'),$$

resulting in a contradiction.

By that $f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta\right)-v_{\min}(H')\geq 0$, it follows

$$\frac{1}{2}\,\mathsf{E}\left[f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta\right)-v_{\min}(H')\,\middle|\,\prod_{i=1}^{d}\xi_i\prod_{j=1}^{d'}\eta_j=1\right]$$
$$\geq\frac{1}{2}\,\mathsf{E}\left[f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta\right)-v_{\min}(H')\,\middle|\,\prod_{i=1}^{d}\xi_i\prod_{j=1}^{d'}\eta_j=1\right]-\frac{1}{2}\,\mathsf{E}\left[f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta\right)-v_{\min}(H')\,\middle|\,\prod_{i=1}^{d}\xi_i\prod_{j=1}^{d'}\eta_j=-1\right]$$
$$=\mathsf{E}\left[\prod_{i=1}^{d}\xi_i\prod_{j=1}^{d'}\eta_j\left(f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta\right)-v_{\min}(H')\right)\right]$$
$$=\mathsf{E}\left[\prod_{i=1}^{d}\xi_i\prod_{j=1}^{d'}\eta_j\,f\left(\hat{\boldsymbol{x}}_\xi,\hat{\boldsymbol{y}}_\eta\right)\right]\geq\tau'_H\,v_{\max}(H').$$

Thus we are able to find $\boldsymbol{\beta}\in\mathbb{B}^d$ and $\boldsymbol{\beta}'\in\mathbb{B}^{d'}$ with $\prod_{i=1}^{d}\beta_i\prod_{j=1}^{d'}\beta'_j=1$, such that

$$f\left(\hat{\boldsymbol{x}}_\beta,\hat{\boldsymbol{y}}_{\beta'}\right)-v_{\min}(H')\geq 2\,\tau'_H\,v_{\max}(H').$$

24

Denote $\boldsymbol{y}'' = \hat{\boldsymbol{y}}_{\beta'}/\|\hat{\boldsymbol{y}}_{\beta'}\| \in \mathbb{S}^m$. Since $\hat{\boldsymbol{x}}_\beta \in [-1,1]^n$, by Lemma A.1, a binary vector $\boldsymbol{x}'' \in \mathbb{B}^n$ can be found in polynomial-time with $f(\boldsymbol{x}'', \boldsymbol{y}'') \geq f(\hat{\boldsymbol{x}}_\beta, \boldsymbol{y}'')$. Below we shall prove either $(\boldsymbol{x}', \boldsymbol{y}')$ or $(\boldsymbol{x}'', \boldsymbol{y}'')$ will satisfy

$$f(\boldsymbol{x}, \boldsymbol{y}) - v_{\min}(H') \geq \tau'_H \left(v_{\max}(H') - v_{\min}(H') \right). \tag{7}$$

Indeed, if $-v_{\min}(H') \geq \tau'_H \left(v_{\max}(H') - v_{\min}(H') \right)$, then $(\boldsymbol{x}', \boldsymbol{y}')$ satisfies (7) in this case since $f(\boldsymbol{x}', \boldsymbol{y}') \geq 0$. Otherwise, if $-v_{\min}(H') < \tau'_H \left(v_{\max}(H') - v_{\min}(H') \right)$, then

$$v_{\max}(H') > \left(1 - \tau'_H \right) \left(v_{\max}(H') - v_{\min}(H') \right) \geq \left(v_{\max}(H') - v_{\min}(H') \right)/2,$$

which implies

$$f\left(\hat{\boldsymbol{x}}_\beta, \hat{\boldsymbol{y}}_{\beta'} \right) - v_{\min}(H') \geq 2\,\tau'_H\, v_{\max}(H') \geq \tau'_H \left(v_{\max}(H') - v_{\min}(H') \right).$$

The above inequality also implies that $f\left(\hat{\boldsymbol{x}}_\beta, \hat{\boldsymbol{y}}_{\beta'} \right) > 0$. Therefore, we have

$$f(\boldsymbol{x}'', \boldsymbol{y}'') \geq f\left(\hat{\boldsymbol{x}}_\beta, \boldsymbol{y}'' \right) = \|\hat{\boldsymbol{y}}_{\beta'}\|^{-d'} f\left(\hat{\boldsymbol{x}}_\beta, \hat{\boldsymbol{y}}_{\beta'} \right) \geq f\left(\hat{\boldsymbol{x}}_\beta, \hat{\boldsymbol{y}}_{\beta'} \right),$$

implying $(\boldsymbol{x}'', \boldsymbol{y}'')$ satisfies (7). Finally, $\arg\max\{f(\boldsymbol{x}', \boldsymbol{y}'), f(\boldsymbol{x}'', \boldsymbol{y}'')\}$ satisfies (7) in both cases. $\square$

A.10 Proof of Theorem 3.11

Proof. The proof is analogous to the proof of Theorem 3.9. We first relax $(M)'$ to $(T)'$ and get an approximate solution $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{y}}^1, \hat{\boldsymbol{y}}^2, \cdots, \hat{\boldsymbol{y}}^{d'})$ using Theorem 3.8. By applying Lemma 3.4 $s + t$ times, we have

$$\mathsf{E}\left[\prod_{i=1}^d \xi_i \prod_{j=1}^{d'} \eta_j f\left(\hat{\boldsymbol{x}}_\xi^1, \hat{\boldsymbol{x}}_\xi^2, \cdots, \hat{\boldsymbol{x}}_\xi^s, \hat{\boldsymbol{y}}_\eta^1, \hat{\boldsymbol{y}}_\eta^2, \cdots, \hat{\boldsymbol{y}}_\eta^t \right) \right] = \prod_{k=1}^s d_k! \prod_{\ell=1}^t d'_\ell! F\left(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{y}}^1, \hat{\boldsymbol{y}}^2, \cdots, \hat{\boldsymbol{y}}^{d'} \right),$$

where

$$\hat{\boldsymbol{x}}_\xi^1 := \sum_{k=1}^{d_1} \xi_k \hat{\boldsymbol{x}}^k, \ \hat{\boldsymbol{x}}_\xi^2 := \sum_{k=d_1+1}^{d_1+d_2} \xi_k \hat{\boldsymbol{x}}^k, \ \cdots, \ \hat{\boldsymbol{x}}_\xi^s := \sum_{k=d_1+d_2+\cdots+d_{s-1}+1}^{d} \xi_k \hat{\boldsymbol{x}}^k,$$

and

$$\hat{\boldsymbol{y}}_\eta^1 := \sum_{\ell=1}^{d'_1} \eta_\ell \hat{\boldsymbol{y}}^\ell, \ \hat{\boldsymbol{y}}_\eta^2 := \sum_{\ell=d'_1+1}^{d'_1+d'_2} \eta_\ell \hat{\boldsymbol{y}}^\ell, \ \cdots, \ \hat{\boldsymbol{y}}_\eta^t := \sum_{\ell=d'_1+d'_2+\cdots+d'_{t-1}+1}^{d'} \eta_\ell \hat{\boldsymbol{y}}^\ell.$$

In the above identity, as one of d_k $(k = 1, 2, \ldots, s)$ or one of d'_ℓ $(\ell = 1, 2, \ldots, t)$ is odd, we are able to move $\prod_{i=1}^d \xi_i \prod_{j=1}^{d'} \eta_j$ into the coefficient of the corresponding vector ($\hat{\boldsymbol{x}}_\xi^k$ or $\hat{\boldsymbol{y}}_\eta^\ell$ whenever appropriate) in the function f. Other derivations are essentially the same as the proof of Theorem 3.9. $\square$

A.11 Proof of Theorem 3.12

Proof. The proof is analogous to that of Theorem 3.10. The main differences are: (i) we use

$$\mathsf{E}\left[\prod_{i=1}^d \xi_i \prod_{j=1}^{d'} \eta_j f\left(\hat{\boldsymbol{x}}_\xi^1, \hat{\boldsymbol{x}}_\xi^2, \cdots, \hat{\boldsymbol{x}}_\xi^s, \hat{\boldsymbol{y}}_\eta^1, \hat{\boldsymbol{y}}_\eta^2, \cdots, \hat{\boldsymbol{y}}_\eta^t \right) \right] = \prod_{k=1}^s d_k! \prod_{\ell=1}^t d'_\ell! F\left(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \hat{\boldsymbol{y}}^1, \hat{\boldsymbol{y}}^2, \cdots, \hat{\boldsymbol{y}}^{d'} \right)$$

instead of (6); and (ii) we use $f\left(\frac{\hat{\boldsymbol{x}}_\xi^1}{d_1}, \frac{\hat{\boldsymbol{x}}_\xi^2}{d_2}, \cdots, \frac{\hat{\boldsymbol{x}}_\xi^s}{d_s}, \frac{\hat{\boldsymbol{y}}_\eta^1}{d'_1}, \frac{\hat{\boldsymbol{y}}_\eta^2}{d'_2}, \cdots, \frac{\hat{\boldsymbol{y}}_\eta^t}{d'_t} \right)$ instead of $f\left(\hat{\boldsymbol{x}}_\xi, \hat{\boldsymbol{y}}_\eta \right)$. $\square$

A.12 Proof of Theorem 3.14

Proof. We may without loss of generality assume $p(\boldsymbol{x})$ is square-free since we have $(x_i)^2 = 1$ for $i = 1, 2, \ldots, n$, which allows us to reduce the power of x_i to 0 or 1. We may further assume $p(\boldsymbol{x})$ to have no constant term. Thus by homogenization

$$p(\boldsymbol{x}) = \sum_{k=1}^{d} F_k(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{k}) x_h^{d-k} := F\left(\underbrace{\begin{pmatrix} \boldsymbol{x} \\ x_h \end{pmatrix}, \begin{pmatrix} \boldsymbol{x} \\ x_h \end{pmatrix}, \cdots, \begin{pmatrix} \boldsymbol{x} \\ x_h \end{pmatrix}}_{d}\right) = F(\underbrace{\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}, \cdots, \bar{\boldsymbol{x}}}_{d}) = f(\bar{\boldsymbol{x}}), \quad (8)$$

where $f(\bar{\boldsymbol{x}}) = p(\boldsymbol{x})$ if $x_h = 1$, and $f(\bar{\boldsymbol{x}})$ is an $(n+1)$-dimensional homogeneous polynomial function of degree d. During this proof, the 'bar' notation, e.g., $\bar{\boldsymbol{x}}$, is reserved for an $(n+1)$-dimensional vector, with the underlying letter $\boldsymbol{x}$ referring to the vector of its first n components, and the subscript 'h' (the subscript of x_h) referring to its last component.

Two immediate observations are in order here. First, $f(\bar{\boldsymbol{x}})$ is square-free with respect to all the variables $x_1, x_2, \cdots, x_n$, but is not square-free with respect to x_h. Second, the last component of the tensor form $\boldsymbol{F}$ is 0, since there is no constant term in the polynomial $p(\boldsymbol{x})$.

(P) is then equivalent to

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

which can be relaxed to an instance of (T) as follows

$$(\bar{T}) \quad \begin{aligned} \max \quad & F(\bar{\boldsymbol{x}}^1, \bar{\boldsymbol{x}}^2, \cdots, \bar{\boldsymbol{x}}^d) \\ \text{s.t.} \quad & \bar{\boldsymbol{x}}^k \in \mathbb{B}^{n+1}, k = 1, 2, \ldots, d. \end{aligned}$$

Let $(\bar{\boldsymbol{u}}^1, \bar{\boldsymbol{u}}^2, \cdots, \bar{\boldsymbol{u}}^d)$ be the feasible solution of $(\bar{T})$ found by Theorem 3.1 with

$$\begin{aligned} F(\bar{\boldsymbol{u}}^1, \bar{\boldsymbol{u}}^2, \cdots, \bar{\boldsymbol{u}}^d) &\geq (n+1)^{-\frac{d-2}{2}} (2/\pi)^{d-1} \ln(1+\sqrt{2}) \, v_{\max}(\bar{T}) \\ &\geq (n+1)^{-\frac{d-2}{2}} (2/\pi)^{d-1} \ln(1+\sqrt{2}) \, v_{\max}(P). \end{aligned}$$

Denote $\bar{\boldsymbol{v}}^k := \bar{\boldsymbol{u}}^k/d$ for all $1 \leq k \leq d$, and consequently

$$F(\bar{\boldsymbol{v}}^1, \bar{\boldsymbol{v}}^2, \cdots, \bar{\boldsymbol{v}}^d) = d^{-d} F(\bar{\boldsymbol{u}}^1, \bar{\boldsymbol{u}}^2, \cdots, \bar{\boldsymbol{u}}^d) \geq d^{-d}(2/\pi)^{d-1} \ln(1+\sqrt{2}) \, (n+1)^{-\frac{d-2}{2}} v_{\max}(P).$$

Notice that for all $1 \leq k \leq d$, $|v_h^k| = |u_h^k/d| = 1/d \leq 1$ and the last component of tensor $\boldsymbol{F}$ is 0. By applying Lemma 3.13, it follows that

$$\mathsf{E}\left[\prod_{k=1}^{d} \eta_k F\left(\begin{pmatrix} \eta_1 \boldsymbol{v}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \eta_2 \boldsymbol{v}^2 \\ 1 \end{pmatrix}, \cdots, \begin{pmatrix} \eta_d \boldsymbol{v}^d \\ 1 \end{pmatrix}\right)\right] = F(\bar{\boldsymbol{v}}^1, \bar{\boldsymbol{v}}^2, \cdots, \bar{\boldsymbol{v}}^d),$$

and

$$\mathsf{E}\left[F\left(\begin{pmatrix} \xi_1 \boldsymbol{v}^1 \\ 1 \end{pmatrix}, \begin{pmatrix} \xi_2 \boldsymbol{v}^2 \\ 1 \end{pmatrix}, \cdots, \begin{pmatrix} \xi_d \boldsymbol{v}^d \\ 1 \end{pmatrix}\right)\right] = 0,$$

where $(\eta_1, \eta_2, \ldots, \eta_d) = \boldsymbol{\eta}^{\mathrm{T}}$ are independent random variables, taking values 1 and -1 with $\mathsf{E}[\eta_k] = v_h^k$ for all $1 \leq k \leq d$, and $(\xi_1, \xi_2, \cdots, \xi_d) = \boldsymbol{\xi}^{\mathrm{T}}$ are i.i.d. random variables, taking values 1 and

26

-1 with equal probability. By combining these two identities, we have, for any constant c, the following identity

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

If we let $c = \max_{\boldsymbol{\beta} \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = -1} \text{Prob}\{\boldsymbol{\eta} = \boldsymbol{\beta}\}$, then in the above identity, the coefficient of each term $F(\cdot)$ is nonnegative. Therefore we are able to find $\boldsymbol{\beta}' = (\beta_1', \beta_2', \cdots, \beta_d')^{\text{T}} \in \mathbb{B}^d$ such that

$$F\left(\begin{pmatrix}\beta_1' \boldsymbol{v}^1 \\ 1\end{pmatrix}, \begin{pmatrix}\beta_2' \boldsymbol{v}^2 \\ 1\end{pmatrix}, \cdots, \begin{pmatrix}\beta_d' \boldsymbol{v}^d \\ 1\end{pmatrix}\right) \geq \tau_0 F(\bar{\boldsymbol{v}}^1, \bar{\boldsymbol{v}}^2, \cdots, \bar{\boldsymbol{v}}^d),$$

with

$$\tau_0 := \left(\sum_{\boldsymbol{\beta} \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = 1} (c + \text{Prob}\{\boldsymbol{\eta} = \boldsymbol{\beta}\}) + \sum_{\boldsymbol{\beta} \in \mathbb{B}^d, \prod_{k=1}^d \beta_k = -1} (c - \text{Prob}\{\boldsymbol{\eta} = \boldsymbol{\beta}\})\right)^{-1}$$
$$\geq \left(2^d c + 1\right)^{-1} \geq \left(2^d \left(\frac{1}{2} + \frac{1}{2d}\right)^d + 1\right)^{-1} \geq \frac{1}{1 + e},$$

where $c \leq \left(\frac{1}{2} + \frac{1}{2d}\right)^d$ is applied because $\mathsf{E}[\eta_k] = v_h^k = \pm 1/d$ for all $1 \leq k \leq d$.

Let us denote $\bar{\boldsymbol{z}}^k = \begin{pmatrix}\boldsymbol{z}^k \\ z_h^k\end{pmatrix} := \begin{pmatrix}\beta_k' \boldsymbol{v}^k \\ 1\end{pmatrix}$ for all $1 \leq k \leq d$, and we have

$$F(\bar{\boldsymbol{z}}^1, \bar{\boldsymbol{z}}^2, \cdots, \bar{\boldsymbol{z}}^d) \geq \tau_0 F(\bar{\boldsymbol{v}}^1, \bar{\boldsymbol{v}}^2, \cdots, \bar{\boldsymbol{v}}^d) \geq \frac{\ln(1 + \sqrt{2})}{1 + e} \left(\frac{2}{\pi}\right)^{d-1} d^{-d}(n + 1)^{-\frac{d-2}{2}} v_{\max}(P). \quad (9)$$

For any $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_d)^{\text{T}} \in \mathbb{B}^d$, denote

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

By noticing $z_h^k = 1$ and $|z_i^k| = |v_i^k| = |u_i^k|/d = 1/d$ for all $1 \leq k \leq d$ and $1 \leq i \leq n$, it follows that

$$2 \leq |z_h(\beta)| \leq 2d \text{ and } |z_i(\beta)| \leq (d + 1)/d + (d - 1)/d = 2, \forall 1 \leq i \leq n.$$

Thus $\boldsymbol{z}(\beta)/z_h(\beta) \in [-1, 1]^n$. By Lemma A.1, a binary vector $\boldsymbol{x}' \in \mathbb{B}^n$ can be found, such that

$$v_{\min}(P) \leq p(\boldsymbol{x}') \leq p\left(\boldsymbol{z}(\beta)/z_h(\beta)\right) = f\left(\bar{\boldsymbol{z}}(\beta)/z_h(\beta)\right).$$

Moreover, we shall argue below that

$$\beta_1 = 1 \implies f(\bar{\boldsymbol{z}}(\beta)) \geq (2d)^d v_{\min}(P). \quad (10)$$

If this were not the case, then $f\left(\bar{\boldsymbol{z}}(\beta)/(2d)\right) < v_{\min}(P) \leq 0$ (by Lemma A.2). Notice that $\beta_1 = 1$ implies $z_h(\beta) > 0$, and thus we have

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

27

which is a contradiction.

Suppose $(\xi_1, \xi_2, \cdots, \xi_d) = \boldsymbol{\xi}^{\mathrm{T}}$ are i.i.d. random variables, taking values 1 and -1 with equal probability. By Lemma 3.4 it follows that

$$
\begin{aligned}
d!F\left((d+1)\bar{\boldsymbol{z}}^1, \bar{\boldsymbol{z}}^2, \cdots, \bar{\boldsymbol{z}}^d\right) &= \mathsf{E}\left[\prod_{k=1}^d \xi_k f\left(\bar{\boldsymbol{z}}(\xi)\right)\right] \\
&= \frac{1}{4}\mathsf{E}\left[f\left(\bar{\boldsymbol{z}}(\xi)\right)\,\bigg|\,\xi_1 = 1, \prod_{k=2}^d \xi_k = 1\right] - \frac{1}{4}\mathsf{E}\left[f\left(\bar{\boldsymbol{z}}(\xi)\right)\,\bigg|\,\xi_1 = 1, \prod_{k=2}^d \xi_k = -1\right] \\
&\quad - \frac{1}{4}\mathsf{E}\left[f\left(\bar{\boldsymbol{z}}(\xi)\right)\,\bigg|\,\xi_1 = -1, \prod_{k=2}^d \xi_k = 1\right] + \frac{1}{4}\mathsf{E}\left[f\left(\bar{\boldsymbol{z}}(\xi)\right)\,\bigg|\,\xi_1 = -1, \prod_{k=2}^d \xi_k = -1\right] \\
&= \frac{1}{4}\mathsf{E}\left[f\left(\bar{\boldsymbol{z}}(\xi)\right)\,\bigg|\,\xi_1 = 1, \prod_{k=2}^d \xi_k = 1\right] - \frac{1}{4}\mathsf{E}\left[f\left(\bar{\boldsymbol{z}}(\xi)\right)\,\bigg|\,\xi_1 = 1, \prod_{k=2}^d \xi_k = -1\right] \\
&\quad - \frac{1}{4}\mathsf{E}\left[f\left(\bar{\boldsymbol{z}}(-\xi)\right)\,\bigg|\,\xi_1 = 1, \prod_{k=2}^d \xi_k = (-1)^{d-1}\right] + \frac{1}{4}\mathsf{E}\left[f\left(\bar{\boldsymbol{z}}(-\xi)\right)\,\bigg|\,\xi_1 = 1, \prod_{k=2}^d \xi_k = (-1)^d\right].
\end{aligned}
$$

By inserting and canceling a constant term, noticing $f\left(\bar{\boldsymbol{z}}(-\xi)\right) = f\left(-\bar{\boldsymbol{z}}(\xi)\right) = (-1)^d f\left(\bar{\boldsymbol{z}}(\xi)\right)$, the above expression further leads to

$$
\begin{aligned}
d!F\left((d+1)\bar{\boldsymbol{z}}^1, \bar{\boldsymbol{z}}^2, \cdots, \bar{\boldsymbol{z}}^d\right) &= \mathsf{E}\left[\prod_{k=1}^d \xi_k f\left(\bar{\boldsymbol{z}}(\xi)\right)\right] \\
&= \frac{1}{4}\mathsf{E}\left[\left(f\left(\bar{\boldsymbol{z}}(\xi)\right) - (2d)^d\, v_{\min}(P)\right)\,\bigg|\,\xi_1 = 1, \prod_{k=2}^d \xi_k = 1\right] \\
&\quad - \frac{1}{4}\mathsf{E}\left[\left(f\left(\bar{\boldsymbol{z}}(\xi)\right) - (2d)^d\, v_{\min}(P)\right)\,\bigg|\,\xi_1 = 1, \prod_{k=2}^d \xi_k = -1\right] \\
&\quad + \frac{(-1)^{d-1}}{4}\mathsf{E}\left[\left(f\left(\bar{\boldsymbol{z}}(\xi)\right) - (2d)^d\, v_{\min}(P)\right)\,\bigg|\,\xi_1 = 1, \prod_{k=2}^d \xi_k = (-1)^{d-1}\right] \\
&\quad + \frac{(-1)^d}{4}\mathsf{E}\left[\left(f\left(\bar{\boldsymbol{z}}(\xi)\right) - (2d)^d\, v_{\min}(P)\right)\,\bigg|\,\xi_1 = 1, \prod_{k=2}^d \xi_k = (-1)^d\right] \\
&\leq \frac{1}{2}\mathsf{E}\left[\left(f\left(\bar{\boldsymbol{z}}(\xi)\right) - (2d)^d\, v_{\min}(P)\right)\,\bigg|\,\xi_1 = 1, \prod_{k=2}^d \xi_k = 1\right],
\end{aligned}
$$

where the last inequality is due to (10). Therefore, we are able to find $\boldsymbol{\beta}'' = (\beta_1'', \beta_2'', \cdots, \beta_d'')^{\mathrm{T}} \in \mathbb{B}^d$ with $\beta_1'' = \prod_{k=2}^d \beta_k'' = 1$, such that

$$
\begin{aligned}
f(\bar{\boldsymbol{z}}(\beta'')) - (2d)^d\, v_{\min}(P) &\geq 2d!F((d+1)\bar{\boldsymbol{z}}^1, \bar{\boldsymbol{z}}^2, \cdots, \bar{\boldsymbol{z}}^d) \\
&\geq \frac{2\ln(1+\sqrt{2})}{1+e}\left(\frac{2}{\pi}\right)^{d-1}(d+1)!d^{-d}(n+1)^{-\frac{d-2}{2}}\, v_{\max}(P),
\end{aligned}
$$

where the last step is due to (9).

By Lemma A.2, a binary vector $\boldsymbol{x}' \in \mathbb{B}^n$ can be found in polynomial-time with $p(\boldsymbol{x}') \geq 0$. Since $\boldsymbol{z}(\beta'')/z_h(\beta'') \in [-1, 1]^n$, by Lemma A.1, a binary vector $\boldsymbol{x}'' \in \mathbb{B}^n$ can be found in polynomial-time with $p(\boldsymbol{x}'') \geq p(\boldsymbol{z}(\beta'')/z_h(\beta''))$. Below we shall prove at least one of $\boldsymbol{x}'$ and $\boldsymbol{x}''$ satisfies

$$p(\boldsymbol{x}) - v_{\min}(P) \geq \tau_P \left(v_{\max}(P) - v_{\min}(P) \right). \tag{11}$$

Indeed, if $-v_{\min}(P) \geq \tau_P \left(v_{\max}(P) - v_{\min}(P) \right)$, then $\boldsymbol{x}'$ satisfies (11) in this case. Otherwise we shall have $-v_{\min}(P) < \tau_P \left(v_{\max}(P) - v_{\min}(P) \right)$, then

$$v_{\max}(P) > (1 - \tau_P) \left(v_{\max}(P) - v_{\min}(P) \right) \geq \left(v_{\max}(P) - v_{\min}(P) \right)/2,$$

which implies

$$f\left(\frac{\bar{\boldsymbol{z}}(\beta'')}{2d} \right) - v_{\min}(P) \geq (2d)^{-d} \frac{2 \ln(1 + \sqrt{2})}{1 + e} \left(\frac{2}{\pi} \right)^{d-1} (d+1)! d^{-d} (n+1)^{-\frac{d-2}{2}} v_{\max}(P)$$
$$\geq \tau_P \left(v_{\max}(P) - v_{\min}(P) \right).$$

The above inequality also implies that $f\left(\bar{\boldsymbol{z}}(\beta'')/(2d) \right) > 0$. Recall that $\beta_1'' = 1$ implies $z_h(\beta'') > 0$. Therefore,

$$p(\boldsymbol{x}'') \geq p\left(\frac{\boldsymbol{z}(\beta'')}{z_h(\beta'')} \right) = f\left(\frac{\bar{\boldsymbol{z}}(\beta'')}{z_h(\beta'')} \right) = \left(\frac{2d}{z_h(\beta'')} \right)^d f\left(\frac{\bar{\boldsymbol{z}}(\beta'')}{2d} \right) \geq f\left(\frac{\bar{\boldsymbol{z}}(\beta'')}{2d} \right),$$

which implies $\boldsymbol{x}''$ satisfies (11). Finally, $\arg\max\{p(\boldsymbol{x}'), p(\boldsymbol{x}'')\}$ satisfies (11) in both cases. $\quad\square$