Approximation Algorithms for Discrete Polynomial Optimization

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

August 19, 2010

Abstract

In this paper, we consider approximation algorithms for optimizing a generic multi-variate 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 shall 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 shall 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.

Keywords: polynomial function optimization; 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, Kowloon Tong, Hong Kong. Email: simaihe@cityu.edu.hk

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

[‡]Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong. Email: zhang@se.cuhk.edu.hk. Research supported by Hong Kong RGC Grants CUHK419409 and CUHK419208.

1 Introduction

This paper is concerned with optimizing a generic (high degree) 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 Problem (P) : $\max_{\boldsymbol{x}\in\{1,-1\}^n} p(\boldsymbol{x})$. This type of problem can be found in a great variety of application areas. For example, the following hypergraph max-covering problem is well studied in the literature, which is precisely Problem (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 $x_i \longrightarrow (x_i + 1)/2$, the problem is transformed to Problem (P), and *vice versa*.

Note that Problem (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 the *Fourier support graph* problem. Mathematically, a polynomial function $p : \{-1, 1\}^n \to \Re$ has Fourier expansion $p(\boldsymbol{x}) = \sum_{S\subset\{1,2,\ldots,n\}} \hat{p}(S) \prod_{i\in S} x_i$, which is also called the 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 approximative solution in polynomial-time? The latter question actually motivates this paper. Indeed, Problem (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, Problem (P) is closely related to find 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 say S is an independent set of vertices if and only if $S \subset V$ 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 maximal over all possible independent sets.

In fact, any unconstrained binary polynomial maximization problem can be transformed into the maximum weighted independent set problem, and is also the most common technique for solving Problem (P) (see e.g. [6, 28]). 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

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

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

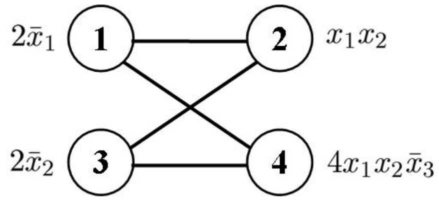

Figure 1: Conflict graph associated with $-2x_1 - 2x_2 + 5x_1x_2 - 4x_1x_2x_3$

their complements, i.e. $\bar{x}_i := 1 - x_i, i = 1, 2, 3$. 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 $f(\boldsymbol{x})$, 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, Problem (P) also has applications in *neural networks* [21, 8, 5], *error-correcting codes* [8, 27], etc. For instance, recently Khot and Naor [23] show that it has applications in the problem of *refutation of random k-CNF formulas* [10, 12, 13, 11].

One important subclass of polynomial function is homogeneous polynomials. Likewise, the homogeneous quadratic case of Problem (P) has been studied extensively; see e.g. [16, 25, 26, 3]. Homogeneous cubic polynomial is also studies by Khot and Naor [23]. 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. [3]), i.e.

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

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

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

Another 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

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

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 to maximize the largest singular value, i.e.

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

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 multi-linear 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:

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

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 three models described above motivate us in studying the mixed integer programming to be discussed in the next section.

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, Nestrov [25] derived a 0.636-approximation bound. 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 [3] derived a 0.56-approximation bound. Remark that all these approximation bounds remain hitherto the best. When the degree of the polynomial function is larger than 2, to our best knowledge, the only known approximation result in the literature is due to Khot and Naor [23], in which they present an $\Omega\left(\sqrt{\frac{\ln n}{n}}\right)$-approximation ratio for $\max_{\boldsymbol{x}\in\{1,-1\}^n} \sum_{1\le i,j,k\le n} a_{ijk}x_i x_j x_k$ with $(a_{ijk})_{n\times n\times n}$ being square-free ($a_{ijk}=0$ whenever two of the indices are equal). However, their approximation algorithm is mainly of theoretical interest, since the particular reduction relies on the ellipsoid method as a key construction.

In this paper we consider the optimization models for a general polynomial function of (any) 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 function optimization problems under quadratic constraints [19, 20]. However, the discrete models studied in this paper have novel features, and the analysis is therefore entirely different from our previous papers [19, 20]. 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 that we study in this paper are all multivariate polynomial functions. The following multi-linear tensor function plays a major role in our discussion:

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

where $\boldsymbol{x}^k \in \Re^{n_k}$ for all $k = 1, 2, \ldots, d$; and the letter 'T' signifies the notion of *tensor*. In the shorthand notation we shall denote $\boldsymbol{F} = (a_{i_1 i_2 \cdots i_d}) \in \Re^{n_1 \times n_2 \times \cdots \times n_d}$ to be a d-th order tensor, and F to be its corresponding multi-linear function. Closely related with the tensor form $\boldsymbol{F}$ is a general d-th degree homogeneous polynomial function $f(\boldsymbol{x})$, where $\boldsymbol{x} \in \Re^n$. We call the tensor form $\boldsymbol{F} = (a_{i_1 i_2 \cdots i_d})$ *super-symmetric* (see [24]) 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 form. 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 $\boldsymbol{F} = (b_{i_1 i_2 \cdots i_d}) \in \Re^{n^d}$, 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 multi-linear function defined by the super-symmetric tensor form $\boldsymbol{F}$, 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 $d_1 + d_2 + \cdots + d_s = d$, $\boldsymbol{x}^k \in \Re^{n_k}$ for all $k = 1, 2, \ldots, s$, and d-th order tensor form $\boldsymbol{F} \in \Re^{n_1^{d_1} \times n_2^{d_2} \times \cdots \times n_s^{d_s}}$; 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 form, and so on.

Beyond the homogeneous polynomial functions described above, a generic multi-variate 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 $\boldsymbol{x} \in \Re^n$, $f_0 \in \Re$, 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 all $k = 1, 2, \ldots, d$.

Throughout we shall adhere to the notation F for a multi-linear function 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 $\boldsymbol{F} \in \Re^{n_1 \times n_2 \times \cdots \times n_d}$, and $n_1 \le n_2 \le \cdots \le n_s$ in the tensor form $\boldsymbol{F} \in \Re^{n_1^{d_1} \times n_2^{d_2} \times \cdots \times n_s^{d_s}}$. 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 will be considered in this paper: discrete binary variables

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

and continuous variables on the unit sphere:

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

Note that in this paper we shall by default use the Euclidean norm for the vectors, the matrices, and the tensors. The decision variables in our models will 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:

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

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

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

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

and their *mixed* forms

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

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

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

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 Problems $(H)'$ and $(M)'$ to have partial symmetric property, and $m_1 \le m_2 \le \cdots \le m_{d'}$ in Problem $(T)'$, and $m_1 \le m_2 \le \cdots \le m_t$ in Problem $(M)'$.

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 (see e.g. [3]) as a subclass, i.e.

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

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 $\max_{\boldsymbol{x} \in S} f(\boldsymbol{x})$, we shall use $v_{\max}(P)$ to denote its optimal value, and $v_{\min}(P)$ to denote the optimal value of its minimization counterpart, i.e.

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

Definition 1 *We call the maximization problem* (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* $\boldsymbol{z} \in S$ *can be found in polynomial-time such that* $f(\boldsymbol{z}) \geq \tau\, v_{\max}(P)$*.*

Definition 2 *We call the maximization problem* (P) *to admit a polynomial-time approximation algorithm with* relative *approximation ratio* $\tau \in (0, 1]$*, if a feasible solution* $\boldsymbol{z} \in S$ *can be found in polynomial-time such that* $f(\boldsymbol{z}) - v_{\min}(P) \geq \tau\, (v_{\max}(P) - v_{\min}(P))$*.*

Regarding to the relative approximation ratios (Definition 2), in some cases it is convenient to use the equivalent form: $v_{\max}(P) - f(\boldsymbol{z}) \leq (1 - \tau)\,(v_{\max}(P) - v_{\min}(P))$.

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 '$\tau = \Omega(\rho)$' to signify the fact that $\tau\ (> 0)$ is at least the order of $\rho\ (> 0)$; that is, there is a positive constant c such that $\tau \geq c\rho$. 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 *Problem* (T) : $\max_{\boldsymbol{x}^k \in \mathbb{B}^{n_k}} F(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d)$ *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 shall remark that when $d = 2$, Problem (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 [3]. Huang and Zhang [22] considered similar problems for the complex variables and derived constant approximation ratios.

When $d = 3$, Problem (T) is a slight generalization of the model considered by Khot and Naor [23], 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 shall discard the assumptions on the symmetry and the square-free property altogether. Their approximation ratio is $\Omega\left(\sqrt{\frac{\ln n_1}{n_1}}\right)$. However, their approximation algorithm is not practically implementable.

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 [3], 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 combing the its first and second components, Problem (T) can be relaxed to

$$\begin{array}{ll} \max & F(\boldsymbol{X}, \boldsymbol{x}^3) \\ \text{s.t.} & \boldsymbol{X} \in \mathbb{B}^{n_1 n_2}, \boldsymbol{x}^3 \in \mathbb{B}^{n_3}. \end{array}$$

This problem is the exact form of Problem (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 $(\hat{\boldsymbol{X}}, \hat{\boldsymbol{x}}^3)$. To recover $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2)$ from $\hat{\boldsymbol{X}}$, we randomly generate

$$\begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{pmatrix} \sim \mathcal{N}\left(\boldsymbol{0}_{n_1+n_2}, \begin{bmatrix} \boldsymbol{I}_{n_1 \times n_1} & \hat{\boldsymbol{X}}/\sqrt{n_1} \\ \hat{\boldsymbol{X}}^{\mathrm{T}}/\sqrt{n_1} & \hat{\boldsymbol{X}}^{\mathrm{T}}\hat{\boldsymbol{X}}/n_1 \end{bmatrix}\right),$$

and let $\hat{\boldsymbol{x}}^1 := \operatorname{sign}(\boldsymbol{\xi})$ and $\hat{\boldsymbol{x}}^2 := \operatorname{sign}(\boldsymbol{\eta})$. It follows that

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

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

Theorem 3.2 *If $F(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{d})$ is square-free and d is odd, then Problem (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 Problem (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 multi-linear 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.

Lemma 3.4 (He, Li, and Zhang [19]) *Suppose $\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d \in \Re^n$, and $\xi_1, \xi_2, \cdots, \xi_d$ are i.i.d. random variables, each takes values 1 and -1 with equal probability. For any super-symmetric d-th order tensor form $\boldsymbol{F}$ and function $f(\boldsymbol{x}) = F(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{d})$, 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 Problem (H) hold under the square-free condition. This is because in this case the decision variables are actually in the multi-linear form. Hence, one can replace any point in the box ($\mathrm{Conv}(\mathbb{B}^n)$) by one of its vertices ($\mathbb{B}^n$) without decreasing its objective function value, due to the linearity.

We further move on to consider the mixed form of discrete polynomial optimization model, Problem (M). It is a generalization of Problems (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 to each* $\boldsymbol{x}^k$ $(k = 1, 2, \ldots, s)$*, and one of* d_k $(k = 1, 2, \ldots, s)$ *is odd, then Problem* $(M) : \max_{\boldsymbol{x}^k \in \mathbb{B}^{n_k}} f(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^s)$ *admits a polynomial-time approximation algorithm with approximation ratio* τ_M*, where*

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

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 to each* $\boldsymbol{x}^k$ $(k = 1, 2, \ldots, s)$*, and all* d_k $(k = 1, 2, \ldots, s)$ *are even, then Problem* $(M) : \max_{\boldsymbol{x}^k \in \mathbb{B}^{n_k}} f(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^s)$ *admits a polynomial-time approximation algorithm with relative approximation ratio* τ_M*.*

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

3.2 Homogeneous Polynomials in Mixed Variables

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

Proposition 3.7 serves the basis for Problem $(T)'$ of general d and d'. In this particular case, Problem $(T)'$ 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 [25].

Recursion is again the tool to deal with higher degrees. For recursion of d with discrete variables $\boldsymbol{x}^k$, same argument as of Theorem 3.1 is used. While for recursion of d' with continuous variables $\boldsymbol{y}^k$, we solve eigenvalue decomposition problems as stipulated in He, Li, and Zhang [19].

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

$$\tau_T' := (2/\pi)^{\frac{2d-1}{2}}(n_1n_2\cdots n_{d-1}m_1m_2\cdots m_{d'-1})^{-\frac{1}{2}} = \Omega\left((n_1n_2\cdots n_{d-1}m_1m_2\cdots m_{d'-1})^{-\frac{1}{2}}\right).$$

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

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 to* $\boldsymbol{x}$, *and either* d *or* d' *is odd, then Problem* $(H)' : \max_{\boldsymbol{x}\in\mathbb{B}^n,\boldsymbol{y}\in\mathbb{S}^m} f(\boldsymbol{x},\boldsymbol{y})$ *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 to* $\boldsymbol{x}$, *and both* d *and* d' *are even, then Problem* $(H)' : \max_{\boldsymbol{x}\in\mathbb{B}^n,\boldsymbol{y}\in\mathbb{S}^m} f(\boldsymbol{x},\boldsymbol{y})$ *admits a polynomial-time approximation algorithm with relative approximation ratio* τ_H'.

By relaxing Problem $(M)'$ to the multi-linear tensor function optimization Problem $(T)'$ 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 to each* $\boldsymbol{x}^k$ $(k=1,2,\dots,s)$, *and one of* d_k $(k=1,2,\dots,s)$ *or one of* d_ℓ' $(\ell=1,2,\dots,t)$ *is odd, then Problem* $(M)' : \max_{\boldsymbol{x}^k\in\mathbb{B}^{n_k},\boldsymbol{y}^\ell\in\mathbb{S}^{m_\ell}} f(\boldsymbol{x}^1,\boldsymbol{x}^2,\cdots,\boldsymbol{x}^s,\boldsymbol{y}^1,\boldsymbol{y}^2,\cdots,\boldsymbol{y}^t)$ *admits a polynomial-time approximation algorithm with approximation ratio* τ_M', *where*

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

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 to 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 Problem* $(M)'$: $\max_{\boldsymbol{x}^k\in\mathbb{B}^{n_k},\boldsymbol{y}^\ell\in\mathbb{S}^{m_\ell}} f(\boldsymbol{x}^1,\boldsymbol{x}^2,\cdots,\boldsymbol{x}^s,\boldsymbol{y}^1,\boldsymbol{y}^2,\cdots,\boldsymbol{y}^t)$ *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 break through 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 equal to 1, to yield a homogeneous form for Function P:

$$p(\boldsymbol{x})=\sum_{k=1}^{d}F_k(\underbrace{\boldsymbol{x},\boldsymbol{x},\cdots,\boldsymbol{x}}_{k})x_h^{d-k}+f_0x_h^d:=F\Bigg(\underbrace{\begin{pmatrix}\boldsymbol{x}\\x_h\end{pmatrix},\begin{pmatrix}\boldsymbol{x}\\x_h\end{pmatrix},\cdots,\begin{pmatrix}\boldsymbol{x}\\x_h\end{pmatrix}}_{d}\Bigg)=F(\underbrace{\bar{\boldsymbol{x}},\bar{\boldsymbol{x}},\cdots,\bar{\boldsymbol{x}}}_{d})=f(\bar{\boldsymbol{x}}),$$

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

Lemma 3.13 (He, Li, and Zhang [20]) *Suppose* $\bar{\boldsymbol{x}}^k=\binom{\boldsymbol{x}^k}{x_h^k}\in\Re^{n+1}$ *with* $|x_h^k|\le 1$ *for* $k=1,2,\ldots,d$*. Let* $\eta_1,\eta_2,\cdots,\eta_d$ *be independent random variables, each takes 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 takes 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 *Problem (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 Problem (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:

$$\begin{array}{lll} (D) & \max & p(\boldsymbol{x}) \\ & \text{s.t.} & x_i \in \{a^i_1, a^i_2, \cdots, a^i_{m_i}\},\ i = 1, 2, \ldots, n. \end{array}$$

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

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

the original decision variables can be equivalently transformed into

$$u_i = j \Longrightarrow x_i = a^i_j,\ i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, m_i,$$

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, Problem (D) is then reformulated as Problem (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 $\{a^i_1, a^i_2, \cdots, a^i_{m_i}\}$ $(i = 1, 2, \ldots, n)$ in Problem (D) are arithmetic sequences. Then it is much easier to transform Problem (D) to Problem (P), without going through the Lagrange interpolation technique. 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. Problems (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 shall 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 $\boldsymbol{A} = (a_{ij}) \in \Re^{n_1 \times n_2}$, denoted by $\|\boldsymbol{A}\|_C$, the maximum over all $I \subset \{1, 2, \dots, n_1\}$ and $J \subset \{1, 2, \dots, n_2\}$, 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, 4]). Alon and Naor in [3] proposed a randomized polynomial-time approximation algorithm that approximates the cut-norm with a factor at least 0.56, which is currently the best. Since a matrix is a second order tensor, it is natural to extend the cut-norm to general high order tensors, specifically, given a d-th order tensor $\boldsymbol{F} = (a_{i_1 i_2 \cdots i_d}) \in \Re^{n_1 \times n_2 \times \cdots \times n_d}$, its cut-norm is defined by

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

In fact, the cut-norm $\|\boldsymbol{F}\|_C$ is closely related to $\|\boldsymbol{F}\|_{\infty \mapsto 1}$, which is the exact form of Problem (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 $\boldsymbol{F} \in \Re^{n_1 \times n_2 \times \cdots \times n_d}$, $\|\boldsymbol{F}\|_C \le \|\boldsymbol{F}\|_{\infty \mapsto 1} \le 2^d \|\boldsymbol{F}\|_C$.*

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

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

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

Observer that $\|\boldsymbol{F}\|_C = \max_{\boldsymbol{z}^k\in\{0,1\}^{n_k},\,k=1,2,\ldots,d} |F(\boldsymbol{z}^1,\boldsymbol{z}^2,\cdots,\boldsymbol{z}^d)|$. For any $\boldsymbol{z}^k \in \{0,1\}^{n_k}$, $k = 1,2,\ldots,d$, 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

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

which implies $\|\boldsymbol{F}\|_C \le \|\boldsymbol{F}\|_{\infty\mapsto 1}$. □

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 $E \subset V\times V$ 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, $\boldsymbol{w}_e \in \Re^m_+$. 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

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

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

$$\begin{array}{ll} \max & \left\|\sum_{1\le i,j\le n} x_i x_j \boldsymbol{w}'_{ij}\right\| \\ \text{s.t.} & \boldsymbol{x}\in\mathbb{B}^n, \end{array}$$

where

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

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

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

This is the exact form of Problem $(H)'$ with $d=2$ and $d'=1$. Although the square-free property to $\boldsymbol{x}$ does not hold in this model (which is a condition of Theorem 3.9), one can still replace any point in the box ($\mathrm{Conv}(\mathbb{B}^n)$) by one of its vertices ($\mathbb{B}^n$) without decreasing its objective function value, since the matrix $F(\bullet,\bullet,\boldsymbol{e}_k) = \left((w'_{ij})_k\right)_{n\times n}$ 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. $\boldsymbol{W}_{ij} \in \Re^{m\times m}$, $\boldsymbol{W}_{ij} \succeq \boldsymbol{0}$), then the matrix-valued max-cut problem can also be formulated as

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

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

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

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 obvious 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 TRUE 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 Max-C-SAT. It is immediately clear that Max-C-SAT problem is NP-hard, since we can easily reduce the max-cut problem to 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 each clause containing no more than d literals. Suppose that clause k 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 $s_k + t_k \le d$, associated with a weight $w_k \ge 0$ for $k = 1,2,\ldots,m$. Then, the Max-C-SAT problem can be formulated in the form of Problem (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 funtion, 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. [2, 7]), and a "market split problem" [9], 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 $\boldsymbol{0} \le \boldsymbol{x} \le \boldsymbol{u}$, such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. The problem can be formulated by the least square method as

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

According to the discussion at the end of Section 3.3, the above problem can be reformulated as a form of Problem (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, Problem (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 Problem (L). Suppose the highest degree of the polynomial equations is d. Then, this least square problem can be reformulated as a form of Problem (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 $\Omega\left((\sum_{i=1}^n \ln u_i)^{-(d-1)} \right)$.

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] J.L.R. Alfonsín, *The Diophantine Frobenius Problem*, Oxford University Press, 2005.

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

[4] 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.

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

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

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

[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] 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.

[10] U. Feige, *Relations between Average Case Complexity and Approximation Complexity*, Proceedings of the 34th Annuals ACM Symposium on Theory of Computing, 534 – 543, 2002.

[11] U. Feige, J.H. Kim, and E. Ofek. *Witnesses for Non-Satisfiability of Dense Random 3CNF Formulas*, the 48th Annual IEEE Symposium on Foundations of Computer Science, 497 –508, 2006.

[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 Jounal 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*, Technical Report SEEM2009-05, Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, 2009.

[20] S. He, Z. Li, and S. Zhang, *General Constrained Polynomial Optimization: an Approximation Approach*, Technical Report SEEM2009-06, Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, 2009.

[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*, to appear in Science in China, Series A.

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

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

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

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

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

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

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 [3]. For general $d \geq 3$, let $\boldsymbol{X} = \boldsymbol{x}^1 (\boldsymbol{x}^d)^{\mathrm{T}}$ and Problem (T) is then relaxed to

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

where we treat $\boldsymbol{X}$ as an $n_1 n_d$-dimensional vector, and $\boldsymbol{F} \in \Re^{n_1 n_d \times n_2 \times \cdots \times n_{d-1}}$ as a $(d-1)$-order tensor. Observe that Problem $(\hat{T})$ is the exact form of Problem (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

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

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

$$
\tilde{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{I}_{n_1 \times n_1} & \hat{\boldsymbol{X}}/\sqrt{n_1} \\ \hat{\boldsymbol{X}}^{\mathrm{T}}/\sqrt{n_1} & \hat{\boldsymbol{X}}^{\mathrm{T}} \hat{\boldsymbol{X}}/n_1 \end{bmatrix} \succeq 0, \tag{2}
$$

and randomly generate

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

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, by Goemans and Williamson [16] it follows 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

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

Thus $\hat{\boldsymbol{x}}^1$ and $\hat{\boldsymbol{x}}^d$ can be found by the randomization process, which concludes the induction step. □

Lemma A.1 *Suppose a polynomial function $p(\boldsymbol{x})$ is square-free, and $\boldsymbol{z} \in \mathrm{Conv}(\mathbb{B}^n)$. Then $\boldsymbol{x}' \in \mathbb{B}^n$ and $\boldsymbol{x}'' \in \mathbb{B}^n$ can be found in polynomial-time, such that $p(\boldsymbol{x}') \le p(\boldsymbol{z}) \le p(\boldsymbol{x}'')$.*

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

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

Let

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

Then

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

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 $p(\boldsymbol{x}') \le p(\boldsymbol{z})$. Using a similar argument, we may find $\boldsymbol{x}'' \in \mathbb{B}^n$ with $p(\boldsymbol{x}'') \ge p(\boldsymbol{z})$. □

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. Problem (H) can be relaxed to

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

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 taken 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 $\frac{1}{d}\sum_{k=1}^{d}\left(\prod_{i\neq k}\beta_i\right)\hat{\boldsymbol{x}}^k \in \mathrm{Conv}(\mathbb{B}^n)$, because for all $1 \leq j \leq n$,

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

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

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

□

Lemma A.2 *Suppose in Problem (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 $k\,(1 \le k \le d)$ 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 $v_{\min}(P) \le 0 \le v_{\max}(P)$. By a randomization process, a binary vector $\boldsymbol{x}' \in \mathbb{B}^n$ can be found in polynomial-time with $p(\boldsymbol{x}') \ge 0$. □

A.3 Proof of Theorem 3.3

Proof. First we assume that $v_{\max}(H) \ge -v_{\min}(H)$. Like in the proof of Theorem 3.2, by relaxing Problem (H) to Problem (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

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

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

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

Applying Lemma 3.4 and we have

$$\begin{aligned}
& \frac{1}{2}\mathsf{E}\left[f\left(\frac{1}{d}\sum_{k=1}^d \xi_k \hat{\boldsymbol{x}}^k\right) - v_{\min}(H) \,\middle|\, \prod_{i=1}^d \xi_i = 1 \right] \\
\ge\ & \frac{1}{2}\mathsf{E}\left[f\left(\frac{1}{d}\sum_{k=1}^d \xi_k \hat{\boldsymbol{x}}^k\right) - v_{\min}(H) \,\middle|\, \prod_{i=1}^d \xi_i = 1 \right] - \frac{1}{2}\mathsf{E}\left[f\left(\frac{1}{d}\sum_{k=1}^d \xi_k \hat{\boldsymbol{x}}^k\right) - v_{\min}(H) \,\middle|\, \prod_{i=1}^d \xi_i = -1 \right] \\
=\ & \mathsf{E}\left[\prod_{i=1}^d \xi_i \left(f\left(\frac{1}{d}\sum_{k=1}^d \xi_k \hat{\boldsymbol{x}}^k\right) - v_{\min}(H)\right)\right] \\
=\ & d^{-d}\mathsf{E}\left[\prod_{i=1}^d \xi_i f\left(\sum_{k=1}^d \xi_k \hat{\boldsymbol{x}}^k\right)\right] - v_{\min}(H)\,\mathsf{E}\left[\prod_{i=1}^d \xi_i\right] \\
=\ & d^{-d} d!\, F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \ldots, \hat{\boldsymbol{x}}^d) \ge \tau_H\, v_{\max}(H) \ge (\tau_H/2)\,(v_{\max}(H) - v_{\min}(H)),
\end{aligned}$$

where the last inequality is due to $v_{\max}(H) \ge -v_{\min}(H)$. 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

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

Noticing $\frac{1}{d}\sum_{k=1}^{d}\beta_k\hat{\boldsymbol{x}}^k \in \mathrm{Conv}(\mathbb{B}^n)$ and applying Lemma A.1, we are able to find $\boldsymbol{x}'' \in \mathbb{B}^n$ with

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

Recall all the derivations above are based on the assumption of $v_{\max}(H) \geq -v_{\min}(H)$. In case this is not true, it implies $-v_{\min}(H) \geq \left(v_{\max}(H) - v_{\min}(H)\right)/2$. By Lemma A.2, we are able to find $\boldsymbol{x}' \in \mathbb{B}^n$ in polynomial time with $f(\boldsymbol{x}') \geq 0$. Thus

$$f(\boldsymbol{x}') - v_{\min}(H) \geq -v_{\min}(H) \geq \left(v_{\max}(H) - v_{\min}(H)\right)/2 \geq \tau_H\left(v_{\max}(H) - v_{\min}(H)\right).$$

By picking $\tilde{\boldsymbol{x}} = \mathrm{argmax}\{f(\boldsymbol{x}'), f(\boldsymbol{x}'')\}$, the desired result is proved. □

A.4 Proof of Theorem 3.5

Proof. Like in the proof of Theorem 3.2, by relaxing Problem (M) to Problem (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

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

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

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

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

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

It is easy to verify that $\prod_{i=1}^{d}\beta_i\hat{\boldsymbol{x}}_\beta^1/d_1 \in \mathrm{Conv}(\mathbb{B}^n)$, and $\hat{\boldsymbol{x}}_\beta^k/d_k \in \mathrm{Conv}(\mathbb{B}^n)$ 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

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

□

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

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

instead of invoking Lemma 3.4 directly, where $\hat{\boldsymbol{x}}^k_\xi\,(1 \le k \le s)$ is defined by (5); and (ii) we use $f\left(\frac{1}{d_1}\hat{\boldsymbol{x}}^1_\xi, \frac{1}{d_2}\hat{\boldsymbol{x}}^2_\xi, \cdots, \frac{1}{d_s}\hat{\boldsymbol{x}}^s_\xi\right)$ 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 $d = d' = 1$, Problem $(T)'$ can be written as

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

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 [25], it admits an approximation ratio of $2/\pi$. Thus the original problem admits a polynomial-time approximation algorithm with approximation ratio $\sqrt{2/\pi}$. □

Proposition A.3 (He, Li, and Zhang [19]) *Problem* (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* $v_{\max}(S) \ge \|\boldsymbol{Q}\|/\sqrt{m_1}$.

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

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

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 $\boldsymbol{Q} = F(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^d, \cdot, \hat{\boldsymbol{y}}^2, \hat{\boldsymbol{y}}^3, \cdots, \hat{\boldsymbol{y}}^{d'-1}, \cdot) \in \Re^{m_1 \times m_{d'}}$. Then by Proposition A.3, 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 $\boldsymbol{X} = \boldsymbol{x}^1 (\boldsymbol{x}^d)^{\mathrm{T}}$. Problem $(T)'$ can be relaxed to the other case with degree $d - 1 + d'$, i.e.

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

By induction, it 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 use the same randomization procedure as (2) in the proof of Theorem 3.1, which deteriorates an additional factor of $\frac{2}{\pi \sqrt{n_1}}$ as (3). Combining these two factors, we are led to a final ratio of τ'_T. $\square$

A.8 Proof of Theorem 3.9

Proof. Like in the proof of Theorem 3.2, by relaxing Problem $(H)'$ to Problem $(T)'$, 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 taken 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 $\hat{\boldsymbol{x}}/d \in \mathrm{Conv}(\mathbb{B}^n)$ and $f(\boldsymbol{x}, \boldsymbol{y})$ is square-free to $\boldsymbol{x}$, by applying Lemma A.1, $\tilde{\boldsymbol{x}} \in \mathbb{B}^n$ can be found in polynomial-time, with

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

□

A.9 Proof of Theorem 3.10

Proof. Following the exact argument 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

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

Since $f(\boldsymbol{x},\boldsymbol{y}')$ is square-free to $\boldsymbol{x}$ 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 $f(\boldsymbol{x}',\boldsymbol{y}')\ge 0\ge v_{\min}(\hat{H})\ge v_{\min}(H')$.

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

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

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

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

resulting in a contradiction.

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

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

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

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

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

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

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

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

which implies

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

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

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

which implies $(\boldsymbol{x}'', \boldsymbol{y}'')$ satisfies (7). Finally, $\operatorname{argmax}\{f(\boldsymbol{x}', \boldsymbol{y}'), f(\boldsymbol{x}'', \boldsymbol{y}'')\}$ satisfies (7) in both cases. □

A.10 Proof of Theorem 3.11

Proof. The proof is analogous to the proof of Theorem 3.9. We first relax Problem $(M)'$ to Problem $(T)'$ 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. □

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 $f\left(\hat{\boldsymbol{x}}_\xi, \hat{\boldsymbol{y}}_\eta\right)$. □

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 all $i = 1, 2, \ldots, n$, 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

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

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 $p(\boldsymbol{x})$.

Problem (P) is then equivalent to

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

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

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

Let $(\bar{\boldsymbol{u}}^1, \bar{\boldsymbol{u}}^2, \cdots, \bar{\boldsymbol{u}}^d)$ be the feasible solution of Problem $(\bar{T})$ 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

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

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 $1 \le k \le d$, and $(\xi_1, \xi_2, \cdots, \xi_d) = \boldsymbol{\xi}^{\mathrm{T}}$ are i.i.d. random variables, taking values 1 and -1 with equal probability. By combining these two identities, we have, for any constant c, the following identity

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

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

$$F\left(\binom{\beta_1' \boldsymbol{v}^1}{1}, \binom{\beta_2' \boldsymbol{v}^2}{1}, \cdots, \binom{\beta_d' \boldsymbol{v}^d}{1}\right) \ge \tau_0\, F(\bar{\boldsymbol{v}}^1, \bar{\boldsymbol{v}}^2, \cdots, \bar{\boldsymbol{v}}^d),$$

with

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

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

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

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

For any $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_d)^{\mathrm{T}} \in \mathbb{B}^d$, 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 $1 \le k \le d$ and $1 \le i \le n$, it follows that

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

Thus $\boldsymbol{z}(\beta)/z_h(\beta) \in \mathrm{Conv}(\mathbb{B}^n)$. 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

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

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

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

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

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

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

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

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

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

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

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

Indeed, if $-v_{\min}(P) \ge \tau_P\left(v_{\max}(P) - v_{\min}(P)\right)$, 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

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

which implies

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

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, $\operatorname{argmax}\{p(\boldsymbol{x}'), p(\boldsymbol{x}'')\}$ satisfies (11) in both cases. $\square$