ISSN: 2194-668X (PRINT VERSION); ISSN: 2194-6698 (ELECTRONIC VERSION)

Journal of the Operations Research Society of China

VOLUME 1 • ISSUE 1 • MARCH 2013

Springer

JORC (2013) 1:1–2
DOI 10.1007/s40305-013-0011-1

PREFACE

Journal of the Operations Research Society of China

Ya-xiang Yuan

Published online: 22 March 2013

I am very pleased to present the first issue of *Journal of the Operations Research Society of China*. I believe that the launch of this new journal is not only a festival for the Operations Research Society of China, but also an important event in the international operational research community.

Economical globalization has dramatically changed the economical systems. Problems and difficulties crop up in optimization and decision making when we try to do dynamic reformations in energy, transportation, telecommunication, financial engineering, urban planning, health care, environmental pollution, natural resource consumption and transnational logistics. In solution of the complicated problems, traditional theories and methodologies of operations research and management science prove to be less useful. We have to find way out. We need to better understand the backgrounds and nature of the intrinsic system. We endeavor to develop new theories, methodologies and modeling approaches.

I hope, this new journal, launched by the Operations Research Society of China, will timely introduce problems in practical optimization and decision making, and report the advances in their solutions. I wish that *Journal of the Operations Research Society of China* be a global forum for the entire community of operations research and management science.

Y.-x. Yuan (✉)
State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, AMSS, Chinese Academy of Sciences, Zhong Guan Cun Donglu 55, Beijing 100190, P.R. China
e-mail: yyx@lsec.cc.ac.cn

I am happy and proud that we have a qualified international editorial board, that is sure to accomplish the mission of this journal. Of course, our goal cannot be achieved without the support from all optimizers and practitioners. The success of this journal depends on your care, support and contributions.

Editor-in-Chief

JORC (2013) 1:3–36
DOI 10.1007/s40305-013-0003-1

REGULAR PAPER

Approximation Algorithms for Discrete Polynomial Optimization

Simai He · Zhening Li · Shuzhong Zhang

Received: 19 December 2012 / Accepted: 22 December 2012 / Published online: 20 February 2013

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

Keywords Polynomial optimization problem · Binary integer programming · Mixed integer programming · Approximation algorithm · Approximation ratio

Simai He was supported in part by Hong Kong General Research Fund (No. CityU143711).
Zhening Li was supported in part by Natural Science Foundation of Shanghai (No. 12ZR1410100) and Ph.D. Programs Foundation of Chinese Ministry of Education (No. 20123108120002).
Shuzhong Zhang was supported in part by U.S. National Science Foundation (No. CMMI-1161242).

S. He
Department of Management Sciences, City University of Hong Kong, Kowloon, Hong Kong
e-mail: simaihe@cityu.edu.hk

Z. Li (✉)
Department of Mathematics, Shanghai University, Shanghai 200444, China
e-mail: zheningli@gmail.com

S. Zhang
Department of Industrial & Systems Engineering, University of Minnesota, Minneapolis, MN 55455, USA
e-mail: zhangs@umn.edu

Mathematics Subject Classification 90C10 · 90C11 · 90C59 · 15A69 · 90C26

1 Introduction

This paper is concerned with optimizing a (high degree) multivariate polynomial function in (mixed) binary variables. Our basic model is to maximize a d-th degree polynomial function $p(\boldsymbol{x})$ where $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ is chosen such that $x_i \in \{1, -1\}$ for $i = 1, 2, \cdots, n$. For ease of referencing, let us call this basic model to be $(P) : \max_{\boldsymbol{x} \in \{1,-1\}^n} p(\boldsymbol{x})$. This type of problem can be found in a great variety of application domains. For example, the following hypergraph max-covering problem is well studied in the literature, which is precisely (P). Given a hypergraph $H = (V, E)$ with V being the set of vertices and E the set of hyperedges (or subsets of V), and each hyperedge $e \in E$ is associated with a real-valued weight $w(e)$. The problem is to find a subset S of the vertices set V, such that the total weight of the hyperedges covered by S is maximized. Denoting $x_i \in \{0, 1\}$ $(i = 1, 2, \cdots, n)$ 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 $x_i \to (x_i + 1)/2$, the problem is transformed to (P), and vice versa.

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

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

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

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

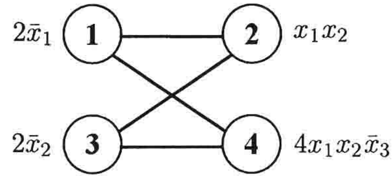

Fig. 1 Conflict graph associated with $-2x_1 - 2x_2 + 5x_1x_2 - 4x_1x_2x_3$

example, such polynomial can be

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

The conflict graph $G(f)$ associated with a polynomial $f(\boldsymbol{x})$ has vertices corresponding to the terms of $f(\boldsymbol{x})$, and each vertex is associated with a term in the polynomial except for the constant term. Two vertices in $G(f)$ are connected by an edge if and only if one of the corresponding terms contains a variable and the other corresponding term contains its complement variable. The weight of a vertex in $G(f)$ is the coefficient of the corresponding term in f. The conflict graph of $f(\boldsymbol{x})$ is shown in Fig. 1. Maximizing the weighted independent set of the conflict graph also solves the binary polynomial optimization problem. Beyond its connection to the graph problems, (P) also has applications in *neural networks* [4, 8, 21], *error-correcting codes* [8, 29], etc. For instance, recently Khot and Naor [24] show that it has applications in the problem of *refutation of random k-CNF formulas* [12, 13].

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

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

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

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

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

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

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

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

These two problems are actually equivalent to

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

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

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

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

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

This is indeed equivalent to

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

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

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

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

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

2 Notations and Model Descriptions

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

2.1 Objective Functions

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

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

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

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

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

We shall also consider in this paper the following:

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

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

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

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

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

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

2.2 Decision Variables

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

$$x \in \mathbb{B}^n := \{z \in \mathbb{R}^n \mid {z_i}^2 = 1, i = 1, 2, \cdots, n\},$$

and continuous variables on the unit sphere:

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

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

2.3 Model Descriptions

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

$$
\begin{array}{lll}
(T) & \max & F\left(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^d\right) \\
 & \text{s.t.} & \boldsymbol{x}^k \in \mathbb{B}^{n_k}, \quad k = 1, 2, \cdots, d; \\
(H) & \max & f(\boldsymbol{x}) = F(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{d}) \\
 & \text{s.t.} & \boldsymbol{x} \in \mathbb{B}^n; \\
(M) & \max & f\left(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^s\right) = F\big(\underbrace{\boldsymbol{x}^1, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^1}_{d_1}, \underbrace{\boldsymbol{x}^2, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^2}_{d_2}, \cdots, \\
 & & \qquad \underbrace{\boldsymbol{x}^s, \boldsymbol{x}^s, \cdots, \boldsymbol{x}^s}_{d_s}\big) \\
 & \text{s.t.} & \boldsymbol{x}^k \in \mathbb{B}^{n_k}, \quad k = 1, 2, \cdots, s; \\
(P) & \max & p(\boldsymbol{x}) = \sum_{k=1}^{d} F_k(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}}_{k}) + f_0 \\
 & \text{s.t.} & \boldsymbol{x} \in \mathbb{B}^n;
\end{array}
$$

and their *mixed* models:

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

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

2.4 Approximation Ratios

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

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

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

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

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

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

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

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. To simplify, we use the notion $\Omega(f(n))$ to signify that there are positive universal constants α and n_0 such that $\Omega(f(n)) \geqslant \alpha f(n)$ for all $n \geqslant n_0$. Throughout our discussion, we shall fix the degree of the objective polynomial function (denoted by d or $d + d'$ in the paper) to be a constant.

3.1 Homogeneous Polynomials in Binary Variables

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

$$\tau_T := (n_1n_2\cdots n_{d-2})^{-\frac{1}{2}}(2/\pi)^{d-1}\ln(1+\sqrt{2}) = \Omega\big((n_1n_2\cdots n_{d-2})^{-\frac{1}{2}}\big).$$

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

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

Our approximation algorithm works for general degree d based on recursion, and is fairly simple. We may take any approximation algorithm for the $d=2$ case, say the algorithm by Alon and Naor [2], as a basis. When $d=3$, noticing that any $n_1\times n_2\times n_3$ third order tensor can be written as an $(n_1n_2)\times n_3$ matrix by combining its first and second modes, (T) can be *relaxed* to

$$\begin{array}{ll}\max & F(X,x^3) := \displaystyle\sum_{1\leqslant i\leqslant n_1,1\leqslant j\leqslant n_2,1\leqslant k\leqslant n_3} a_{ijk}X_{ij}x^3_k\\ \text{s.t.} & X\in\mathbb{B}^{n_1n_2},\quad x^3\in\mathbb{B}^{n_3}.\end{array}$$

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

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

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

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

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

DR 3.1 (Decomposition Routine)

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

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

- Randomly generate

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

  and compute

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

  repeat if necessary, until $(\hat{\boldsymbol{x}}^1)^{\mathrm{T}}\boldsymbol{M}\hat{\boldsymbol{x}}^2 \geqslant \frac{2}{\pi\sqrt{n_1}}\boldsymbol{M} \bullet \hat{\boldsymbol{X}}$.
- Output: binary vectors $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2)$.

---

where

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

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

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

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

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

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

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

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

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

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

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

3.2 Homogeneous Polynomials in Mixed Variables

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

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

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

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

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

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

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

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

Theorem 3.11 *If*

$$F\big(\underbrace{x^1, x^1, \cdots, x^1}_{d_1}, \cdots, \underbrace{x^s, x^s, \cdots, x^s}_{d_s}, \underbrace{y^1, y^1, \cdots, y^1}_{d_1'}, \cdots, \underbrace{y^t, y^t, \cdots, y^t}_{d_t'}\big)$$

is square-free in each x^k $(k = 1, 2, \cdots, s)$, and one of d_k $(k = 1, 2, \cdots, s)$ or one of d_ℓ' $(\ell = 1, 2, \cdots, t)$ is odd, then $(M)'$: $\max_{x^k \in \mathbb{B}^{n_k}, y^\ell \in \mathbb{S}^{m_\ell}} f(x^1, x^2, \cdots, x^s, y^1, y^2, \cdots, y^t)$ admits a polynomial-time approximation algorithm with approximation ratio τ_M', where

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

Theorem 3.12 *If*

$$F(\underbrace{x^1, x^1, \cdots, x^1}_{d_1}, \cdots, \underbrace{x^s, x^s, \cdots, x^s}_{d_s}, \underbrace{y^1, y^1, \cdots, y^1}_{d'_1}, \cdots, \underbrace{y^t, y^t, \cdots, y^t}_{d'_t})$$

is square-free in each x^k $(k = 1, 2, \cdots, s)$, *and all* d_k $(k = 1, 2, \cdots, s)$ *and all* d'_ℓ $(\ell = 1, 2, \cdots, t)$ *are even, then* $(M)'$: $\max_{x^k \in \mathbb{B}^{n_k}, y^\ell \in \mathbb{S}^{m_\ell}} f(x^1, x^2, \cdots, x^s, y^1, y^2, \cdots, y^t)$ *admits a polynomial-time approximation algorithm with relative approximation ratio* τ'_M.

3.3 Inhomogeneous Polynomials in Binary Variables

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

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

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

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

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

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

and

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

Our last result is the following theorem.

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

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

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

$$\begin{array}{lll} (D) & \max & p(\boldsymbol{x}) \\ & \text{s.t.} & x_i \in \left\{a_1^i, a_2^i, \cdots, a_{m_i}^i\right\}, \quad i = 1, 2, \cdots, n. \end{array}$$

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

$$x_i = \sum_{j=1}^{m_i} a_j^i \prod_{1\leqslant k\leqslant m_i,\, k\neq j} \frac{u_i - k}{j-k}, \quad i = 1, 2, \cdots, n,$$

the original decision variables can be equivalently transformed into

$$u_i = j \quad \Longrightarrow \quad x_i = a_j^i, \quad i = 1, 2, \cdots, n, \; j = 1, 2, \cdots, m_i,$$

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

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

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

4 Examples of Application

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