A CORRELATIVELY SPARSE LAGRANGE MULTIPLIER EXPRESSION RELAXATION FOR POLYNOMIAL OPTIMIZATION

ZHENG QU* AND XINDONG TANG†

Abstract. In this paper, we consider polynomial optimization with correlative sparsity. We construct correlatively sparse Lagrange multiplier expressions (CS-LMEs) and propose CS-LME reformulations for polynomial optimization problems using the Karush-Kuhn-Tucker optimality conditions. Correlatively sparse sum-of-squares (CS-SOS) relaxations are applied to solve the CS-LME reformulation. We show that the CS-LME reformulation inherits the original correlative sparsity pattern, and the CS-SOS relaxation provides sharper lower bounds when applied to the CS-LME reformulation, compared with when it is applied to the original problem. Moreover, the convergence of our approach is guaranteed under mild conditions. In numerical experiments, our new approach usually finds the global optimal value (up to a negligible error) with a low relaxation order, for cases where directly solving the problem fails to get an accurate approximation. Also, by properly exploiting the correlative sparsity, our CS-LME approach requires less computational time than the original LME approach to reach the same accuracy level.

Key words. polynomial optimization, correlative sparsity, Lagrange multiplier expressions, Moment-SOS relaxations

MSC codes. 90C23, 90C06, 90C22

1. Introduction.

Let n be a positive integer, and let $x := (x_1, \dots, x_n)$ be the variable in the n-dimensional Euclidean space. Denote by $\mathbb{R}[x]$ be the ring of real coefficient polynomials in n indeterminates. We consider the polynomial optimization problem

$$(1.1)\qquad \begin{cases} \min\limits_{x\in\mathbb{R}^n} & f(x) \\ \text{s.t.} & g(x) \geq 0, \quad h(x) = 0. \end{cases}$$

In the above, $f \in \mathbb{R}[x]$ is a polynomial, and $g \in \mathbb{R}[x]^m$ and $h \in \mathbb{R}[x]^\ell$ are tuples of polynomial functions. In [10], Lasserre introduced a hierarchy of semidefinite programming (SDP) relaxations to provide a sequence of lower bounds for (1.1), which converges to the global optimal value of (1.1), under some compactness assumptions. This approach is known as *the Moment-SOS relaxations* and has been intensively explored in the last two decades for global solutions of polynomial optimization problems. For (1.1), Nie introduced the *Lagrange multiplier expressions* (LMEs) [23], whose existence is guaranteed when $g(x)$ and $h(x)$ are given by generic polynomial functions. LMEs can be applied to construct the *LME reformulation* of (1.1) using the Karush-Kuhn-Tucker (KKT) optimality conditions, which guarantees the moment relaxation being exact when the relaxation order is big enough and the global minimum for (1.1) is attainable. However, these approaches are usually computationally expensive. Indeed, even for unconstrained polynomial optimization problems, i.e., $m = \ell = 0$, the moment relaxation for (1.1) is an SDP problem with matrices of size up to $\binom{n+d}{n} \times \binom{n+d}{n}$, where $d \in \mathbb{N}$ is the relaxation order such that $2d \geq \deg(f)$.

Given the polynomial optimization problem (1.1), let $(\mathcal{I}_1, \dots, \mathcal{I}_s)$ be subsets of $[n] := \{1, \dots, n\}$ such that $\bigcup_{i=1}^s \mathcal{I}_i = [n]$, and denote $x^{(i)} := (x_j)_{j\in\mathcal{I}_i}$. The equation (1.1) is said to follow the *correlative sparsity pattern* (csp) $(\mathcal{I}_1, \dots, \mathcal{I}_s)$ if

*Department of Mathematics, the University of Hong Kong, Pokfulam Road, Hong Kong. (email: zhengqu@hku.hk)

†Department of Applied Mathematics, the Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. (email: xindong.tang@polyu.edu.hk)

(1) there exist $f_1, f_2, \ldots, f_s$ such that every $f_i \in \mathbb{R}[x^{(i)}]$ and $f(x) = f_1(x^{(1)}) + \cdots + f_s(x^{(s)})$;

(2) there exist partitions $I = (I_1, \ldots, I_s)$ of $[m]$ and $E = (E_1, \ldots, E_s)$ of $[\ell]$, such that for all $i \in [s]$, we have $g_{j_1} \in \mathbb{R}[x^{(i)}]$ and $h_{j_2} \in \mathbb{R}[x^{(i)}]$ for every $j_1 \in I_i$ and $j_2 \in E_i$.

For convenience, we let $m_i := |I_i|$ and $\ell_i := |E_i|$, and denote

$$g^{(i)} := (g_j : j \in I_i), \quad h^{(i)} := (h_j : j \in E_i).$$

Then, both $g^{(i)}$ and $h^{(i)}$ are subsets of $\mathbb{R}[x^{(i)}]$, and the polynomial optimization (1.1) with csp $(\mathcal{I}_1, \ldots, \mathcal{I}_s)$ can be written in the following way:

$$(1.2) \qquad \left\{ \begin{array}{cl} \min\limits_{x \in \mathbb{R}^n} & f_1(x^{(1)}) + f_2(x^{(2)}) + \cdots + f_s(x^{(s)}) \\ \text{s.t.} & g^{(1)}(x^{(1)}) \geq 0, \ldots, g^{(s)}(x^{(s)}) \geq 0, \\ & h^{(1)}(x^{(1)}) = 0, \ldots, h^{(s)}(x^{(s)}) = 0. \end{array} \right.$$

In this paper, we are interested in problems with csp $\{\mathcal{I}_1, \ldots, \mathcal{I}_s\}$ that satisfies the *running intersection property* (RIP), meaning that for each $1 \leq i \leq s-1$, $\mathcal{I}_{i+1} \cap (\mathcal{I}_1 \cup \cdots \cup \mathcal{I}_i) \subset \mathcal{I}_t$ for some $t \in \{1, \ldots, i\}$; see Definition 2.2. The Moment-SOS relaxation with correlative sparsity is studied in [34], and the convergence results are proved in [5, 9, 11, 25] for the case when the RIP holds. Recently, Wang *et al.* developed the software `TSSOS` [15] that implements correlative and term sparse SOS relaxations for polynomial optimization (see also [16, 36, 37]), and it has been used in many applications [17, 35].

Note that for any polynomial optimization problem, the trivial csp, i.e., $s = 1$ with $\mathcal{I}_1 = [n]$, always exists. Our primary interests lie in the cases where n is much bigger than $\max_{i \in [s]} |\mathcal{I}_i|$. For polynomial optimization (1.1) with the given csp, we aim to construct reformulations similar to Nie's LME reformulation introduced in [23], while maintaining the correlative sparsity of (1.1). Our main contributions are:

- For polynomial optimization with the given csp, we provide a systematic way to construct correlatively sparse LMEs (CS-LMEs), which are polynomial functions in x and some auxiliary variables.
- Based on CS-LMEs, we proposed correlatively sparse reformulations using the KKT optimality conditions. We show that under some general conditions, the reformulation inherits the csp and the running intersection property (RIP) from the original polynomial optimization, and their optimal values are identical.
- We show that for a given relaxation order, correlatively sparse SOS (CS-SOS) relaxations always provide tighter lower bounds for the optimal value of the polynomial optimization problem when the CS-LME reformulation is applied. The asymptotic convergence of our approach is proved under some standard assumptions. Numerical experiments are given to show the superiority of our CS-LME approach.

This paper is organized as follows. Some preliminaries for polynomial optimization and Lagrange multiplier expressions are given in Section 2. In Section 3, CS-LMEs are studied, and reformulations based on CS-LMEs are proposed. Section 4 studies the CS-SOS relaxations for solving CS-LME relaxations. Numerical experiments are presented in Section 5, and we conclude our approach and discuss future work in Section 6. In Appendix A, we briefly recall the general methodology for the computation of LMEs and CS-LMEs.

2. Preliminaries.

2.1. Notation and definitions. Let r be a positive integer. Denote $[r] := \{1, \dots, r\}$ and let $\mathbf{I}_r$ be the r-by-r identity matrix. When the dimension is clear, we use $\mathbf{0}$ (resp., $\mathbf{1}$) to denote the all-zero (resp., all-one) vector. Given two vectors $v, w \in \mathbb{R}^r$, we denote by $v \circ w$ the entry-wise product of v and w, and $v \perp w$ means that $v^\top w = 0$. For $v \in \mathbb{R}^r$ and $1 \le i \le j \le r$, we denote by $v_{i:j}$ the subvector formed by the elements of v indexed from i to j, i.e., $v_{i:j} := [v_i, \cdots, v_j]^\top$.

Let $z = (z_1, \dots, z_r)$ be a tuple of variables. Denote by $\mathbb{R}[z]$ the ring of polynomials in variables $z_1, \dots, z_r$ with real coefficients, and let $\mathbb{R}[z]^{r \times k}$ (resp., $\mathbb{R}[z]^r$) be the set of all $r \times k$ matrices (resp., r-dimensional vectors) whose entries are polynomials in z. For a polynomial $p \in \mathbb{R}[z]$, denote by $\deg(p)$ the degree of p. For an integer $d \in \mathbb{N}$, let $\mathbb{R}[z]_d$ be the $\mathbb{R}$-vector space of real polynomials in r variables of degrees at most d. A polynomial $p \in \mathbb{R}[z]$ is a sum-of-squares (SOS) if there exist $\sigma_1, \dots, \sigma_t \in \mathbb{R}[z]$ such that $p = (\sigma_1)^2 + \cdots + (\sigma_t)^2$. Denote by $\Sigma[z]$ the set of SOS polynomials in z and let $\Sigma[z]_d := \Sigma[z] \cap \mathbb{R}[z]_d$. For $p \in \mathbb{R}[z]$ and $\mathcal{R}, \mathcal{S} \subseteq \mathbb{R}[z]$, we define $p \cdot \mathcal{R} := \{p \cdot q : q \in \mathcal{R}\}$ and $\mathcal{R} + \mathcal{S} := \{r + s : r \in \mathcal{R},\ s \in \mathcal{S}\}$.

Given a tuple $g = (g_1, \dots, g_m) \subseteq \mathbb{R}[z]$, the *quadratic module* of $\mathbb{R}[z]$ generated by g is the set

$$\mathrm{Qmod}(g) := \Sigma[z] + g_1 \cdot \Sigma[z] + \cdots + g_m \cdot \Sigma[z], \tag{2.1}$$

and the $2d$th truncation of $\mathrm{Qmod}(g)$ is the set

$$\mathrm{Qmod}(g)_{2d} := \Sigma[z]_{2d} + g_1 \cdot \Sigma[z]_{2d-\deg(g_1)} + \cdots + g_m \cdot \Sigma[z]_{2d-\deg(g_m)}. \tag{2.2}$$

For a tuple $h = (h_1, \dots, h_\ell) \subset \mathbb{R}[z]$, the *ideal* of $\mathbb{R}[z]$ generated by h is the set

$$\mathrm{Ideal}(h) := h_1 \cdot \mathbb{R}[z] + \cdots + h_\ell \cdot \mathbb{R}[z],$$

and the $2d$th truncation of $\mathrm{Ideal}(h)$ is the set

$$\mathrm{Ideal}(h)_{2d} := h_1 \cdot \mathbb{R}[z]_{2d-\deg(h_1)} + \cdots + h_\ell \cdot \mathbb{R}[z]_{2d-\deg(h_\ell)}.$$

For two polynomial tuples h and g, denote

$$\mathrm{IQ}(h, g) := \mathrm{Ideal}(h) + \mathrm{Qmod}(g), \quad \mathrm{IQ}(h, g)_{2d} := \mathrm{Ideal}(h)_{2d} + \mathrm{Qmod}(g)_{2d}. \tag{2.3}$$

Then, it is clear that every polynomial $p \in \mathrm{IQ}(h, g) \subseteq \mathbb{R}[z]$ is nonnegative over the set $\mathcal{K} := \{z \in \mathbb{R}^r : h(z) = 0,\ \ g(z) \ge 0\}$. Conversely, when $\mathrm{IQ}(h, g)$ is *archimedean*, i.e., when there exists $p \in \mathrm{IQ}(h, g)$ such that $\{z \in \mathbb{R}^r : p(z) \ge 0\}$ is compact (see [12]), all positive polynomials over $\mathcal{K}$ are in $\mathrm{IQ}(h, g)$. This result is referred to as Putinar's Positivstellensatz [32]. Moreover, when $h = 0$ has finitely many real roots, or when some general optimality conditions hold, a polynomial $f \in \mathbb{R}[z]$ is nonnegative over $\mathcal{K}$ if and only if $f \in \mathrm{IQ}(h, g)_{2d}$ for all d that is sufficiently large (see [19, 21]).

Throughout the paper, $x = (x_1, \dots, x_n)$ is the tuple of n variables. Given the csp $(\mathcal{I}_1, \dots, \mathcal{I}_s)$, for each $i \in [s]$, we fix a certain ordering for elements in $\mathcal{I}_i$ and denote by $x^{(i)}$ the tuple of variables $(x_k : k \in \mathcal{I}_i)$. The jth variable of $x^{(i)}$, denoted by $x_j^{(i)}$, corresponds to the variable x_k if j is the order of k in $\mathcal{I}_i$. For example, if $\mathcal{I}_1$ is ordered as $(1, 3, 5, 6)$, then $x_2^{(1)} = x_3$. For polynomial $p \in \mathbb{R}[x]$, denote by $\nabla p \in \mathbb{R}[x]^n$ the gradient of p and

$$\nabla_i p := \begin{bmatrix} \frac{\partial p}{\partial x_1^{(i)}} & \cdots & \frac{\partial p}{\partial x_{n_i}^{(i)}} \end{bmatrix}^\top \in \mathbb{R}[x]^{n_i}. \tag{2.4}$$

When the dimension of the ambient space is clear, we use e_i to denote the i-th standard basis vector whose ith entry is 1 while all other entries are zeros. For $k \in \mathcal{I}_i$, denote

$$e_k^{(i)} := e_j \in \mathbb{R}^{n_i}, \tag{2.5}$$

where j is the order of k in the tuple $\mathcal{I}_i$. For instance, if $\mathcal{I}_1$ is ordered as $(1,3,5,6)$, then $e_3^{(1)} = e_2 \in \mathbb{R}^4$.

2.2. Moment-SOS relaxation. Denote by $f_{\min}$ the optimal value of the polynomial optimization problem (1.1). Denote by $\mathcal{K}$ the feasible set of (1.1), i.e., $\mathcal{K} := \{x \in \mathbb{R}^n : h(x) = 0,\ g(x) \geq 0\}$. Then finding the global minimum of (1.1) is equivalent to

$$\left\{ \begin{array}{cl} \max & \gamma \\ \text{s.t.} & f - \gamma \in \mathcal{P}_{d_0}(\mathcal{K}). \end{array} \right. \tag{2.6}$$

In the above, d_0 is the degree of f, and $\mathcal{P}_{d_0}(\mathcal{K})$ is the cone of nonnegative polynomials over $\mathcal{K}$ with degrees not greater than d_0. A computationally tractable relaxation for (2.6) is called *the Moment-SOS relaxation*. Given the relaxation order $d \in \mathbb{N}$ such that $2d \geq \max\{\deg(f), \deg(g), \deg(h)\}$, the dth order SOS relaxation of (2.6) (and (1.1)) is

$$\left\{ \begin{array}{cl} \max & \gamma \\ \text{s.t.} & f - \gamma \in \mathrm{IQ}(h,g)_{2d}. \end{array} \right. \tag{2.7}$$

Its dual problem corresponds to the so-called dth order moment relaxation of (1.1), and this primal-dual pair is referred to as the Moment-SOS relaxation. Both (2.7) and its dual problems can be written as SDP problems. We refer to [6, 10, 12, 13, 14, 20, 22, 24] for more references about polynomial optimization and moment problems.

For a relaxation order d, denote by θ_d the optimal value of (2.7). Clearly θ_d provides a lower bound of $f_{\min}$, i.e. $\theta_d \leq f_{\min}$. Convergence of the Moment-SOS relaxation relies on Putinar's Positivestellenstaz [32].

THEOREM 2.1 ([10]). *If $IQ(g,h)$ is archimedean, then* $\lim_{d\to+\infty} \theta_d = f_{\min}$.

We would like to remark that under some conditions, the Moment-SOS relaxations have finite convergence, i.e., $\theta_d = f_{\min}$ for all d that is big enough. We refer to [2, 3, 8, 19, 21] for more related work. The Moment-SOS relaxations have been implemented in the software `GloptiPoly 3` [7]. In this paper, we also call Moment-SOS relaxations ”dense relaxations” or ”dense SOS relaxations” to distinguish them from SOS relaxations exploiting the sparsity.

2.3. Correlatively sparse SOS relaxation. Let us consider the problem (1.2) with csp $(\mathcal{I}_1, \ldots, \mathcal{I}_s)$. For polynomial tuples $h^{(i)}, g^{(i)} \in \mathbb{R}[x^{(i)}]$, we denote by

$$\mathrm{IQ}_{\mathcal{I}_i}(h^{(i)}, g^{(i)})$$

the set given by (2.1-2.3) with $z = x^{(i)}$. To exploit the correlative sparsity of problem (1.2), we consider the following relaxation for problem (1.2):

$$\left\{ \begin{array}{cl} \max & \gamma \\ \text{s.t.} & f - \gamma \in \mathrm{IQ}_{\mathcal{I}_1}\left(h^{(1)}, g^{(1)}\right)_{2d} + \cdots + \mathrm{IQ}_{\mathcal{I}_s}\left(h^{(s)}, g^{(s)}\right)_{2d}. \end{array} \right. \tag{2.8}$$

We refer to (2.8) as the dth order *CS-SOS relaxation* of (1.2) [11, 25, 16, 34], and denote its optimal value by ρ_d. To demonstrate the convergence results for CS-SOS relaxations, we need the following property of csps.

DEFINITION 2.2. *We say that the csp $(\mathcal{I}_1, \dots, \mathcal{I}_s)$ satisfies the running intersection property (RIP) if for every $i \in [s-1]$, there exists $t \le i$ such that*

$$\mathcal{I}_{i+1} \bigcap \bigcup_{j=1}^{i} \mathcal{I}_j \subseteq \mathcal{I}_t. \tag{2.9}$$

Convergence of the CS-SOS relaxation is derived from the following sparse version of Putinar's Positivestellenstaz.

THEOREM 2.3 ([5, 9, 11]). *Suppose $(\mathcal{I}_1, \dots, \mathcal{I}_s)$ satisfies the RIP property, and $\mathrm{IQ}_{\mathcal{I}_i}(g^{(i)}, h^{(i)})$ is archimedean for each $i \in [s]$. If $f(x) := f_1(x^{(1)}) + \cdots + f_s(x^{(s)})$ is positive on the semi-algebraic set $\bigcap_{i=1}^{s}\{x \in \mathbb{R}^n : g^{(i)}(x) \ge 0, h^{(i)}(x) = 0\}$, then*

$$f \in \mathrm{IQ}_{\mathcal{I}_1}\left(h^{(1)}, g^{(1)}\right) + \cdots + \mathrm{IQ}_{\mathcal{I}_s}\left(h^{(s)}, g^{(s)}\right).$$

Therefore, under the same conditions as that in Theorem 2.3, we have:

$$\lim_{d \to +\infty} \rho_d = f_{\min}. \tag{2.10}$$

Aside from the correlative sparsity, one can also exploit the *term sparsity* of polynomial optimization problems, or combine both kinds of sparsity to obtain the so-called *correlative and term sparsity SOS relaxations* (CS-TSSOS) of (1.2), whose convergence is guaranteed with the term sparsity being given by the *maximal chordal extension* when the CS-SOS relaxation is convergent [37]. Since this paper mainly concerns correlative sparsity, we refer to [16, 36, 37] for more details on the exploitation of term sparsity. The CS-TSSOS relaxations have been recently implemented in the software `TSSOS` [15].

2.4. Optimality conditions and Lagrange multiplier expressions. For the polynomial optimization problem (1.1), the Karush-Kuhn-Tucker (KKT) conditions can be described by the following polynomial system in $(x, \lambda) \in \mathbb{R}^{n+m+\ell}$:

$$\left\{ \begin{array}{c} \nabla f(x) = \sum_{j=1}^{m} \lambda_j \nabla g_j(x) + \sum_{j=1}^{\ell} \lambda_{m+j} \nabla h_j(x), \\ h(x) = 0,\ 0 \le \lambda_{1:m} \perp g(x) \ge 0. \end{array} \right. \tag{2.11}$$

The pair (x, λ) satisfying (2.11) is called a KKT pair, and the first component x of a KKT pair is called a KKT point of (1.1). Under some constraint qualification conditions, every minimizer of (1.1), if it exists, must be a KKT point. In this case, minimizing f over the KKT system (2.11) returns the same optimal value and optimal solutions as the original problem (1.1). Moreover, conditions guaranteeing the convergence of the dense SOS relaxations are milder for the minimization over the KKT ideal than that for the original problem (1.1). In particular, convergence can still occur even when the semi-algebraic set given by (2.11) is noncompact [4, 26].

A drawback, however, of working on the KKT system (2.11) rather than on the original feasible region $\mathcal{K} \subseteq \mathbb{R}^n$ is the augmentation of the number of variables from n to $n+m+\ell$, which causes a significant increase on the computational cost. To deal with this undesired complexity growth, Nie [23] proposed polynomial Lagrange multipliers expressions. For the polynomial optimization problem (1.1), let $\hat{m} := m + \ell$, and let

$c := (c_1, \ldots, c_{\hat{m}})$ be an enumeration for the constraining pair (g, h). We denote

$$G(x) := \begin{bmatrix} \nabla c_1(x) & \nabla c_2(x) & \cdots & \nabla c_{\hat{m}}(x) \\ c_1(x) & 0 & \cdots & 0 \\ 0 & c_2(x) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & c_{\hat{m}}(x) \end{bmatrix}, \quad \mathbf{f}(x) := \begin{bmatrix} \nabla f(x) \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{2.12}$$

Then, the following equation holds at every KKT pair (x, λ):

$$G(x) \cdot \lambda = \mathbf{f}(x). \tag{2.13}$$

If there exists a matrix of polynomials $L \in \mathbb{R}[x]^{\hat{m} \times n}$ and $D \in \mathbb{R}[x]^{\hat{m} \times \hat{m}}$ such that

$$\begin{bmatrix} L(x) & D(x) \end{bmatrix} G(x) = \mathbf{I}_{\hat{m}}, \ \ \forall x \in \mathbb{R}^n, \tag{2.14}$$

then the Lagrange multipliers λ can be expressed as polynomials in x:

$$\begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{\hat{m}} \end{bmatrix} = \begin{bmatrix} p_1(x) \\ \vdots \\ p_{\hat{m}}(x) \end{bmatrix} := L(x) \nabla f(x). \tag{2.15}$$

The polynomial vector $p(x) := (p_1(x), \ldots, p_{\hat{m}}(x))$ is called the *Lagrange multiplier expression* (LME). Denote

$$c_{eq}(x) := \begin{bmatrix} \nabla f(x) - \sum_{j=1}^{m} p_j(x) \nabla g_j(x) - \sum_{j=1}^{\ell} p_{m+j}(x) \nabla h_j(x) \\ h(x) \\ p_{1:m}(x) \circ g(x) \end{bmatrix},$$

and

$$c_{in}(x) := \begin{bmatrix} p_{1:m}(x) \\ g(x) \end{bmatrix}.$$

Then, $x \in \mathbb{R}^n$ is a KKT point if and only if x satisfies $c_{eq}(x) = 0, \ c_{in}(x) \geq 0$. Based on the LME (2.15), Nie [23] proposed the following reformulation of (1.1):

$$\left\{ \begin{array}{cl} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & c_{eq}(x) = 0, \quad c_{in}(x) \geq 0 \end{array} \right. \tag{2.16}$$

It is clear that when the minimum of (1.1) is attained at some KKT points, the optimal values of (1.1) and (2.16) are identical. In fact, the existence of LMEs guarantees that every minimizer of (1.1), if it exists, must be a KKT point [23, Proposition 5.1], thus solving (1.1) is equivalent to solving the reformulation (2.16). When Moment-SOS relaxations are applied, finite convergence is guaranteed under some generic conditions:

THEOREM 2.4. *[23, Theorem 3.3] Suppose LMEs exist and (2.16) has a nonempty feasible set. Denote by θ_d the optimal value of the dth order SOS relaxation* (2.7) *of the polynomial optimization problem* (2.16)*. Then, we have $f_{\min} = \theta_d$ holds for all d big enough, if $IQ(c_{eq}, c_{in})$ is archimedean and the minimum value of (1.1) is attained at a KKT point.*

Recently, LMEs have been widely used in various problems given by polynomial functions, such as bilevel polynomial optimization, Nash equilibrium problems, tensor computation, etc. We refer to [27, 28, 29, 30, 31] for applications of LMEs.

One wonders when LMEs exist, i.e., when there exist matrices $L(x), D(x)$ such that (2.14) holds. We say that the constraining tuple (g,h) is *nonsingular* if the matrix $G(x)$ given in (2.12) has a full column rank for all $x \in \mathbb{C}^n$. For (1.1), LMEs exist if and only if its constraining tuple is nonsingular [23, Proposition 5.1]. We would like to remark that when the polynomials $c_1, \dots, c_{\hat{m}}$ are generic[1], the nonsingularity condition holds. However, there are cases when LMEs do not exist; see Example 3.1 for a concrete example and also [23, 28] for more details. In the following example, we give the matrices $L(x)$ and $D(x)$ for a special box-constrained problem. The general methodology for formulating LMEs can be found in Appendix A.

Example 2.5. Consider the polynomial optimization problem with box constraints

$$
\text{(2.17)} \qquad \left\{\begin{array}{cl} \min\limits_{x\in\mathbb{R}^4} & f(x_1,x_2,x_3,x_4) := x_1^4x_2^2 + x_1^2x_2^4 + x_3^6 \\ & \qquad\qquad\qquad -3x_1^2x_2^2x_3^2 + x_3^3 + x_3x_4^2 - 2x_3^2x_4 \\ \text{s.t.} & x_1 \geq 0,\ 1-x_1 \geq 0,\ x_2 \geq 0,\ 1-x_2 \geq 0, \\ & x_3 \geq 0,\ 1-x_3 \geq 0,\ x_4 \geq 0,\ 1-x_4 \geq 0. \end{array}\right.
$$

Note that in this problem, since all variables are nonnegative, by the inequality of arithmetic and geometric means, we have

$$
\begin{aligned} x_1^4x_2^2 + x_1^2x_2^4 + x_3^6 &\geq 3\sqrt[3]{x_1^4x_2^2 \cdot x_1^2x_2^4 \cdot x_3^6} = 3x_1^2x_2^2x_3^2, \\ x_3^3 + x_3x_4^2 &\geq 2\sqrt{x_3^3 \cdot x_3x_4^2} = 2x_3^2x_4, \end{aligned}
$$

where the equailities hold when $x_1 = x_2 = \cdots = x_4$. So the global minimum of (2.17) is 0 with minimizers (t,t,t,t) for all $t \in [0,1]$. Let $g(x) := (g_1(x), \dots, g_8(x))$ with

$$
\text{(2.18)} \qquad \begin{array}{llll} g_1(x) = x_1, & g_2(x) = 1-x_1, & g_3(x) = x_2, & g_4(x) = 1-x_2, \\ g_5(x) = x_3, & g_6(x) = 1-x_3, & g_7(x) = x_4, & g_8(x) = 1-x_4. \end{array}
$$

The constraining tuple g is nonsingular and (2.14) holds with

$$
L(x) = \operatorname{diag}(L_1(x), L_2(x), L_3(x), L_4(x)), \quad D(x) = \operatorname{diag}(D_1(x), D_2(x), D_3(x), D_4(x))
$$

being block-diagonal matrices. The matrices in the diagonal of L are given by $L_i(x) = \begin{bmatrix} 1-x_i \\ -x_i \end{bmatrix}$ and the matrices in the diagonal of D are given by $D_i(x) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ for each $i \in [4]$. Accordingly, the LMEs are

$$
\text{(2.19)} \qquad p_{2i-1}(x) = (1-x_i)\cdot\frac{\partial f}{\partial x_i}(x), \quad p_{2i}(x) = -x_i \cdot \frac{\partial f}{\partial x_i}(x), \quad (i = 1, \dots, 4).
$$

In particular, $p_5(x)$, $p_6(x)$ can be explicitly written as

$$
\text{(2.20)} \qquad \begin{aligned} p_5(x) = {} & 6(x_1^2x_2^2x_3^2 - x_3^6 - x_1^2x_2^2x_3 + x_3^5) \\ & -3x_3^3 + 4x_3^2x_4 - x_3x_4^2 + 3x_3^2 - 4x_3x_4 + x_4^2, \\ p_6(x) = {} & 6x_1^2x_2^2x_3^2 - 6x_3^6 - 3x_3^3 + 4x_3^2x_4 - x_3x_4^2, \end{aligned}
$$

which involve all the four variables x_1, x_2, x_3, x_4.

[1]We say a property holds generically if it holds for all points of input data but a set of Lebesgue measure zero.

3. Correlatively sparse LMEs and reformulations.

We consider polynomial optimization problem (1.2) with the csp $(\mathcal{I}_1, \dots, \mathcal{I}_s)$ satisfying the RIP. For $i \in [s]$, we denote by $G^{(i)}$ the polynomial matrix G given in (2.12) associated with $(g^{(i)}, h^{(i)})$. That is, if we let $c^{(i)} = (g^{(i)}, h^{(i)})$, and $\hat{m}_i := m_i + \ell_i$, then

$$G^{(i)}(x^{(i)}) := \begin{bmatrix} \nabla_i c_1^{(i)}(x^{(i)}) & \nabla_i c_2^{(i)}(x^{(i)}) & \cdots & \nabla_i c_{\hat{m}_i}^{(i)}(x^{(i)}) \\ c_1^{(i)}(x^{(i)}) & 0 & \cdots & 0 \\ 0 & c_2^{(i)}(x^{(i)}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & c_{\hat{m}_i}^{(i)}(x^{(i)}) \end{bmatrix}. \tag{3.1}$$

As mentioned in Section 2.4, one can reformulate polynomial optimization problems with LMEs, from which the Moment-SOS relaxation gives a tighter lower bound for the polynomial optimization. To apply LMEs, the KKT system of (1.2) corresponds to the following semialgebraic set on $x \in \mathbb{R}^n$, $\lambda^{(1)} \in \mathbb{R}^{\hat{m}_1}, \dots, \lambda^{(s)} \in \mathbb{R}^{\hat{m}_s}$:

$$\left\{ \begin{array}{c} \nabla f_1(x) + \cdots + \nabla f_s(x) = \sum_{i=1}^{s} \left(\sum_{j=1}^{m_i} \lambda_j^{(i)} \nabla g_j^{(i)}(x) + \sum_{j=1}^{\ell_i} \lambda_{m_i+j}^{(i)} \nabla h_j^{(i)}(x) \right), \\ h^{(i)}(x) = 0, \ \ i \in [s], \\ 0 \le \lambda_{1:m_i}^{(i)} \perp g^{(i)}(x) \ge 0, \ \ i \in [s]. \end{array} \right. \tag{3.2}$$

Hereinafter, we additionally assume that the nonsingularity condition holds for the constraining pair $(g^{(i)}, h^{(i)})$ within every $\mathcal{I}_i$. That is:

Assumption 1. For each $i \in [s]$, there exist polynomial matrices $L^{(i)}(x^{(i)}) \in \mathbb{R}[x^{(i)}]^{\hat{m}_i \times n_i}$ and $D^{(i)}(x^{(i)}) \in \mathbb{R}[x^{(i)}]^{\hat{m}_i \times \hat{m}_i}$ such that

$$\begin{bmatrix} L^{(i)}(x^{(i)}) & D^{(i)}(x^{(i)}) \end{bmatrix} G^{(i)}(x^{(i)}) = \mathbf{I}_{\hat{m}_i}. \tag{3.3}$$

By [23, Proposition 5.2], (3.3) holds if and only if the matrix $G^{(i)}(x^{(i)})$ have full column rank for all $x^{(i)} \in \mathbb{C}^{n_i}$. For such cases, we say the pair $(g^{(i)}, h^{(i)})$ is *nonsingular*. This is satisfied if all polynomials in $g^{(i)}$ and $h^{(i)}$ are generic polynomials in $x^{(i)}$.

3.1. Limitation of the original LME for exploiting correlative sparsity.

For the polynomial optimization (1.2) with correlative sparsity, LMEs exist if and only if the constraining tuple of all the constraints is nonsingular, by [23, Proposition 5.1]. In general, Assumption 1 is a necessary but not sufficient condition of the nonsingularity for the constraining tuple (g, h) of all constraints in (1.2). This can be seen in the following example.

Example 3.1. Consider the following example with three variables $x = (x_1, x_2, x_3)$ and two constraints

$$\left\{ \begin{array}{cl} \min\limits_{x \in \mathbb{R}^3} & f_1(x_1, x_2) + f_2(x_2, x_3) \\ \text{s.t.} & 1 - x_1^2 - x_2^2 \ge 0 \\ & 1 - x_2^2 - x_3^2 \ge 0 \end{array} \right. \tag{3.4}$$

Let $\mathcal{I}_1 = \{1, 2\}$ and $\mathcal{I}_2 = \{2, 3\}$. Then (3.4) has the csp $(\mathcal{I}_1, \mathcal{I}_2)$ with

$$g^{(1)} = \left(1 - x_1^2 - x_2^2\right), \quad g^{(2)} = \left(1 - x_2^2 - x_3^2\right), \quad h^{(1)} = h^{(2)} = \emptyset. \tag{3.5}$$

The matrix $G(x)$ associated to (3.4) is

$$G(x) = \begin{bmatrix} -2x_1 & 0 \\ -2x_2 & -2x_2 \\ 0 & -2x_3 \\ 1-x_1^2-x_2^2 & 0 \\ 0 & 1-x_2^2-x_3^2 \end{bmatrix},$$

whose rank is 1 at $x=(0,1,0)$. Thus the constraining tuple of (3.4) is not nonsingular, and LMEs do not exist. On the other hand, we have

$$G^{(1)}(x_1,x_2) = \begin{bmatrix} -2x_1 \\ -2x_2 \\ 1-x_1^2-x_2^2 \end{bmatrix}, \quad G^{(2)}(x_2,x_3) = \begin{bmatrix} -2x_2 \\ -2x_3 \\ 1-x_2^2-x_3^2 \end{bmatrix}.$$

One may check that Assumption 1 holds with

$$L^{(1)}(x_1,x_2) = \begin{bmatrix} -\frac{1}{2}x_1 & -\frac{1}{2}x_2 \end{bmatrix}, \quad L^{(2)}(x_2,x_3) = \begin{bmatrix} -\frac{1}{2}x_2 & -\frac{1}{2}x_3 \end{bmatrix}, \tag{3.6}$$

and $D^{(1)} = D^{(2)} = 1$.

Remark 3.2. See also Example 5.1(ii), Example 5.2, Example 5.5 and Example 5.7 in Section 5 for cases which satisfy Assumption 1 but do not admit LMEs.

Another concern related to the original LME approach is that the LME reformulation (2.16), if exists, usually cannot inherit the csp of (1.2). Indeed, the LME reformulation (2.16) may have constraints that involve all the variables, as demonstrated by the following example.

Example 3.3. Consider the polynomial optimization problem (2.17) with box constraints. Let $\mathcal{I}_1 = \{1,2,3\}$ and $\mathcal{I}_2 = \{3,4\}$. Then (2.17) has the csp $(\mathcal{I}_1,\mathcal{I}_2)$ with $h^{(1)} = h^{(2)} = \emptyset$ and

$$\begin{aligned} &f_1(x^{(1)}) = x_1^4x_2^2 + x_1^2x_2^4 + x_3^6 - 3x_1^2x_2^2x_3^2, \quad f_2(x^{(2)}) = x_3^3 + x_3x_4^2 - 2x_3^2x_4, \\ &g^{(1)}(x^{(1)}) = (x_1,\, 1-x_1,\, x_2,\, 1-x_2), \quad g^{(2)}(x^{(2)}) = (x_3,\, 1-x_3,\, x_4,\, 1-x_4). \end{aligned} \tag{3.7}$$

In view of (2.20), the LME reformulation (2.16) of (2.17) does not have correlative sparsity, as the nonnegativity conditions for Lagrange multipliers $p_5(x) \geq 0$ and $p_6(x) \geq 0$ involve all variables.

In the next two subsections, we provide a systematic method to construct LMEs for (1.2) which leverages the correlative sparsity pattern $(\mathcal{I}_1,\ldots,\mathcal{I}_s)$.

3.2. Correlatively sparse LMEs: two blocks. We begin with the case of two blocks, i.e., $s=2$. Before giving a formal presentation of our approach, we would like to expose the underlying idea through the following example of three variables

$$\begin{cases} \min & f_1(x_1,x_2) + f_2(x_2,x_3) \\ \text{s.t.} & g_1(x_1,x_2) \geq 0, \\ & g_2(x_2,x_3) \geq 0. \end{cases} \tag{3.8}$$

The problem (3.8) has the csp $(\mathcal{I}_1,\mathcal{I}_2)$ with $\mathcal{I}_1 = \{1,2\}$ and $\mathcal{I}_2 = \{2,3\}$. Recall that for each $i \in [s]$, the partial gradient ∇_i is defined as in (2.4). Under Assumption 1, there exist polynomial matrices $L^{(1)} \in \mathbb{R}[x_1, x_2]^2$, $D^{(1)} \in \mathbb{R}[x_1, x_2]$, $L^{(2)} \in \mathbb{R}[x_2, x_3]^2$ and $D^{(2)} \in \mathbb{R}[x_2, x_3]$ such that for each $i = 1, 2$,

$$\begin{bmatrix} L^{(i)}(x_1, x_2) & D^{(i)}(x_1, x_2) \end{bmatrix} \begin{bmatrix} \nabla_i g_i(x_1, x_2) \\ g_i(x_1, x_2) \end{bmatrix} = 1. \tag{3.9}$$

The KKT system of (3.8) is

$$\left\{ \begin{array}{rcl} \frac{\partial f_1}{\partial x_1}(x_1, x_2) & = & \lambda_1 \cdot \frac{\partial g_1}{\partial x_1}(x_1, x_2), \\ \frac{\partial f_1}{\partial x_2}(x_1, x_2) + \frac{\partial f_2}{\partial x_2}(x_2, x_3) & = & \lambda_1 \cdot \frac{\partial g_1}{\partial x_2}(x_1, x_2) + \lambda_2 \cdot \frac{\partial g_2}{\partial x_2}(x_2, x_3), \\ \frac{\partial f_2}{\partial x_3}(x_2, x_3) & = & \lambda_2 \cdot \frac{\partial g_2}{\partial x_3}(x_2, x_3), \\ 0 \ \le \ g_1(x_1, x_2) & \perp & \lambda_1 \ \ge \ 0, \\ 0 \ \le \ g_2(x_2, x_3) & \perp & \lambda_2 \ \ge \ 0. \end{array} \right. \tag{3.10}$$

Clearly the csp structure is broken when f_1, f_2, g_1, g_2 are dense polynomials, due to the second equation above. Introducing an auxiliary variable ν, we rewrite (3.10) as

$$\left\{ \begin{array}{rcl} \nabla_1 f_1(x_1, x_2) + \begin{bmatrix} 0 \\ \nu \end{bmatrix} & = & \lambda_1 \cdot \nabla_1 g_1(x_1, x_2), \\ 0 \le \ g_1(x_1, x_2) & \perp & \lambda_1 \ \ge 0, \\ \nabla_2 f_2(x_2, x_3) - \begin{bmatrix} \nu \\ 0 \end{bmatrix} & = & \lambda_2 \cdot \nabla_2 g_2(x_2, x_3), \\ 0 \le \ g_2(x_2, x_3) & \perp & \lambda_2 \ \ge 0, \end{array} \right. \tag{3.11}$$

Thus by (3.9), for any $(x_1, x_2, \lambda_1, \lambda_2, \nu)$ satisfying (3.11), we must have

$$\begin{aligned} \lambda_1 &= L^{(1)}(x_1, x_2) \left(\nabla_1 f_1(x_1, x_2) + \begin{bmatrix} 0 \\ \nu \end{bmatrix} \right), \\ \lambda_2 &= L^{(2)}(x_2, x_3) \left(\nabla_2 f_2(x_2, x_3) - \begin{bmatrix} \nu \\ 0 \end{bmatrix} \right). \end{aligned} \tag{3.12}$$

Under some constraint qualification conditions, we arrive at a reformulation for (3.8) which possess the csp with two blocks of variables

$$(x_1, x_2, \nu), \ \ (x_2, x_3, \nu)$$

by plugging (3.12) back into (3.11) to replace λ_1 and λ_2.

Example 3.4. Consider the polynomial optimization problem (3.4) as a special case of (3.8). Recall that Assumption 1 holds with $L^{(1)}$ and $L^{(2)}$ given in (3.6). In view of (3.12), we have

$$\begin{aligned} \lambda_1 = p^{(1)}(x_1, x_2, \nu) &:= -\frac{x_1}{2} \frac{\partial f_1}{\partial x_1}(x_1, x_2) - \frac{x_2}{2} \frac{\partial f_1}{\partial x_2}(x_1, x_2) - \frac{x_2}{2}\nu, \\ \lambda_2 = p^{(2)}(x_2, x_3, \nu) &:= -\frac{x_2}{2} \frac{\partial f_2}{\partial x_2}(x_2, x_3) - \frac{x_3}{2} \frac{\partial f_2}{\partial x_3}(x_2, x_3) + \frac{x_2}{2}\nu. \end{aligned} \tag{3.13}$$

Suppose the minimum value of (3.4) is attained at a KKT point x^*. Then there exists $\nu^* \in \mathbb{R}$ such that (3.11) holds at (x^*, ν^*) with λ_1, λ_2 given by (3.13). Taking (3.11) as constraints with λ_i being substituted by $p^{(i)}$ for every $i = 1, 2$, we arrive at the following optimization problem

$$
(3.14)\qquad \begin{cases} \min\limits_{x,\nu} & f_1(x_1,x_2) + f_2(x_2,x_3) \\ \text{s.t.} & \nabla_1 f_1(x_1,x_2) + \begin{bmatrix} 0 \\ \nu \end{bmatrix} = -2p^{(1)}(x_1,x_2,\nu)\cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \\ & 0 \le \left(1 - x_1^2 - x_2^2\right) \perp p^{(1)}(x_1,x_2,\nu) \ge 0, \\ & \nabla_2 f_2(x_2,x_3) - \begin{bmatrix} \nu \\ 0 \end{bmatrix} = -2p^{(2)}(x_2,x_3,\nu)\cdot \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}, \\ & 0 \le \left(1 - x_2^2 - x_3^2\right) \perp p^{(2)}(x_2,x_3,\nu) \ge 0. \end{cases}
$$

Then (x^*, ν^*) is a global minimizer for (3.14). As we will formally introduce later, polynomials $p^{(1)}, p^{(2)}$ representing λ_1, λ_2 are called *correlatively sparse LMEs* (CS-LMEs), and (3.14) is called the *CS-LME reformulation* for (3.4).

Recall from Example 3.1 that (3.4) does not admit LMEs, thus the LME reformulation (2.16) is not available for (3.4). One may consider a reformulation of (3.4) using the KKT system (3.10) by taking λ_1, λ_2 as new variables. Then the total number of variables in this approach is 5 and there is no correlative sparsity anymore. Instead, by appropriately adding extra variable ν, we obtained the CS-LME reformulation (3.14) which maintains to a degree the original csp structure: we have 3 variables in each of the two blocks.

Now we present formally the CS-LME approach for the polynomial optimization problem (1.2) with two block csp structure. Given the csp $(\mathcal{I}_1, \mathcal{I}_2)$, we introduce extra variables $\nu := (\nu_k)_{k\in\mathcal{I}_1\cap\mathcal{I}_2}$. Then, the gradient of the objective function $\nabla f_1(x) + \nabla f_2(x)$ can be split into two terms such that one only involves $(x^{(1)}, \nu)$ and the other one only has $(x^{(2)}, \nu)$. Recall that for $i \in \{1, 2\}$ and $k \in \mathcal{I}_i$, the vector $e_k^{(i)}$ is defined in (2.5). Let

$$
(3.15)\qquad \begin{aligned} F^{(1)}(x^{(1)},\nu) &:= \nabla_1 f_1(x^{(1)}) + \textstyle\sum_{k\in\mathcal{I}_1\cap\mathcal{I}_2} \nu_k e_k^{(1)} \in \mathbb{R}^{n_1}, \\ F^{(2)}(x^{(2)},\nu) &:= \nabla_2 f_2(x^{(2)}) - \textstyle\sum_{k\in\mathcal{I}_1\cap\mathcal{I}_2} \nu_k e_k^{(2)} \in \mathbb{R}^{n_2}. \end{aligned}
$$

Then, $(x, \lambda^{(1)}, \lambda^{(2)})$ is a KKT tuple of (1.2) if and only if there exists $\nu = (\nu_k)_{k\in\mathcal{I}_1\cap\mathcal{I}_2}$ such that

$$
(3.16)\qquad \begin{cases} F^{(1)}(x^{(1)},\nu) = \sum_{j=1}^{m_1} \lambda_j^{(1)} \nabla_1 g_j^{(1)}(x^{(1)}) + \sum_{j=1}^{\ell_1} \lambda_{m_1+j}^{(1)} \nabla_1 h_j^{(1)}(x^{(1)}), \\ F^{(2)}(x^{(2)},\nu) = \sum_{j=1}^{m_2} \lambda_j^{(2)} \nabla_2 g_j^{(2)}(x^{(2)}) + \sum_{j=1}^{\ell_2} \lambda_{m_2+j}^{(2)} \nabla_2 h_j^{(2)}(x^{(2)}), \\ h^{(1)}(x^{(1)}) = 0, \quad h^{(2)}(x^{(2)}) = 0, \\ 0 \le \lambda_{1:m_1}^{(1)} \perp g^{(1)}(x^{(1)}) \ge 0, \quad 0 \le \lambda_{1:m_2}^{(2)} \perp g^{(2)}(x^{(2)}) \ge 0. \end{cases}
$$

Under Assumption 1, if we let

$$
p^{(1)}(x^{(1)},\nu) := L^{(1)}(x^{(1)})F^{(1)}(x^{(1)},\nu), \quad p^{(2)}(x^{(2)},\nu) := L^{(2)}(x^{(2)})F^{(2)}(x^{(2)},\nu),
$$

then by (3.3), for any $(x, \lambda^{(1)}, \lambda^{(2)}, \nu)$ satisfying (3.16), we have

$$
\lambda^{(1)} = p^{(1)}(x^{(1)},\nu), \quad \lambda^{(2)} = p^{(2)}(x^{(2)},\nu).
$$

The polynomial vectors $p^{(1)}$, $p^{(2)}$ are called *correlatively sparse Lagrange multiplier expression* (CS-LME) for $\lambda^{(1)}$ and $\lambda^{(2)}$ respectively. Replacing $\lambda^{(1)}$, $\lambda^{(2)}$ by the polynomial vectors $p^{(1)}(x^{(1)},\nu)$ and $p^{(2)}(x^{(2)},\nu)$, we get the following reformulation of (1.2):

$$
(3.17)\qquad \begin{cases} \min\limits_{x\in\mathbb{R}^n} & f_1(x^{(1)})+f_2(x^{(2)}) \\ \text{s.t.} & F^{(i)}(x^{(i)},\nu)=\sum_{j=1}^{m_i} p_j^{(i)}(x^{(i)},\nu)\nabla_i g_j^{(i)}(x^{(i)}) \\ & \qquad\qquad +\sum_{j=1}^{\ell_i} p_{m_i+j}^{(i)}(x^{(i)},\nu)\nabla_i h_j^{(i)}(x^{(i)}), \quad (i=1,2) \\ & h^{(i)}(x^{(i)})=0, \quad 0\le p_{1:m_i}^{(i)}(x^{(i)},\nu)\perp g^{(i)}(x^{(i)})\ge 0. \quad (i=1,2) \end{cases}
$$

The reformulation (3.14) in Example 3.4 is a special case of (3.17). One may check the polynomial optimization problem (3.17) has the csp with two blocks of variables:

$$
(x^{(1)},\nu), \quad (x^{(2)},\nu).
$$

Example 3.5. Consider the polynomial optimization problem with box constraints (2.17) in Example 2.5. Its csp is given in Example 3.3, and we have $\mathcal{I}_1\cap\mathcal{I}_2=\{3\}$. So we need to introduce a new variable $\nu\in\mathbb{R}$. The $f_1,f_2,g^{(1)},g^{(2)}$ are given as in (3.7), and we let

$$
F^{(1)}(x^{(1)},\nu)=\nabla_1 f_1(x^{(1)})+\nu e_3^{(1)}, \quad F^{(2)}(x^{(2)},\nu)=\nabla_2 f_2(x^{(2)})-\nu e_3^{(2)}.
$$

Moreover, denoting by $F_j^{(i)}$ the jth entry of $F^{(i)}$ for $j\in\{1,2\}$, we get CS-LMEs:

$$
(3.18)\qquad \begin{aligned} p_{2j-1}^{(1)}(x^{(1)},\nu)&=(1-x_j)F_j^{(1)}(x^{(1)},\nu), & p_{2j}^{(1)}(x^{(1)},\nu)&=-x_jF_j^{(1)}(x^{(1)},\nu), \\ p_{2j-1}^{(2)}(x^{(2)},\nu)&=(1-x_{2+j})F_j^{(2)}(x^{(2)},\nu), & p_{2j}^{(2)}(x^{(2)},\nu)&=-x_{2+j}F_j^{(2)}(x^{(2)},\nu). \end{aligned}
$$

Note that when CS-LMEs are given as above, the first equality constraints in (3.17) are reduced to one single equation $F_3^{(1)}(x^{(1)},\nu)=0$, and the complementarity conditions are reduced to

$$
x_j(1-x_j)F_j^{(1)}(x^{(1)},\nu)=0, \quad x_{2+j}(1-x_{2+j})F_j^{(2)}(x^{(2)},\nu)=0, \quad (j=1,2).
$$

Consequently, the CS-LME reformulation to (2.17) is

$$
(3.19)\qquad \begin{cases} \min\limits_{x\in\mathbb{R}^n} & x_1^4x_2^2+x_1^2x_2^4+x_3^6-3x_1^2x_2^2x_3^2+x_3^3+x_3x_4^2-2x_3^2x_4 \\ \text{s.t.} & x_j(1-x_j)F_j^{(1)}(x^{(1)},\nu)=0,\ x_{2+j}(1-x_{2+j})F_j^{(2)}(x^{(2)},\nu)=0, \quad (j=1,2) \\ & (1-x_j)F_j^{(1)}(x^{(1)},\nu)\ge 0,\ (1-x_{2+j})F_j^{(2)}(x^{(2)},\nu)\ge 0, \quad (j=1,2) \\ & -x_jF_j^{(1)}(x^{(1)},\nu)\ge 0,\ -x_{2+j}F_j^{(2)}(x^{(2)},\nu)\ge 0, \quad (j=1,2) \\ & F_3^{(1)}(x^{(1)},\nu)=0,\ 0\le x_1,\dots,x_4\le 1. \end{cases}
$$

Later in Section 5, we will compare the numerical performance of solving the CS-LME reformulation (3.19) of (2.17) with solving it directly and solving its LME reformulation (2.16), all using `CS-TSSOS` [37].

To summarize, the LME approach proposed by Nie [23] allows for tightening the classical Moment-SOS relaxation by incorporating necessary polynomial constraints, provided that certain nonsingularity conditions hold. However, usually this approach cannot keep the csp from the original polynomial optimization problem. Moreover, when the nonsingularity condition fails, LMEs do not exist. In contrast, one can try to find CS-LME instead by adding some new variables. In the above, we demonstrate how to find CS-LMEs for the two-block cases. In the next subsection, we provide a systematic way to construct CS-LMEs for an arbitrary number of blocks.

3.3. Correlatively sparse LME: multi-blocks. We introduce how to construct CS-LMEs for an arbitrary number of blocks in this subsection. Hereinafter, we assume that the correlative sparsity pattern $(\mathcal{I}_1, \ldots, \mathcal{I}_s)$ satisfies the RIP. Without loss of generality, we also assume the following conditions hold:

1. $\mathcal{I}_i$ is not included in $\mathcal{I}_j$ for any two distinct $i, j \in [s]$;
2. $\mathcal{I}_{i+1} \cap \cup_{j=1}^{i} \mathcal{I}_j \neq \emptyset$ for any $i \in [s-1]$.

We remark that under the RIP condition, the second condition always holds unless there exists a proper subset S of $[s]$ such that $(\cup_{i \in S} \mathcal{I}_i) \cap (\cup_{i \notin S} \mathcal{I}_i) = \emptyset$, for which we can solve the polynomial optimization problems for variables within S and outside of S separately.

To construct CS-LME coherent with the csp $(\mathcal{I}_1, \ldots, \mathcal{I}_s)$, we first build a directed tree with nodes corresponding to the elements in the collection $\{\mathcal{I}_1, \ldots, \mathcal{I}_s\}$.

Algorithm 3.1 Clique Tree Construction

Input: $(\mathcal{I}_1, \ldots, \mathcal{I}_s)$ satisfying the RIP.

$V = \{1, \ldots, s\}$ and $A = \emptyset$.
for $i = 1, \ldots, s-1$ **do**
  if $\mathcal{I}_{i+1} \bigcap \bigcup_{j=1}^{i} \mathcal{I}_j \neq \emptyset$ **then**
    Find the largest $t \leq i$ such that $\mathcal{I}_{i+1} \bigcap \bigcup_{j=1}^{i} \mathcal{I}_j \subseteq \mathcal{I}_t$.
    $A = A \bigcup \{(i+1, t)\}$.
  end if
end for

Output: $G(V, A)$

The *correlative sparsity pattern (csp) graph* associated with (1.2) is the undirected graph $G^{csp} = G(W, E)$, with nodes $W = [n]$ and edges E satisfying $\{k_1, k_2\} \in E$ if there exists $i \in [s]$ such that $k_1 \in \mathcal{I}_i$ and $k_2 \in \mathcal{I}_i$. Since $(\mathcal{I}_1, \ldots, \mathcal{I}_s)$ satisfies the RIP, the corresponding csp graph G^{csp} is chordal[2] and $\{\mathcal{I}_1, \ldots, \mathcal{I}_s\}$ is the list of maximal cliques of G^{csp}, because we assumed that $\mathcal{I}_i$ is not contained in $\mathcal{I}_j$ for any distinct $i, j \in [s]$. A *clique tree* of the graph G^{csp} is a tree on the set $V = [s]$ such that for every pair of distinct nodes $i, j \in [s]$, we have $\mathcal{I}_i \cap \mathcal{I}_j \subseteq \mathcal{I}_k$ for any $k \in [s]$ on the path connecting i and j in the tree. Clique tree exists because G^{csp} is chordal; see [1, Theorem 3.1]. The output $G(V, A)$ of Algorithm 3.1 is a directed tree whose underlying undirected graph is a clique tree of the graph G^{csp}. This follows from [1, Theorem 3.4]. The directions indicate the "parent-child" relation between cliques on the tree. We refer to [1] for more details on chordal graphs and clique trees.

Given the clique tree $G(V, A)$ produced by Algorithm 3.1, for each $i \in [s]$, we denote the indices of children of the node i by

$$\mathcal{D}_i := \{t : (t, i) \in A\}, \tag{3.20}$$

[2] A graph is chordal if all its cycles of length at least four have an edge that joins two nonconsecutive nodes.

and the index of the parent of node i by

$$\mathcal{A}_i := \{t : (i,t) \in A\}. \tag{3.21}$$

For each $i \in \{2, \ldots, s\}$, $\mathcal{D}_i$ can be empty sets and $\mathcal{A}_i$ contains exactly one element. When $(i,t) \in A$, we let

$$\mathcal{C}_{i,t} := \mathcal{I}_i \bigcap \mathcal{I}_t \tag{3.22}$$

be the indices of all variables shared by blocks i and t. Then, we introduce a group of auxiliary variables:

$$\{\nu_{i,t,k} : (i,t) \in A, k \in \mathcal{C}_{i,t}\}. \tag{3.23}$$

In other words, for each arc $(i,t) \in A$, we need the same number of auxiliary variables as the number of variables shared by the block i and block t. For every $i \in [s]$, define

$$\mathcal{J}_i := \{(i,t,k) : t \in \mathcal{A}_i,\ k \in \mathcal{C}_{i,t}\} \cup \{(t,i,k) : t \in \mathcal{D}_i,\ k \in \mathcal{C}_{t,i}\}\,, \tag{3.24}$$

and (recall that the vector $e_k^{(i)}$ is defined in (2.5))

$$\nu^{(i)} := -\sum_{t \in \mathcal{A}_i} \sum_{k \in \mathcal{C}_{i,t}} \nu_{i,t,k} e_k^{(i)} + \sum_{t \in \mathcal{D}_i} \sum_{k \in \mathcal{C}_{t,i}} \nu_{t,i,k} e_k^{(i)} \in \mathbb{R}^{n_i}. \tag{3.25}$$

Clearly, the vector $\nu^{(i)}$ only depends on variables in the group (3.23) indexed by $\mathcal{J}_i$ for each $i \in [s]$. We illustrate how to construct new variables in the following example.

Example 3.6. Consider the following csp pattern:

$$\begin{array}{c} \mathcal{I}_1 = \{1,2,3,4\},\ \ \mathcal{I}_2 = \{1,2,5,6\},\ \ \mathcal{I}_3 = \{1,2,7,8\}, \\ \mathcal{I}_4 = \{1,2,9,10\},\ \ \mathcal{I}_5 = \{1,2,11,12\}. \end{array} \tag{3.26}$$

Then the set of edges A in the clique tree $G(V,A)$ produced by Algorithm 3.1 is

$$A = \{(2,1),\ (3,2),\ (4,3),\ (5,4)\},$$

and $\mathcal{D}_i = \{i+1\}$ for each $i = 1, \ldots, 4$, $\mathcal{A}_i = \{i-1\}$ for each $i = 2, \ldots, 5$. Thus

$$\begin{aligned} \mathcal{J}_1 &= \{(2,1,1),\ (2,1,2)\}, \\ \mathcal{J}_2 &= \{(2,1,1),\ (2,1,2)\} \cup \{(3,2,1),\ (3,2,2)\}, \\ \mathcal{J}_3 &= \{(3,2,1),\ (3,2,2)\} \cup \{(4,3,1),\ (4,3,2)\}, \\ \mathcal{J}_4 &= \{(4,3,1),\ (4,3,2)\} \cup \{(5,4,1),\ (5,4,2)\}, \\ \mathcal{J}_5 &= \{(5,4,1),\ (5,4,2)\}. \end{aligned}$$

An illustration of the directed tree obtained from Algorithm 3.1 and auxiliary variables are given in Figure 1. For this clique tree, we have $|\mathcal{J}_1| = |\mathcal{J}_5| = 2$ and $|\mathcal{J}_2| = |\mathcal{J}_3| = |\mathcal{J}_4| = 4$.

With new variables $\nu_{i,t,k}$ and vectors $\nu^{(i)}$ given by (3.25), we rewrite the KKT system (3.2). For each $i \in [s]$, consider the following system on $(x^{(i)}, \lambda^{(i)}, \nu^{(i)}) \in \mathbb{R}^{n_i+m_i+\ell_i+|\mathcal{J}_i|}$:

$$\left\{ \begin{array}{c} \nabla_i f_i(x^{(i)}) + \nu^{(i)} = \sum_{j=1}^{m_i} \lambda_j^{(i)} \nabla_i g_j^{(i)}(x^{(i)}) + \sum_{j=1}^{\ell_i} \lambda_{m_i+j}^{(i)} \nabla_i h_j^{(i)}(x^{(i)}), \\ h^{(i)}(x^{(i)}) = 0, \\ 0 \le \lambda_{1:m_i}^{(i)} \perp g^{(i)}(x^{(i)}) \ge 0. \end{array} \right. \tag{3.27}$$

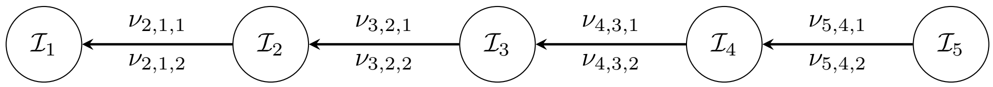

FIG. 1. *Clique tree returned by Algorithm 3.1 and auxiliary variables for the csp pattern (3.26).*

PROPOSITION 3.7. *Let $x := (x_1, \ldots, x_n) \in \mathbb{R}^n$ and $\lambda := (\lambda^{(1)}, \ldots, \lambda^{(s)}) \in \mathbb{R}^{m+\ell}$. The pair (x, λ) is a KKT pair of (1.2) if and only if there exists a group of auxiliary variables $\{\nu_{i,t,k} : (i,t) \in A, k \in \mathcal{C}_{i,t}\}$ such that (3.27) holds for all $i \in [s]$.*

Proof. By lifting all the vectors into $\mathbb{R}^n$ (i.e., filling in 0 to the coordinates that are not in $\mathcal{I}_i$), we can rewrite the first equation in (3.27) as

$$\nabla f_i(x) + \hat{\nu}^{(i)} = \sum_{j=1}^{m_i} \lambda_j^{(i)} \nabla g_j^{(i)}(x) + \sum_{j=1}^{\ell_i} \lambda_{m_i+j}^{(i)} \nabla h_j^{(i)}(x), \tag{3.28}$$

where $\hat{\nu}^{(i)} \in \mathbb{R}^n$ is obtained by lifting $\nu^{(i)}$ into $\mathbb{R}^n$:

$$\hat{\nu}^{(i)} := -\sum_{t\in\mathcal{A}_i} \sum_{k\in\mathcal{C}_{i,t}} \nu_{i,t,k} e_k + \sum_{t\in\mathcal{D}_i} \sum_{k\in\mathcal{C}_{t,i}} \nu_{t,i,k} e_k.$$

If there exists (x, λ) and $\{\nu_{i,t,k} : (i,t) \in A, k \in \mathcal{C}_{i,t}\}$ such that (3.27) holds for all $i \in [s]$, then

$$\begin{aligned}
\nabla f(x) &= \sum_{i=1}^{s} \nabla f_i(x) \\
&\overset{(3.28)}{=} \sum_{i=1}^{s} \left(\sum_{j=1}^{m_i} \lambda_j^{(i)} \nabla g_j^{(i)}(x) + \sum_{j=1}^{\ell_i} \lambda_{m_i+j}^{(i)} \nabla h_j^{(i)}(x) - \hat{\nu}^{(i)} \right) \\
&= \sum_{i=1}^{s} \left(\sum_{j=1}^{m_i} \lambda_j^{(i)} \nabla g_j^{(i)}(x) + \sum_{j=1}^{\ell_i} \lambda_{m_i+j}^{(i)} \nabla h_j^{(i)}(x) \right) - \sum_{i=1}^{s} \hat{\nu}^{(i)} \\
&= \sum_{i=1}^{s} \left(\sum_{j=1}^{m_i} \lambda_j^{(i)} \nabla g_j^{(i)}(x) + \sum_{j=1}^{\ell_i} \lambda_{m_i+j}^{(i)} \nabla h_j^{(i)}(x) \right).
\end{aligned}$$

Therefore (x, λ) is a KKT pair of (1.2). In the following, we show the other direction.

Let (x, λ) be a KKT pair of (1.2). For each fixed $k \in [n]$, denote

$$\mathcal{P}_k := \{(i,t) \in A : k \in \mathcal{C}_{i,t}\}, \quad \mathcal{Q}_k := \{i : k \in \mathcal{I}_i\}.$$

In other words, $\mathcal{Q}_k$ corresponds to the set of cliques that contain k and $G(\mathcal{Q}_k, \mathcal{P}_k)$ is the subgraph of $G(V, A)$ induced by the nodes $\mathcal{Q}_k$. Then by [1, Theorem 3.2], for each $k \in [n]$, the underlying undirected graph of $G(\mathcal{Q}_k, \mathcal{P}_k)$ is a tree.

This allows us to deduce the solvability of the following system of linear equations for each fixed $k \in [n]$:

$$\begin{aligned}
&\sum_{t\in\mathcal{P}_k(i)} \nu_{i,t,k} - \sum_{t\in\mathcal{P}'_k(i)} \nu_{t,i,k} \\
&\quad = \frac{\partial f_i}{\partial x_k}(x) - \left(\sum_{j=1}^{m_i} \lambda_j^{(i)} \frac{\partial g_j^{(i)}}{\partial x_k}(x) + \sum_{j=1}^{\ell_i} \lambda_{m_i+j}^{(i)} \frac{\partial h_j^{(i)}}{\partial x_k}(x) \right), \quad \forall i \in \mathcal{Q}_k.
\end{aligned} \tag{3.29}$$

In the above,

$$\mathcal{P}_k(i) := \{t : (i,t) \in \mathcal{P}_k\}, \quad \mathcal{P}'_k(i) := \{t : (t,i) \in \mathcal{P}_k\}. \tag{3.30}$$

Indeed, the linear system (3.29) can be written as

$$Bv = b, \tag{3.31}$$

where $B \in \mathbb{R}^{|\mathcal{Q}_k|\times|\mathcal{P}_k|}$ is the incidence matrix of $(\mathcal{Q}_k, \mathcal{P}_k)$ and $b \in \mathbb{R}^{|\mathcal{Q}_k|}$ is a vector satisfying $\mathbf{1}^\top b = 0$. Since the underlying undirected graph of $G(\mathcal{Q}_k, \mathcal{P}_k)$ is a tree, we have $\operatorname{rank}(B) = |\mathcal{P}_k| = |\mathcal{Q}_k| - 1$ and $\mathbf{1}^\top B = 0$. Therefore (3.31), and thus (3.29) for each $k \in [n]$, are solvable. In other words, there exist $\{\nu_{i,t,k} : (i,t) \in A,\ k \in \mathcal{C}_{i,t}\}$ such that (3.29) holds for all $k \in [n]$. So the following equations hold at (x, λ):

$$\begin{aligned} &\sum_{k\in\mathcal{I}_i}\sum_{t\in\mathcal{P}_k(i)} \nu_{i,t,k} e_k - \sum_{k\in\mathcal{I}_i}\sum_{t\in\mathcal{P}'_k(i)} \nu_{t,i,k} e_k \\ = &\sum_{k\in\mathcal{I}_i}\left(\frac{\partial f_i}{\partial x_k}(x) - \sum_{j=1}^{m_i}\lambda_j^{(i)}\frac{\partial g_j^{(i)}}{\partial x_k}(x) - \sum_{j=1}^{\ell_i}\lambda_{m_i+j}^{(i)}\frac{\partial h_j^{(i)}}{\partial x_k}(x)\right)e_k, \quad \forall i \in [s]. \end{aligned} \tag{3.32}$$

Note that for any $k \notin \mathcal{I}_i$, we have

$$\frac{\partial f_i}{\partial x_k}(x) \equiv \frac{\partial g_1^{(i)}}{\partial x_k}(x) \equiv \cdots \equiv \frac{\partial g_{m_i}^{(i)}}{\partial x_k}(x) \equiv \frac{\partial h_1^{(i)}}{\partial x_k}(x) \equiv \cdots \equiv \frac{\partial h_{\ell_i}^{(i)}}{\partial x_k}(x) \equiv 0.$$

Therefore, (3.32) yields that for each $i \in [s]$,

$$\begin{aligned} &\sum_{k\in\mathcal{I}_i}\left(\sum_{t\in\mathcal{P}_k(i)}\nu_{i,t,k} - \sum_{t\in\mathcal{P}'_k(i)}\nu_{t,i,k}\right)e_k \\ = &\sum_{k\in[n]}\left(\frac{\partial f_i}{\partial x_k}(x) - \sum_{j=1}^{m_i}\lambda_j^{(i)}\frac{\partial g_j^{(i)}}{\partial x_k}(x) - \sum_{j=1}^{\ell_i}\lambda_{m_i+j}^{(i)}\frac{\partial h_j^{(i)}}{\partial x_k}(x)\right)e_k \\ = &\nabla f_i(x) - \sum_{j=1}^{m_i}\lambda_j^{(i)}\nabla g_j^{(i)}(x) - \sum_{j=1}^{\ell_i}\lambda_{m_i+j}^{(i)}\nabla h_j^{(i)}(x). \end{aligned} \tag{3.33}$$

In light of (3.20)-(3.21), for each fixed $i \in [s]$ we have

$$\{(t,k) : k \in \mathcal{I}_i, t \in \mathcal{P}_k(i)\} = \{(t,k) : t \in \mathcal{A}_i, k \in \mathcal{C}_{i,t}\}.$$

We then obtain

$$\begin{aligned} &\sum_{k\in\mathcal{I}_i}\left(\sum_{t\in\mathcal{P}_k(i)}\nu_{i,t,k} - \sum_{t\in\mathcal{P}'_k(i)}\nu_{t,i,k}\right)e_k \\ = &\sum_{t\in\mathcal{A}_i}\sum_{k\in\mathcal{C}_{i,t}}\nu_{i,t,k}e_k - \sum_{t\in\mathcal{D}_i}\sum_{k\in\mathcal{C}_{t,i}}\nu_{t,i,k}e_k \\ = &-\hat{\nu}^{(i)}. \end{aligned}$$

Therefore, (3.28) holds, and the first equation in (3.27) is satisfied. ∎

Remark 3.8. In Algorithm 3.1, even if we replace line 4 by

$$\text{Find an arbitrary } t \le i \text{ such that } \mathcal{I}_{i+1} \bigcap \bigcup_{j=1}^{i} \mathcal{I}_j \subseteq \mathcal{I}_t, \tag{3.34}$$

the resulting tree is still a clique tree, and the induced subtree property still holds for $G(V, A)$. Hence, Proposition 3.7, as well as all the results that will follow, still hold if line 4 of Algorithm 3.1 is replaced with (3.34). This is because the only key property of $G(V, A)$ needed in the proof of Proposition 3.7 is the *induced subtree property* (see [1, Theorem 3.2]) satisfied by the clique tree. However, using an arbitrary t as in (3.34) may create a large number of children for some nodes (see Example 3.9 below), which will increase the number of variables in $\nu^{(i)}$ (hence the size of blocks and the computational cost). In other words, one would prefer a tree with a large depth and small breadth. That is why we propose to choose the largest t in Algorithm 3.1.

Example 3.9. Consider the csp pattern (3.26) again. If we use (3.34) to replace line 4 in Algorithm 3.1, then another possible directed clique tree and auxiliary variables is shown in Figure 2. For this clique tree, we have $|\mathcal{J}_1| = 8$, $|\mathcal{J}_2| = |\mathcal{J}_3| = |\mathcal{J}_4| = |\mathcal{J}_5| = 2$.

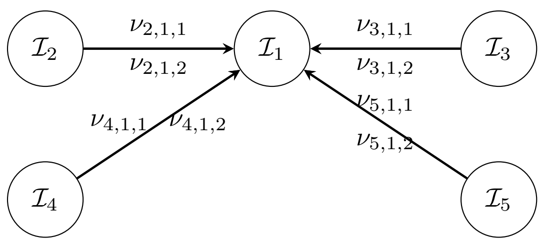

FIG. 2. *Another possible clique tree and auxiliary variables for the csp pattern* (3.26).

Under Assumption 1, (3.27) implies that the ith group of Lagrange multipliers can be expressed by a tuple of polynomials which only depends on variables indexed by $\mathcal{I}_i$ and $\mathcal{J}_i$, say, $x^{(i)}$ and $\nu^{(i)}$ (by abuse of notation, here $\nu^{(i)}$ means the tuple of all variables involved in the vector $\nu^{(i)}$). We let

$$z^{(i)} := (x^{(i)}, \nu^{(i)}), \quad F^{(i)}(z^{(i)}) := \nabla_i f_i(x^{(i)}) + \nu^{(i)}. \tag{3.35}$$

THEOREM 3.10. *Under Assumption 1, a vector $x \in \mathbb{R}^n$ is a KKT point of* (1.2) *if and only if the following system* (3.36) *holds for each $i \in [s]$:*

$$\begin{cases} F^{(i)}(z^{(i)}) = \sum_{j=1}^{m_i} p_j^{(i)}\left(z^{(i)}\right) \nabla_i g_j^{(i)}(x^{(i)}) + \sum_{j=1}^{\ell_i} p_{m_i+j}^{(i)}\left(z^{(i)}\right) \nabla_i h_j^{(i)}(x^{(i)}), \\ \qquad 0 \le p_{1:m_i}^{(i)}(z^{(i)}) \perp g^{(i)}(x^{(i)}) \ge 0, \ h^{(i)}(x^{(i)}) \ge 0, \end{cases} \tag{3.36}$$

where

$$p^{(i)}(z^{(i)}) := L^{(i)}(x^{(i)}) \cdot F^{(i)}(z^{(i)}), \tag{3.37}$$

and $z^{(i)}$ and $F^{(i)}$ are defined in (3.35).

Proof. Recall the matrix of polynomials $G^{(i)}(x^{(i)})$ defined in (3.1). The system (3.27) is equivalent to

$$\begin{cases} G^{(i)}(x^{(i)})\lambda^{(i)} = \left[\, F^{(i)}(z^{(i)})^\top \ 0 \ \cdots \ 0 \,\right]^\top, \\ \lambda_1^{(i)}, \ldots, \lambda_{m_i}^{(i)} \ge 0, \ g_1^{(i)}, \ldots, g_{m_i}^{(i)} \ge 0, \ h_1^{(i)}, \ldots, h_{\ell_i}^{(i)} \ge 0, \end{cases} \tag{3.38}$$

By Assumption 1, the first equation in (3.38) holds if and only if

$$\lambda^{(i)} = L^{(i)}(x^{(i)}) \cdot F^{(i)}(z^{(i)}).$$

Thus, it remains to replace $\lambda^{(i)}$ with $p^{(i)}(z^{(i)})$ in (3.27) and apply Proposition 3.7. □

Remark 3.11. For the polynomial optimization problem (1.2) with general csp $(\mathcal{I}_1, \ldots, \mathcal{I}_s)$, we call the vector of polynomials $p^{(i)}(z^{(i)})$ defined in (3.37) the CS-LMEs of $\lambda^{(i)}$. By Proposition 3.7 and Theorem 3.10, CS-LMEs exist when Assumption 1 is satisfied.

3.4. A CS-LME reformulation. In the rest of the paper, we give a CS-LME reformulation for the polynomial optimization problem (1.2) under Assumption 1. For each $i \in [s]$, denote

$$\phi^{(i)}(z^{(i)}) := \left[\begin{array}{c} F^{(i)}(z^{(i)}) - \sum_{j=1}^{m_i} p_j^{(i)}(z^{(i)}) \nabla_i g_j^{(i)}(x^{(i)}) - \sum_{j=1}^{\ell_i} p_{m_i+j}^{(i)}(z^{(i)}) \nabla_i h_j^{(i)}(x^{(i)}) \\ h^{(i)}(x^{(i)}) \\ p_{1:m}^{(i)}(z^{(i)}) \circ g^{(i)}(x^{(i)}) \end{array} \right],$$

and

$$\psi^{(i)}(z^{(i)}) := \left[\begin{array}{c} p_{1:m}^{(i)}(z^{(i)}) \\ g^{(i)}(x^{(i)}) \end{array} \right].$$

Here, the polynomial $p_j^{(i)} \in \mathbb{R}[z^{(i)}]$ is the jth entry of the CS-LME $p^{(i)}$ defined in (3.37). Based on Theorem 3.10, we propose the following CS-LME typed reformulation of (1.2):

$$\text{(3.39)} \qquad \left\{ \begin{array}{lll} f_c := & \min\limits_{z^{(1)},\ldots,z^{(s)}} & f_1(x^{(1)}) + \cdots + f_s(x^{(s)}) \\ & \text{s.t.} & \psi^{(i)}(z^{(i)}) \geq 0, \;\; i \in [s] \\ & & \phi^{(i)}(z^{(i)}) = 0, \;\; i \in [s] \end{array} \right.$$

The previous reformulation (3.17) for the case $s = 2$ is a special case of (3.39). If we let

$$\text{(3.40)} \qquad \hat{\mathcal{I}}_i := \mathcal{I}_i \bigcup \mathcal{J}_i,$$

then (3.39) has the csp $(\hat{\mathcal{I}}_1, \ldots, \hat{\mathcal{I}}_s)$. Suppose the global minimum $f_{\min}$ of (1.2) is attained at some KKT point, then at least one minimizer of (1.2) is feasible for (3.39), thus $f_{\min} \geq f_c$. Since the feasible set of (3.39) is contained in the feasible set of (1.2), we have $f_{\min} \leq f_c$. So, we conclude the following from the statement above:

THEOREM 3.12. *If the minimum $f_{\min}$ of (1.2) is attained at a KKT point, then the minimal value (1.2) and (3.39) are identical, i.e., $f_{\min} = f_c$.*

Remark 3.13. Suppose the minimum value $f_{\min}$ is attainable. If the nonsingularity condition holds for (1.2), then $f_{\min}$ is attained at KKT points, since the nonsingularity implies the linear independence constraint qualification conditions (LICQ) hold on $\mathbb{C}^n$. However, this is not necessarily true if we replace the nonsingularity condition of (g, h) by that of every $(g^{(i)}, h^{(i)})$, i.e., Assumption 1, since Assumption 1 does not guarantee the LICQ to hold at every feasible point. For such cases, f_c may or may not equal $f_{\min}$. Nevertheless, it does not mean the KKT conditions must fail at minimizers of (1.2) if the nonsingularity condition does not hold. Indeed, it may happen that the constraining tuple is singular, but the LICQ condition holds at a minimizer, thus $f_c = f_{\min}$; see Example 5.1(ii), Example 5.5 and Example 5.7.

4. Correlatively sparse LME based SOS relaxation. This section studies the correlatively sparse SOS relaxations for solving the CS-LME reformulation (3.39).

4.1. RIP of the CS-LME reformulation. First, we establish the RIP for (3.39). Recall that for each $i \in [s]$, the set of indices of variables $\hat{\mathcal{I}}_i$ is given in (3.40).

LEMMA 4.1. *The csp* $(\hat{\mathcal{I}}_1, \ldots, \hat{\mathcal{I}}_s)$ *satisfies the RIP in Definition 2.2.*

Proof. Note that

$$\begin{aligned}
&\mathcal{J}_{i+1} \bigcap \bigcup_{j=1}^{i} \mathcal{J}_j \\
= &\Big(\{(i+1, t, k) : t \in \mathcal{A}_{i+1}, k \in \mathcal{C}_{i+1,t}\} \bigcup \{(t, i+1, k) : t \in \mathcal{D}_{i+1}, k \in \mathcal{C}_{t,i+1}\} \Big) \\
&\qquad \bigcap \bigcup_{j=1}^{i} \Big(\{(j, t, k) : t \in \mathcal{A}_j, k \in \mathcal{C}_{j,t}\} \bigcup \{(t, j, k) : t \in \mathcal{D}_j, k \in \mathcal{C}_{t,j}\} \Big).
\end{aligned}$$

Since $t \in \mathcal{D}_{i+1}$ implies $t > i+1$, and $t \in \mathcal{A}_j$ implies $t < j$, we have

$$\begin{aligned}
\mathcal{J}_{i+1} \bigcap \bigcup_{j=1}^{i} \mathcal{J}_j = &\{(i+1, t, k) : t \in \mathcal{A}_{i+1}, k \in \mathcal{C}_{i+1,t}\} \bigcap \\
&\bigcup_{j=1}^{i} \Big(\{(j, t, k) : t \in \mathcal{A}_j, k \in \mathcal{C}_{j,t}\} \bigcup \{(t, j, k) : t \in \mathcal{D}_j,\ k \in \mathcal{C}_{t,j}\} \Big) \\
\subseteq &\{(i+1, t, k) : t \in \mathcal{A}_{i+1}, k \in \mathcal{C}_{i+1,t}\}.
\end{aligned}$$

Let $\mathcal{A}_{i+1} = \{t\}$ for some $t \in [s]$. Then $i+1 \in \mathcal{D}_t$ and so

$$\begin{aligned}
\mathcal{J}_t &= \{(t, i, k) : i \in \mathcal{A}_t, k \in \mathcal{C}_{t,i}\} \cup \{(i, t, k) : i \in \mathcal{D}_t, k \in \mathcal{C}_{i,t}\} \\
&\supseteq \{(i+1, t, k) : k \in \mathcal{C}_{i+1,t}\}.
\end{aligned}$$

Note that $\mathcal{I}_i$ is the set of indices of variables $x^{(i)}$ and $\mathcal{J}_i$ is the set of indices of the auxiliary variables $\nu^{(i)}$. Hence $\mathcal{I}_i \cap \mathcal{J}_j = \emptyset$ for each pair of $i, j \in [s]$. In particular,

$$\begin{aligned}
&\hat{\mathcal{I}}_{i+1} \bigcap \bigcup_{j=1}^{i} \hat{\mathcal{I}}_j \\
= &\Big\{ \mathcal{I}_{i+1} \bigcup \mathcal{J}_{i+1} \Big\} \bigcap \left\{ \bigcup_{j=1}^{i} \Big(\mathcal{I}_j \bigcup \mathcal{J}_j \Big) \right\} \\
= &\left\{ \mathcal{I}_{i+1} \bigcap \left\{ \bigcup_{j=1}^{i} \Big(\mathcal{I}_j \bigcup \mathcal{J}_j \Big) \right\} \right\} \bigcup \left\{ \mathcal{J}_{i+1} \bigcap \left\{ \bigcup_{j=1}^{i} \Big(\mathcal{I}_j \bigcup \mathcal{J}_j \Big) \right\} \right\} \\
= &\left\{ \mathcal{I}_{i+1} \bigcap \bigcup_{j=1}^{i} \mathcal{I}_j \right\} \bigcup \left\{ \mathcal{J}_{i+1} \bigcap \bigcup_{j=1}^{i} \mathcal{J}_j \right\}.
\end{aligned}$$

Therefore, we have

$$\hat{\mathcal{I}}_{i+1} \bigcap \bigcup_{j=1}^{i} \hat{\mathcal{I}}_j = \left\{ \mathcal{I}_{i+1} \bigcap \bigcup_{j=1}^{i} \mathcal{I}_j \right\} \bigcup \left\{ \mathcal{J}_{i+1} \bigcap \bigcup_{j=1}^{i} \mathcal{J}_j \right\} \subseteq \mathcal{I}_t \cup \mathcal{J}_t = \hat{\mathcal{I}}_t. \qquad \square$$

4.2. Convergence of the CS-LME based SOS relaxation. For the polynomial optimization problem (3.39) with the csp $(\hat{\mathcal{I}}_1, \ldots, \hat{\mathcal{I}}_s)$, the dth order correlatively sparse SOS relaxation is

$$(4.1) \qquad \begin{cases} \vartheta_d := & \max \quad \gamma \\ & \text{s.t.} \quad \sum_{i=1}^{s} f_i - \gamma \in \sum_{i=1}^{s} \mathrm{IQ}_{\hat{\mathcal{I}}_i}\left(\phi^{(i)}, \psi^{(i)}\right)_{2d}. \end{cases}$$

Note that for each $i \in [s]$, $h^{(i)}$ is contained in $\phi^{(i)}$, $g^{(i)}$ is contained in $\psi^{(i)}$, and $\mathcal{I}_i \subseteq \hat{\mathcal{I}}_i$. It follows that

$$\sum_{i=1}^{s} \mathrm{IQ}_{\hat{\mathcal{I}}_i}\left(h^{(i)}, g^{(i)}\right)_{2d} \subseteq \sum_{i=1}^{s} \mathrm{IQ}_{\hat{\mathcal{I}}_i}\left(\phi^{(i)}, \psi^{(i)}\right)_{2d}.$$

Therefore, (4.1) is a tighter relaxation than (2.8). In particular, we have

$$(4.2) \qquad \vartheta_d \geq \rho_d, \ \ \forall d \geq d_0.$$

THEOREM 4.2. *Assume that:*
1. *at least one minimizer of* (1.2) *is a KKT point, and*
2. *for each* $i \in [s]$, $\mathrm{IQ}_{\mathcal{I}_i}\left(h^{(i)}, g^{(i)}\right)$ *is archimedean.*

Then

$$(4.3) \qquad \lim_{d \to +\infty} \vartheta_d = f_{\min}.$$

Proof. By the definition of CS-SOS relaxation, we have

$$\vartheta_d \leq f_c, \ \ \forall d \in \mathbb{N}.$$

The first condition, together with Theorem 3.10, implies that $f_c = f_{\min}$. Then we have $\vartheta_d \leq f_{\min}$, and the convergence follows directly by (2.10) and (4.2). ∎

Remark 4.3. In Theorem 4.2, if we substitute the condition that $\mathrm{IQ}_{\mathcal{I}_i}\left(h^{(i)}, g^{(i)}\right)$ is archimedean by the archimedeanness of $\mathrm{IQ}_{\hat{\mathcal{I}}_i}\left(\phi^{(i)}, \psi^{(i)}\right)$, then the conclusion still holds. However, $\mathrm{IQ}_{\hat{\mathcal{I}}_i}\left(\phi^{(i)}, \psi^{(i)}\right)$ is not archimedean in general, even if $\mathrm{IQ}_{\mathcal{I}_i}\left(h^{(i)}, g^{(i)}\right)$ is archimedean. To see this, consider the CS-LME reformulation (3.14) for the optimization problem in Example 2.5. In (3.14), tuples $h^{(1)}, g^{(1)}, h^{(2)}, g^{(2)}$ are given by (3.5), and it is clear that both $\mathrm{IQ}_{\mathcal{I}_1}\left(h^{(1)}, g^{(1)}\right)$ and $\mathrm{IQ}_{\mathcal{I}_2}\left(h^{(2)}, g^{(2)}\right)$ are archimedean. Moreover, $(\phi^{(1)}, \psi^{(1)})$ corresponds to the first two constraints in (3.14), and $(\phi^{(2)}, \psi^{(2)})$ is given by the last two constraints in (3.14). For any fixed $\nu \in \mathbb{R}$, consider the following polynomial optimization problem in variables (x_1, x_2):

$$(4.4) \qquad \begin{cases} \min & f_1(x_1, x_2) + \nu x_2 \\ \text{s.t.} & 1 - x_1^2 - x_2^2 \geq 0. \end{cases}$$

Then one may check that $(x_1, x_2, \nu) \in \left\{z^{(1)} \in \mathbb{R}^3 : \phi^{(1)}(z^{(1)}) = 0, \psi^{(1)}(z^{(1)}) \geq 0\right\}$ if and only if (x_1, x_2) is a KKT point for (4.4). Since (4.4) has a compact feasible set, and the constraint qualification condition holds at all feasible points, (4.4) has a KKT point for any $\nu \in \mathbb{R}$. This implies that the semi-algebraic set

$$\left\{z^{(1)} \in \mathbb{R}^3 : \phi^{(1)}(z^{(1)}) = 0, \psi^{(1)}(z^{(1)}) \geq 0\right\}$$

is unbounded, and thus $\mathrm{IQ}_{\hat{\mathcal{I}}_1}\left(\phi^{(1)}, \psi^{(1)}\right)$ is not archimedean. Similarly, one can also show that $\mathrm{IQ}_{\hat{\mathcal{I}}_2}\left(\phi^{(2)}, \psi^{(2)}\right)$ is not archimedean neither.

Remark 4.4. The archimedean condition of $\mathrm{IQ}_{\mathcal{I}_i}\left(h^{(i)}, g^{(i)}\right)$ for each $i \in [s]$ is also required in Theorem 2.3 to ensure the convergence of the CS-SOS relaxation. We wish to point out that this archimedean condition is not required for obtaining the CS-LMEs (3.37) and the CS-LME reformulation (3.39). There may exist polynomial optimization problems with compact feasible sets, for which, however, $\mathrm{IQ}_{\mathcal{I}_i}\left(h^{(i)}, g^{(i)}\right)$ is not archimedean for some $i \in [s]$ (e.g., Example 2.5 and Example 5.3). For such cases, one may add redundant constraints to $g^{(i)}$ to obtain the archimedeanness. Such a redundant constraint can either be a replication of existing constraints, or be the ball constraint as $M - \|x^{(i)}\|^2 \geq 0$ if an *a priori* bound M is known.[3] However, adding redundant constraints is inconvenient and usually unnecessary in practice. Indeed, even if the archimedean conditions fail to hold (or further, if the feasible set of (1.2) is unbounded), we can still formulate and solve the CS-LME reformulations with CS-SOS relaxations. In practical computation, finite convergence is observed numerically with a low relaxation order for solving CS-LME reformulations, regardless of whether the archimedean condition for $\mathrm{IQ}_{\mathcal{I}_i}\left(h^{(i)}, g^{(i)}\right)$ holds or not. We refer to Section 5 for examples where the archimedean condition is not satisfied, while our approach can still find global minimum successfully.

4.3. Comparison of the SDP problem scale. In this section, we compare the scale of the corresponding SDP problems in different relaxation approaches. We assume that the functions $f_1 \in \mathbb{R}[x^{(1)}], \dots, f_s \in \mathbb{R}[x^s]$ are all dense polynomials and both LMEs and CS-LMEs exist for (1.2). For the convenience of reference, we nominate the four approaches for solving (1.2) as follows:

- (SOS): Applying the dense SOS relaxation to (1.2);
- (CS-SOS): Applying the CS-SOS relaxation to (1.2);
- (LME): Applying the CS-SOS relaxation to the LME reformulation (2.16);
- (CS-LME): Applying the CS-SOS relaxation to the CS-LME reformulation (3.39).

We first consider the two-block case. Denote by $k := |\mathcal{C}_{1,2}|$ the number of overlapping elements in $\mathcal{I}_1$ and $\mathcal{I}_2$. Then $|\mathcal{I}_1 \cup \mathcal{I}_2| = n_1 + n_2 - k$ is the total number of variables. The CS-LME reformulation (3.17) has the csp $(\hat{\mathcal{I}}_1, \hat{\mathcal{I}}_2)$ such that $|\hat{\mathcal{I}}_1| = n_1 + k$ and $|\hat{\mathcal{I}}_2| = n_2 + k$. In Table 1, we compare the maximal size of the positive semidefinite (PSD) matrices appearing in the SDP formulation of the four relaxation methods. In Table 2, we display the values of the binomial numbers in Table 1 for some examples of n_1, n_2, k, d.

From Table 1 and Table 2, we conclude that for the same order of relaxation, the smallest scale SDP problem is given by CS-SOS. On the other hand, CS-SOS may need higher relaxation order d to converge than the other three methods. For the case when $s = 2$, the complexity growth of the LME approach is the same as that of the SOS approach. Thus, despite its potentially faster convergence speed, the LME approach

[3] It is important to note that there are two ways to replicate existing constraints. For the constraint $g_j \in \mathbb{R}[x^{(i)}]$ that is not assigned to $g^{(i)}$, we may add its replication to $g^{(i)}$ and obtain a new constraining tuple $\hat{g}^{(i)}$, then consider the KKT system and construct CS-LMEs for $\hat{g}^{(i)}$, as long as the new constraining tuple $\hat{g}^{(i)}$ is also nonsingular. On the other hand, one may add g_j to $\psi^{(i)}$ in the CS-LME reformulation. These two ways produce different CS-LME reformulations with identical optimal values, since the former may get different CS-LMEs from the original problem. However, if we add a redundant ball constraint $M - \|x^{(i)}\|^2 \geq 0$ which can never be active (e.g., let $M := n_i \cdot \hat{M}$ with $\hat{M} > \|x^{(i)}\|_\infty$, thus its Lagrange multiplier must be 0), then these two ways are equivalent.

TABLE 1
The maximal PSD matrix size in the dth order relaxation of the four methods when $s = 2$.

Relaxation approach	Maximal PSD matrix size in dth order relaxation
SOS	$\binom{n_1+n_2-k+d}{d} \times \binom{n_1+n_2-k+d}{d}$
CS-SOS	$\binom{\max\{n_1,n_2\}+d}{d} \times \binom{\max\{n_1,n_2\}+d}{d}$
LME	$\binom{n_1+n_2-k+d}{d} \times \binom{n_1+n_2-k+d}{d}$
CS-LME	$\binom{\max\{n_1,n_2\}+k+d}{d} \times \binom{\max\{n_1,n_2\}+k+d}{d}$

TABLE 2
For each n_1, n_2, k and d, we display sequentially the four binomial values appearing in Table 1: $\binom{n_1+n_2-k+d}{d}$ for SOS, $\binom{\max\{n_1,n_2\}+d}{d}$ for CS-SOS, $\binom{n_1+n_2-k+d}{d}$ for LME, and $\binom{\max\{n_1,n_2\}+k+d}{d}$ for CS-LME.

(n_1, n_2, k)	$d = 2$	$d = 3$	$d = 4$
(4, 3, 1)	(28, 15, 28, 21)	(84, 35, 84, 56)	(210, 70, 210, 126)
(5, 5, 2)	(45, 21, 45, 36)	(165, 56, 165, 120)	(495, 126, 495, 330)
(10, 10, 2)	(190, 66, 190, 91)	(1330, 286, 1330, 455)	(7315, 1001, 7315, 1820)
(15, 15, 3)	(406, 136, 406, 190)	(4060, 816, 4060, 1330)	(31465, 3876, 31465, 7315)
(20, 20, 5)	(666, 231, 666, 351)	(8436, 1771, 8436, 3276)	(82251, 10626, 82251, 23751)

suffers from the same rapid complexity growth just as the dense SOS approach. In contrast, our CS-LME approach leads to SDP problems of a scale comparable with that of CS-SOS, and thus enjoys a less aggressive complexity growth. Meanwhile, it is expected to converge faster than CS-SOS as it incorporates the first-order optimality condition in the relaxation just as the LME approach, as shown in Section 5.

In the above, we compared the maximal PSD matrix size in the SDP problems arising from different relaxation approaches when $s = 2$. To examine the number and size of all the PSD matrices in the SDP problems, one needs, in addition, the structure information of the functions (f, g, h). The next example compares the SDP problem scale in detail for a box-constrained problem with a quadratic objective function.

Example 4.5. Let N and k be positive integers and

$$\mathcal{I}_1 = \{1, \ldots, N\}, \quad \mathcal{I}_2 = \{N+1-k, \ldots, 2N-k\}.$$

Note that this is a special two-block case with $n_1 = n_2 = N$. Consider problem (1.2) with this csp $(\mathcal{I}_1, \mathcal{I}_2)$ and box constraints

$$\begin{aligned} g^{(1)} &= (x_1,\, 1-x_1, \ldots,\, x_{N-k},\, 1-x_{N-k}), \\ g^{(2)} &= (x_{N+1-k},\, 1-x_{N+1-k}, \ldots,\, x_{2N-k},\, 1-x_{2N-k}). \end{aligned} \tag{4.5}$$

The LMEs and CS-LMEs can be similarly given as in (2.19) and (3.18) respectively, and we omit explicit expressions of them for the cleanness of this paper. Let f_1 and f_2 be quadratic functions. We present in Table 3 the number and size of all the PSD matrices in the four different approaches. Table 4 is an instantiation of the numbers in Table 3 for the special case when $N = 10$ and $k = 2$.

Now we consider the general multi-block case. If there exists a common variable in all the blocks, i.e., if there is some $j \in [n]$ such that $j \in \mathcal{I}_i$ for all $i \in [s]$ (e.g., $s = 2$ or Example 5.2), then the LME reformulation does not have correlative sparsity. In this case, the SDP problem scale of the LME approach grows similarly to that of the

TABLE 3
Size and number of PSD matrices in the dth order relaxation of the four methods for the box constrained problem (4.5) with quadratic objective functions.

Relaxation approach	Size and number of PSD matrices size in the dth order relaxation
SOS	one PSD matrix of size $\binom{2N-k+d}{d} \times \binom{2N-k+d}{d}$, $4N-2k$ PSD matrices of size $\binom{2N-k+d-1}{d-1} \times \binom{2N-k+d-1}{d-1}$.
CS-SOS	two PSD matrices of size $\binom{N+d}{d} \times \binom{N+d}{d}$, $4N-2k$ PSD matrices of size $\binom{N+d-1}{d-1} \times \binom{N+d-1}{d-1}$.
LME	one PSD matrix of size $\binom{2N-k+d}{d} \times \binom{2N-k+d}{d}$, $8N-4k$ PSD matrices of size $\binom{2N-k+d-1}{d-1} \times \binom{2N-k+d}{d}$.
CS-LME	two PSD matrices of size $\binom{N+k+d}{d} \times \binom{N+k+d}{d}$, $8N-4k$ PSD matrices of size $\binom{N+k+d-1}{d-1} \times \binom{N+k+d-1}{d-1}$.

TABLE 4
Instantiation of Table 3 when $N = 10$ and $k = 2$. For example, the bottom-right block reads as follows: the 4th order relaxation of the CS-LME approach corresponds to an SDP problem with two 1820-by-1820 PSD matrices and seventy-two 455-by-455 PSD matrices.

Relaxation approach	$d = 2$	$d = 3$	$d = 4$
SOS	$(1, 190)$, $(36, 19)$	$(1, 1330)$, $(36, 190)$	$(1, 7315)$, $(36, 1330)$
CS-SOS	$(2, 66)$, $(36, 11)$	$(2, 286)$, $(36, 66)$	$(2, 1001)$, $(36, 286)$
LME	$(1, 190)$, $(72, 19)$	$(1, 1330)$, $(72, 190)$	$(1, 7315)$, $(72, 1330)$
CS-LME	$(2, 91)$, $(72, 13)$	$(2, 455)$, $(72, 91)$	$(2, 1820)$, $(72, 455)$

dense SOS relaxations. However, in general, though the LME reformulation usually breaks the csp of the original problem, it may have a weaker correlative sparsity. The following example is such an exposition.

Example 4.6. Let $N > k$ be two positive integers. Consider the following csp

$$\mathcal{I}_i = \{(N-k)(i-1)+1, \ldots, (N-k)(i-1)+N\}, \ \forall i = 1, \ldots, s. \tag{4.6}$$

When $N = 3$ and $k = 2$, it corresponds to the csp of the Broyden tridiagonal function [11, Example 3.4]. The directed clique tree (V, A) associated to the sparsity pattern (4.6) is given by

$$A = \{(i, i-1) : i = 2, \ldots, s\}.$$

For each arc $(i, i-1) \in A$, the set of joint indices is:

$$\mathcal{C}_{i,i-1} = \mathcal{I}_i \cap \mathcal{I}_{i-1} = \{(N-k)(i-1)+1, \ldots, (N-k)(i-2)+N\}.$$

Note that $|\mathcal{I}_i| = N$ and $|\mathcal{C}_{i,i-1}| = k$ for each $i \in [s]$. The auxiliary variables are:

$$\bigcup_{i=2}^{s} \bigcup_{j=1}^{k} \{\nu_{i,i-1,(N-k)(i-1)+j}\}. \tag{4.7}$$

For the sparsity pattern (4.6), the maximal clique size in the csp graph of the CS-LME reformulation (3.39) is

$$N + 2k.$$

Table 5

The maximal PSD matrix size in dth order relaxation of the four methods when the csp is given by (4.6).

Relaxation approach	Maximal PSD matrix size in dth order relaxation
SOS	$\binom{(N-k)(s-1)+N+d}{d} \times \binom{(N-k)(s-1)+N+d}{d}$
CS-SOS	$\binom{N+d}{d} \times \binom{N+d}{d}$
LME	$\binom{(N-k)\left\lfloor \frac{N-1}{N-k} \right\rfloor+N+d}{d} \times \binom{(N-k)\left\lfloor \frac{N-1}{N-k} \right\rfloor+N+d}{d}$
CS-LME	$\binom{2k+N+d}{d} \times \binom{2k+N+d}{d}$

In contrast, the maximal clique size in the original LME reformulation (2.16) is

$$(N-k)\left\lfloor \frac{N-1}{N-k} \right\rfloor + N.$$

We give in Table 5 the maximal PSD matrix size of the four methods for solving (1.2) with csp given by (4.6). Table 5 shows that the SDP problem scale of CS-LME is significantly smaller than SOS and LME when $N \gg k$. Recall that N is the size of the blocks while k is the number of overlapping variables between two successive blocks. Thus N/k can be seen as a measure of the partial separability of the problem. We speculate that the larger N/k is, the more efficient the CS-LME approach is compared with the other three approaches[4]. See Example 5.6 for a numerical evidence with $N = 15$, $k = 2$ and $s = 10$.

Remark 4.7. To end this section, we would like to point out that for small-scale problems, the LME approach has outstanding performance, especially when the SOS approach cannot find the global minimum with a low relaxation order, see [23]. For small-scale problems with csp, the LME approach may still be faster than the CS-LME approach because the latter needs to add auxiliary variables to maintain the csp. See Example 5.1 for a numerical example of a small-scale problem.

In general, we expect CS-LME to perform better than the other three approaches when the cliques in the csp graph of the CS-LME reformulation are not much larger than the cliques in the csp graph of the LME reformulation. Since $|\hat{\mathcal{I}}_i| = |\mathcal{I}_i| + |\mathcal{J}_i|$, this occurs when

1. The number of overlapping variables between any two blocks $\mathcal{I}_i$ and $\mathcal{I}_j$ is small;
2. Each node in the directed clique tree $G(V, A)$ returned by Algorithm 3.1 has a small number of children.

These two conditions ensure that only a small number of auxiliary variables $|\mathcal{J}_i|$ must be added to each block.

5. Numerical experiments.

In this section, we present numerical experiments that apply CS-LMEs to solve polynomial optimization problems with a given csp. We directly call the software `TSSOS` [5] [36, 37] to solve the CS-TSSOS relaxation of the CS-LME reformulation (3.39). Note that CS-TSSOS relaxation exploits both correlative and term sparsity in the polynomial optimization problem. As recalled in Section 2.3, the convergence of CS-TSSOS is guaranteed when the CS-SOS relaxation

[4] The overall performance depends on both the SDP problem scale and the convergence rate with respect to the relaxation order d.

[5] https://github.com/wangjie212/TSSOS

is convergent (with option `TS="block"`). The software `Mosek` is applied to solve the SDPs with default settings. The computation is implemented in a Lenovo x1 Yoga laptop, with an Intel® Core(TM) i7-1185G7 CPU at 3.00GHz×4 cores and 16GB of RAM, in the Windows 11 operating system.

For all polynomial optimization problems in this section, we compare the performance of several approaches. First, we solve the problem directly by `CS-TSSOS` with options `TS="block"` and `TS="MD"` respectively (see [36] for more details). Then, we solve the LME reformulation (2.16) introduced in [23] when it exists. Note that when original LMEs are applied, correlative sparsity for the reformulation is usually corrupted. Last, we solve the CS-LME reformulation (3.39). For both LME reformulation (2.16) and CS-LME reformulation (3.39), the `CS-TSSOS` is called with the option `TS="MD"`. Besides that, we use the MATLAB software `Gloptipoly 3` [7] to implement dense relaxations with `Mosek` being applied to solve the SDPs. We say a relaxation *'fail to solve'* when we cannot get a sensible optimal value for it. This is the case when we suspect SDP is unbounded as `Mosek` reaches a negative objective value with a huge absolute value ($< -10^6$).

Example 5.1. (i) Consider the polynomial optimization problem (2.17) in Example 2.5. As mentioned in Example 2.5, its global minimum equals 0. The CS-LMEs for this problem are given by (3.18), and the CS-LME reformulation is (3.19). One may check that the archimedean condition is not satisfied by $\mathrm{IQ}_{\mathcal{I}_1}(h^{(1)}, g^{(1)})$. Besides that, the LME is given by (2.19). Numerical results for solving this problem are presented in Table 6. In the table, 'd' means the relaxation order, 'l' represents the term sparsity level. The columns 'no LME+`block`' and 'no LME+`MD`' are numerical results of applying `CS-TSSOS` directly to the polynomial optimization problem with `TS="block"` and `TS="MD"` respectively, the column 'LME' corresponds to solving the LME reformulation, and the column 'CS-LME' represents the relaxation results of the CS-LME reformulation. The 'error' is the absolute value of the difference of optimal value for this polynomial optimization problem and the approximation computed by the semidefinite relaxation, and 'time' is the time consumption in seconds for computing this approximation. When a superscript * is marked, it means this lower bound was computed with the highest level of term sparsity within the current relaxation order.

From the table, one can see that when there were no LMEs exploited, `CS-TSSOS` could not get an approximation for the global minimum of this problem with high accuracy (say, the error is less than 10^{-6}). Particularly, when $d = 3$, the computed optimal values for both 'no LME+`block`' and 'no LME+`MD`' are less than -10^{13}, and we marked 'fail to solve' in the table. Besides that, when $d = 3$, `Gloptipoly 3` failed to solve the problem (unboundedness suspected), and obtained an approximated value with error equaling $3\cdot 10^{-9}$ in 0.50 second when $d = 4$. In contrast, the LME approach took around 0.23 second to get the approximated global minimum, and the CS-LME approach obtained the approximated minimum in 0.53 second.

(ii) For the polynomial optimization problem in Example 3.3, if we keep the objective function and the csp, but change the constraints to

$$g^{(1)}(x^{(1)}) = \left(1 - {x^{(1)}}^T x^{(1)},\, x_1^{(1)},\, x_2^{(1)}\right), \quad g^{(2)}(x^{(2)}) = \left(1 - {x^{(2)}}^T x^{(2)},\, x_1^{(2)},\, x_2^{(2)}\right)\ ,$$

then the CS-LME becomes

$$\begin{aligned} \lambda_1^{(1)} &= -\frac{1}{2}{x^{(1)}}^\top F^{(1)}, \quad \lambda_2^{(1)} = F_1^{(1)} + 2x_1^{(1)}\lambda_1^{(1)}, \quad \lambda_3^{(1)} = F_2^{(1)} + 2x_2^{(1)}\lambda_1^{(1)}, \\ \lambda_1^{(2)} &= -\frac{1}{2}{x^{(2)}}^\top F^{(2)}, \quad \lambda_2^{(2)} = F_1^{(2)} + 2x_1^{(2)}\lambda_1^{(2)}, \quad \lambda_3^{(2)} = F_2^{(2)} + 2x_2^{(2)}\lambda_1^{(2)}. \end{aligned}$$

However, one may check this problem does not have LMEs.

TABLE 6
Numerical results for Example 5.1(i)

d	l	no LME+`block`		no LME+`MD`		LME		CS-LME	
		error	time	error	time	error	time	error	time
3	1	fail to solve		fail to solve		not defined		not defined	
3	2	*fail to solve		fail to solve					
3	3			fail to solve					
3	4			*fail to solve					
4	1	0.0134	0.06s	0.0437	0.03s	$2 \cdot 10^{-8}$	0.23s	0.0014	0.36s
4	2	*0.0134	0.07s	0.0437	0.03s			$\mathbf{1 \cdot 10^{-7}}$	**0.53s**
4	3			0.0140	0.13s				
⋮	⋮	⋮	⋮	⋮	⋮				
10	1	0.0038	25.76s	0.0337	8.26s				
10	2	0.0038	74.82s	0.0152	11.18s				

With the new constraints, one can similarly check that the global minimum is still 0. Numerical results for solving this problem are presented in Table 7, where symbols and notation are similarly defined as in Table 6. From the table, one can see that without CS-LMEs, `CS-TSSOS` cannot find the global minimum with satisfying error in 61 seconds for the option `TS="block"`, and in 78 seconds for the option `TS="MD"`. Besides that, `Gloptipoly` got the lower bound $-2 \cdot 10^{-5}$ in 0.30 second for $d = 3$, and got $-5 \cdot 10^{-9}$ in 0.52 second for $d = 4$. For the CS-LME approach, we obtained an approximation $-9 \cdot 10^{-7}$ for the global minimum in 1.69 seconds.

TABLE 7
Numerical results for Example 5.1(ii)

d	l	no LME+`block`		no LME+`MD`		CS-LME	
		error	time	error	time	error	time
3	1	0.0146	0.02s	0.0531	0.01s	not defined	
3	2	*0.0140	0.02s	0.0480	0.01s		
4	1	0.0074	0.04s	0.0495	0.04s	0.0018	0.41s
4	2	*0.0070	0.06s	0.0450	0.04s	0.0016	0.42s
5	1	0.0045	0.15s	0.0492	0.14s	$2 \cdot 10^{-5}$	0.98s
5	2	*0.0044	0.28s	0.0448	0.16s	$\mathbf{9 \cdot 10^{-7}}$	**1.69s**
10	1	0.0049	6.45s	0.0437	13.82s		
10	2	*0.0034	61.41s	0.0245	12.99s		
10	3			0.0035	59.52s		
⋮	⋮			⋮	⋮		
10	8			*0.0034	78.16s		

For all remaining examples in this section, symbols and notation in tables are similarly defined as in Table 6, and we shall not repeat explaining them, for the neatness of this paper.

Example 5.2. Consider the csp given in Example 3.6. For each $i = 1, \ldots, 5$, we let $f_i(x^{(i)})$ be the Choi-Lam's form

$$f_i(x^{(i)}) = (x_1^{(i)} x_2^{(i)})^2 + (x_1^{(i)} x_3^{(i)})^2 + (x_2^{(i)} x_3^{(i)})^2 + {x_4^{(i)}}^4 - 4 x_1^{(i)} x_2^{(i)} x_3^{(i)} x_4^{(i)},$$

and let

$$g^{(i)} = (1 - x^{(i)\top} x^{(i)}), \quad h^{(i)} = \emptyset.$$

Again, by the inequality of arithmetic and geometric means, all f_i are nonnegative, and $f_i(x^{(i)}) = 0$ when $x_1^{(i)} = \cdots = x_4^{(i)}$. Thus we know the optimal value for minimizing $f_1(x^{(1)}) + \cdots + f_5(x^{(5)})$ over the set given by $g^{(i)}(x^{(i)}) \geq 0$ for all $i = 1, \ldots, 5$ is 0. For this problem, the CS-LMEs can be given as

$$\lambda^{(i)} = -\frac{x^{(i)\top} F^{(i)}}{2}.$$

However, there do not exist LMEs, which can be similarly shown as in Example 3.1. Numerical results of solving this problem using CS-TSSOS directly, the LME approach, and the CS-LME approach are presented in Table 8.

From the table, one can see that without CS-LMEs, `CS-TSSOS` cannot find the global minimum with the option `TS="MD"` (interestingly, it returned the same lower bound -0.1709 for all $d = 2, \ldots, 15$), and cannot get an approximation for the global minimum with an error less than 0.0001 in 6807 seconds with `TS="block"`. Moreover, `Gloptipoly 3` obtained the lower bound -0.1709 when $d = 2$ using 0.99 second, and obtained the lower bound -0.0135 in 346.42 seconds when $d = 3$. In contrast, the CS-LME approach took 11.35 seconds to obtain an approximated minimum with an error equal to $6 \cdot 10^{-6}$, and took 107.62 seconds to obtain an approximated minimum with an error equal to $3 \cdot 10^{-8}$.

TABLE 8
Numerical results for Example 5.2

d	l	no LME+`block`		no LME+`MD`		CS-LME	
		error	time	error	time	error	time
2	1	*0.0531	0.01s	*0.1709	0.01s	not defined	
3	1	*0.0480	0.01s	*0.1709	0.02s	0.0080	5.82s
4	1	*0.0495	0.04s	*0.1709	0.05s	$1 \cdot 10^{-5}$	6.55s
4	2					$6 \cdot 10^{-6}$	11.35s
5	1	*0.0450	0.04s	*0.1709	0.16s	$\mathbf{3 \cdot 10^{-8}}$	**107.62s**
⋮	⋮	⋮	⋮	⋮	⋮		
15	1	*0.0001	6807.45s	*0.1709	554.09s		

Example 5.3. Consider the box-constrained problem in Example 4.5. Let $n_1 = n_2 = 10,\ k = 2$, and let $(i = 1, 2)$

$$f_i(x^{(i)}) = \left(\sum\nolimits_{j=1}^{10} x_j^{(i)} + 1\right)^2 - 4\left(\sum\nolimits_{j=1}^{9} x_j^{(i)} x_{j+1}^{(i)} + x_1^{(i)} + x_{10}^{(i)}\right).$$

The LMEs and CS-LMEs can be similarly given by (2.19) and (3.18), respectively. One may check that the archimedean condition is not satisfied by $\mathrm{IQ}_{\mathcal{I}_1}(h^{(1)}, g^{(1)})$. Furthermore, for $d = 2, \ldots, 3$, the structure of SDPs obtained by the dense relaxation, CS-SOS relaxations, the LME approach, and the CS-LME approach are given in Table 4.

The minimum for this problem is achieved at the KKT point $(1, 0, \ldots, 0, 1)$, which equals 0 (see also [23]). This can also be numerically certified by `Gloptipoly 3` via the *flat truncation* [18]. Indeed, `Gloptipoly 3` got an approximation to the global minimum $-3 \cdot 10^{-8}$ in 26.25 seconds. Numerical results of solving this problem using CS-TSSOS directly, the LME approach and the CS-LME approach are presented in Table 9. From the table, one can see that without LMEs, `CS-TSSOS` could not find an approximation for the global minimum with a desired accuracy when `TS` = "`MD`" within 11.51 seconds, and took 36.67 seconds to get the minimum when `TS` = "`block`". The LME approach took 15.79 seconds to get the approximation with the desired accuracy. In contrast, the CS-LME approach only took 2.46 seconds to get an approximated global minimum with the error equal to $4 \cdot 10^{-7}$.

TABLE 9
Numerical results for Example 5.3

d	l	no LME+`block`		no LME+`MD`		LME		CS-LME	
		error	time	error	time	error	time	error	time
2	1	*0.0067	1.08s	0.0739	0.10s	$^*1 \cdot 10^{-7}$	15.79s	$^*\mathbf{4 \cdot 10^{-7}}$	**2.46s**
3	1	$9 \cdot 10^{-9}$	36.67s	*0.0558	0.78s				
4	1			*0.0105	11.51s				

Example 5.4. Let $s = 2$ and

$$\mathcal{I}_1 = \{1, 2, 3, 7\}, \quad \mathcal{I}_2 = \{4, 5, 6, 7\}.$$

Consider the polynomial optimization problem (1.2) with csp $\{\mathcal{I}_1, \mathcal{I}_2\}$, where

$$\begin{gathered} f_1(x^{(1)}) = x_1^4 x_2^2 + x_2^4 x_3^2 + x_3^4 x_1^2 - 3(x_1 x_2 x_3)^2 + x_2^2 + x_7^2(x_1^2 + x_2^2 + x_3^2), \\ f_2(x^{(2)}) = x_4 x_5 (10 - x_6) + x_7^2 (x_4 + 2x_5 + 3x_6); \\ g_1^{(1)}(x^{(1)}) = x_1 - x_2 x_3, \;\; g_2^{(1)}(x^{(1)}) = -x_2 + x_3^2, \\ g_1^{(2)}(x^{(2)}) = 1 - x_4 - x_5 - x_6, \;\; g_2^{(2)}(x^{(2)}) = x_4, \;\; g_3^{(2)}(x^{(2)}) = x_5, \;\; g_4^{(2)}(x^{(2)}) = x_6. \end{gathered}$$

Since $x_1^4 x_2^2 + x_2^4 x_3^2 + x_3^4 x_1^2 \geq 3(x_1 x_2 x_3)^2$ by the inequality of arithmetic and geometric means, we have $f_1(x^{(1)}) \geq 0$ with the equality holds when $x_1 = x_2 = x_3 = x_7 = 0$. On the other hand, f_2 is nonnegative on the feasible set given by $g^{(2)}(x^{(2)}) \geq 0$, and $f_2(x^{(2)}) = 0$ when $x_4 x_5 = 0$ and $x_7 = 0$. So, the global minimum for this problem is 0, which is attain at $(0, 0, 0, t, 0, 0, 0)$ and $(0, 0, 0, 0, t, 0, 0)$ for all $t \in [0, 1]$. Also, one may check that this problem has an unbounded feasible set. For this problem, let

$$F^{(1)} = \nabla_1 f_1 + \nu_{2,1,7} e_4, \quad F^{(2)} = \nabla_2 f_2 + \nu_{2,1,7} e_4,$$

then the CS-LMEs are

$$\begin{gathered} \lambda_1^{(1)} = F_1^{(1)}, \quad \lambda_2^{(1)} = [-x_3, -1, 0, 0] \cdot F^{(1)}, \\ \lambda_1^{(2)} = -x_{4:6}^{\top} F_{1:3}^{(2)}, \quad \lambda_2^{(2)} = F_1^{(2)} + \lambda_1^{(2)}, \quad \lambda_3^{(2)} = F_2^{(2)} + \lambda_1^{(2)}, \quad \lambda_4^{(2)} = F_3^{(2)} + \lambda_1^{(2)}. \end{gathered}$$

The numerical results for solving this problem are presented in Table 10. From the table, one can see that when there were no LMEs exploited, `CS-TSSOS` could not get an approximation for the global minimum of this problem with an error less than 0.0001 within 271.95 seconds, while the original LME approach took around 84.13 seconds to get the approximated value with an error equaling $2 \cdot 10^{-7}$. Moreover, when $d = 3$ and 4, `Gloptipoly 3` failed to solve the problem (unboundedness suspected), and it took 2264 seconds to get the lower bound -120.82 when $d = 5$. In contrast, the CS-LME approach obtained an approximated minimum whose error was $9 \cdot 10^{-8}$ in 18.54 seconds.

Table 10
Numerical results for Example 5.4

d	l	no LME+`block` error	no LME+`block` time	no LME+`MD` error	no LME+`MD` time	LME error	LME time	CS-LME error	CS-LME time
3	1	fail to solve		fail to solve		not defined		not defined	
3	2	$^* > 10^8$	0.18s	fail to solve		not defined		not defined	
⋮	⋮			⋮	⋮	⋮	⋮	⋮	⋮
3	5			*fail to solve		not defined		not defined	
4	1	$> 10^7$	0.47s	fail to solve		1519.49	4.95s	645.71	0.77s
4	2	$^* > 10^5$	0.59s	$> 10^6$	0.45s	35.36	5.28s	23.62	0.94s
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
5	2	*265.61	2.60s	$> 10^5$	1.44s	$2 \cdot 10^{-7}$	84.13s	0.0324	5.42s
⋮	⋮	⋮	⋮	⋮	⋮			⋮	⋮
6	1	18.19	6.28	102.78	5.59s			$\mathbf{9 \cdot 10^{-8}}$	**18.54s**
⋮	⋮	⋮	⋮	⋮	⋮				
8	2	*0.0001	224.77s	0.0079	75.64s				
⋮	⋮			⋮	⋮				
8	5			*0.0002	322.68s				

Example 5.5. Let $s = 5$ and

$$\mathcal{I}_1 = \{1,2,3,4,17,18,19\},\ \mathcal{I}_2 = \{5,6,7,8,18,19,20\},$$
$$\mathcal{I}_3 = \{9,10,18,19,20\},\ \mathcal{I}_4 = \{11,12,17,18\},\ \mathcal{I}_5 = \{13,14,15,16,17\}.$$

Consider the polynomial optimization problem (1.2) with csp $(\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_5)$, where

$$\begin{aligned}
f_1(x^{(1)}) &= (x_1 - x_{17})^2 + (x_2 - x_{18})^2 + (x_3 - x_{19})^2 + x_4^2 x_{17},\\
f_2(x^{(2)}) &= x_{18}^2 + x_{19}^2 + x_{20}^2 - x_5(x_6 + x_7 + x_8),\\
f_3(x^{(3)}) &= x_9 x_{10}(20 - x_{18} - x_{19} - x_{20}),\\
f_4(x^{(4)}) &= (x_{11} - x_{17})^2 + (x_{12} + x_{18} - 1)^2,\\
f_5(x^{(5)}) &= (x_{17} - x_{13} + x_{14})^2 + x_{15} x_{16},\\
g^{(1)}(x^{(1)}) &= \left(\mathbf{1}_7 - x^{(1)},\ x^{(1)}_{1:4} + \mathbf{1}_7\right),\\
g^{(2)}(x^{(2)}) &= \left(3 - 2\sum\nolimits_{j=1}^{3} x_j^{(2)} - \sum\nolimits_{j=5}^{7} x_j^{(2)} - x_4^{(2)},\ x_1^{(2)}, \dots,\ x_7^{(2)}\right),\\
g^{(3)}(x^{(3)}) &= \left(1 - \sum\nolimits_{j=1}^{5} x_j^{(3)},\ x_1^{(3)},\ x_2^{(3)}\right),\\
g^{(4)}(x^{(4)}) &= 1 - {x^{(4)}}^\top x^{(4)}, \quad g^{(5)}(x^{(5)}) \ = \ x^{(5)}.
\end{aligned}$$

It is clear that except $f^{(2)}$, all other $f^{(i)}$ are nonnegative over the set given by $(g^{(1)}, g^{(2)}, \dots, g^{(5)})$. For $f^{(2)}$, its minimum $-\frac{9}{8}$ is attained at the KKT point $x^{(2)} = \left(\frac{3}{4}, 0, 0, \frac{3}{2}, 0, 0, 0\right)$. Indeed, one may check that the global minimum for this problem is $-\frac{9}{8}$. For this problem, the set of edges is

$$A = \{(2,1),\ (3,2),\ (4,1),\ (5,4)\}.$$

The auxiliary variables are

$$\nu_{2,1,18},\ \nu_{2,1,19},\ \nu_{3,2,18},\ \nu_{3,2,19},\ \nu_{3,2,20},\ \nu_{4,1,17},\ \nu_{4,1,18},\ \nu_{5,4,17}.$$

If we let $F^{(i)}$ be given as in (3.35), then CS-LMEs are

$$\begin{gathered}
\lambda^{(1)}_{1:4} = -\frac{1}{2}\cdot F^{(1)}_{1:4}\circ(\mathbf{1}+x_{1:4}),\quad \lambda^{(1)}_{5:7} = -F^{(1)}_{5:7},\quad \lambda^{(1)}_{8:11} = F^{(1)}_{1:4}+\lambda^{(1)}_{1:4};\\
\lambda^{(2)}_1 = -\tfrac{1}{3}{F^{(2)}}^\top x^{(2)},\quad \lambda^{(2)}_{2:4} = 2\lambda^{(2)}_1 + F^{(2)}_{1:3},\quad \lambda^{(2)}_{5:8} = 2\lambda^{(2)}_1 + F^{(2)}_{4:7};\\
\lambda^{(3)}_1 = -{F^{(3)}}^\top x^{(3)},\quad \lambda^{(3)}_{2:3} = \lambda^{(3)}_1 + F^{(3)}_{1:2};\quad \lambda^{(4)} = -\tfrac{1}{2}{F^{(4)}}^\top x^{(4)};\quad \lambda^{(5)} = F^{(5)}.
\end{gathered}$$

We would like to remark that the tuple $(g^{(1)}, g^{(2)}, \ldots, g^{(5)})$ is singular, so original LMEs do not exist. The numerical results for solving this problem are presented in Table 11. From the table, one can see that when there were no LMEs exploited, `CS-TSSOS` could not get an approximation for the global minimum with an error less than 0.001 in 7697.33 seconds. Moreover, `Gloptipoly 3` suspected unboundedness when $d = 3$, and the 4th order dense relaxation cannot be solved due to the memory limit. In contrast, the CS-LME approach obtained an approximated minimum whose error was $1\cdot 10^{-7}$ in 53.73 seconds.

TABLE 11
Numerical results for Example 5.5

d	l	no LME+`block`		no LME+`MD`		CS-LME	
		error	time	error	time	error	time
2	1	fail to solve		$^* > 10^6$	0.28s	9.5731	1.32s
2	2	$^* > 10^6$	0.37			0.3085	1.50s
⋮	⋮					⋮	⋮
2	5					*0.1417	10.05s
3	1	1.6047	3.59s	1295.25	0.71s	$4\cdot 10^{-7}$	60.41s
3	2	*fail to solve		1276.92	0.76s	$\mathbf{1\cdot 10^{-7}}$	**53.73s**
⋮	⋮	⋮	⋮	⋮	⋮		
5	2	*0.0069	16663.91s	9.3531	252.95s		
5	3			0.0862	7697.33s		

For the following two examples, we do not run `Gloptipoly 3` for solving them, since the problem scales are too large for dense SOS relaxations.

Example 5.6. Consider the correlative sparsity pattern given in Example 4.6. Let $s = 10$, $N = 15$, and $k = 2$. For each $i \in [10]$, let

$$\begin{aligned}
f_i(x) = \left({x^{(i)}}^T x^{(i)}\right)^2 &- 4\left((x^{(i)}_1 x^{(i)}_2)^2 + \cdots + (x^{(i)}_4 x^{(i)}_5)^2 + (x^{(i)}_5 x^{(i)}_1)^2\right)\\
&+ \left(x^{(i)}_1 + \cdots + x^{(i)}_5 - (x^{(i)}_{6:10})^\top x^{(i)}_{11:15}\right)^2.
\end{aligned}$$

Consider the unconstrained polynomial optimization problem

$$\min_x \quad f_1(x^{(1)}) + \cdots + f_{10}(x^{(10)}). \tag{5.1}$$

For each $i \in [10]$, the $\left({x^{(i)}}^T x^{(i)}\right)^2 - 4\left((x^{(i)}_1 x^{(i)}_2)^2 + \cdots + (x^{(i)}_4 x^{(i)}_5)^2 + (x^{(i)}_5 x^{(i)}_1)^2\right)$ is the *Horn's form* [33], which is a nonnegative homogeneous polynomial. Thus the global minimum of (5.1) is 0. For unconstrained problems, the system

$$\phi^{(i)}(x^{(i)}, \nu^{(i)}) = 0, \quad \psi^{(i)}(x^{(i)}, \nu^{(i)}) \geq 0, \quad \forall i \in [10]$$

reduces to

$$F^{(1)}(x^{(1)}, \nu^{(1)}) = F^{(2)}(x^{(2)}, \nu^{(2)}) = \cdots = F^{(10)}(x^{(10)}, \nu^{(10)}) = 0,$$

where every $F^{(i)}$ is given in (3.35) with auxiliary variables given in (4.7). Thus, the CS-LME typed reformulation (3.39) becomes

$$\begin{array}{ll} \min & f_1(x^{(1)}) + \cdots + f_{10}(x^{(10)}) \\ \text{s.t.} & F^{(1)}(x^{(1)}, \nu^{(1)}) = F^{(2)}(x^{(2)}, \nu^{(2)}) = \cdots = F^{(10)}(x^{(10)}, \nu^{(10)}) = 0 \end{array} \tag{5.2}$$

Similarly, the original LME reformulation (3.39) for (5.7) becomes

$$\begin{array}{ll} \min & f_1(x^{(1)}) + \cdots + f_{10}(x^{(10)}) \\ \text{s.t.} & \nabla(f_1 + f_2 + \cdots + f_{10})(x) = 0 \end{array} \tag{5.3}$$

The numerical results for solving this problem are presented in Table 12. From the table, one can see that when there were no LMEs exploited, `CS-TSSOS` could not get a sensible approximation for the global minimum of this problem within 487.31 seconds, while the original LME approach took around 270.40 seconds to get an approximated global minimum. In contrast, the CS-LME approach obtained an approximated minimum whose error was $7 \cdot 10^{-10}$ in 20.48 seconds.

TABLE 12
Numerical results for Example 5.6

d	l	no LME+`block`		no LME+`MD`		LME		CS-LME	
		error	time	error	time	error	time	error	time
2	1	*fail to solve		*fail to solve		fail to solve		fail to solve	
2	2					*fail to solve		*fail to solve	
3	1	$^* > 10^8$	78.43s	$^* > 10^8$	7.26s	$6 \cdot 10^{-11}$	270.40s	$\mathbf{7 \cdot 10^{-10}}$	**20.48s**
4	1	*out of memory		$^* > 10^6$	487.31s				
5	1			*out of memory					

Example 5.7. In this example, we present numerical results by varying the number of blocks s. For each $i \in [s]$, let $\mathcal{I}_i := \{9i-8, 9i-7, \ldots, 9i+1\}$. Consider the following optimization problem

$$\left\{ \begin{array}{ll} \min\limits_{x} & f_1(x^{(1)}) + f_2(x^{(2)}) + \cdots + f_s(x^{(s)}) \\ s.t. & x_1^{(1)} \geq 0,\ x_1^{(i)} + x_2^{(i)} + \cdots + x_{10}^{(i)} \leq 1,\ x_{2:10}^{(i)} \geq 0,\ \ i \in [s] \end{array} \right. \tag{5.4}$$

In the above,

$$f_i(x^{(i)}) = \sum\nolimits_{j=1}^{3} x_{2j}^{(i)} x_{2j+1}^{(i)} + \left(\sum\nolimits_{j=7}^{9} (x_j^{(i)})^3 + x_7^{(i)} x_8^{(i)} x_9^{(i)} \right) x_{10}^{(i)}, \ \ i \in [s].$$

Since all variables are nonnegative, and each $f_i(x^{(i)})$ reaches 0 at $x^{(i)} = \mathbf{0}$, it is clear that (5.4) has the csp $(\mathcal{I}_1, \ldots, \mathcal{I}_s)$ and its minimum value equals 0. Moreover, because for all $s \geq 2$, the matrix $G(x)$ given as in (2.12) does not have full column rank at

e_{10}. So (5.4) does not have LMEs. For each $i \in [s-1]$, we have the auxiliary variable $\nu_{i+1,i,9i+1}$. Let $F^{(i)}$ be given in (3.35), then CS-LMEs are

$$\begin{gathered}\lambda_1^{(1)} = -{F^{(1)}}^\top x^{(1)}, \quad \lambda_{2:11}^{(1)} = F^{(1)} + \lambda_1^{(1)};\\ \lambda_1^{(i)} = -{F^{(i)}}^\top x^{(i)}, \quad \lambda_{2:10}^{(i)} = F_{2:10}^{(i)} + \lambda_1^{(i)}, \quad (i = 2, \ldots, s).\end{gathered}$$

The numerical results for solving this problem with $s = 2, \ldots, 7$ are presented in Table 13. In the table, 's' represents the quantity s in (5.4), and all other symbols and notations are similarly defined as in Table 6 (see Example 5.1). When $s = 2$, one can see that when there were no CS-LMEs exploited, `CS-TSSOS` could not get an approximation for the global minimum of this problem with an error less than 0.01 in 11366.94 seconds. In contrast, the CS-LME approach obtained an approximated minimum whose error was $5 \cdot 10^{-9}$ in 1192.89 seconds. Moreover, when $s = 3, \ldots, 7$, we do not present numerical results with relaxation order $d = 3$ since we cannot get lower bounds that are close to 0. Also, results of approaches without CS-LMEs are not presented for $s \geq 3$ and $d = 4$, because close lower bounds cannot be computed by these approaches with reasonable time consumption.

TABLE 13
Numerical results for Example 5.7

s	d	l	no LME+`block`		no LME+`MD`		CS-LME	
			error	time	error	time	error	time
2	3	1	0.0735	71.54s	5369.40	2.88s	3.6067	16.85s
	3	2	*0.0230	196.25s	624.22	4.25s	0.0680	35.02s
	3	3			0.0238	78.46s	0.0091	353.71s
	3	4			*0.0230	216.41s	0.0071	834.23s
	4	1	0.0205	11366.94s	23.77	682.21	$\mathbf{5 \cdot 10^{-9}}$	**1192.89s**
	4	2			0.0104	71235.47	-	
3	4	1					$\mathbf{7 \cdot 10^{-8}}$	**1965.19s**
4	4	1					$\mathbf{2 \cdot 10^{-7}}$	**2432.45s**
5	4	1					$\mathbf{3 \cdot 10^{-7}}$	**2868.47s**
6	4	1					$\mathbf{3 \cdot 10^{-7}}$	**4136.36s**
7	4	1					$\mathbf{3 \cdot 10^{-7}}$	**4567.80s**

6. Conclusions and discussions.

We consider correlatively sparse polynomial optimization problems. We introduce CS-LMEs to construct CS-LME reformations for polynomial optimization problems. Under some general assumptions, we show that correlative SOS relaxations can get tighter lower bounds when solving the CS-LME reformulation instead of the original optimization problem. Moreover, asymptotic convergence is guaranteed if the sequel of CS-SOS relaxations for the original polynomial optimization is convergent. Numerical examples are presented to show the superiority of our new approach.

For future work, one wonders if the CS-SOS relaxation has finite convergence for solving CS-LME reformulations. Indeed, finite convergence for the original LME reformulation in [23] is guaranteed under mild conditions. As demonstrated in Section 5, the CS-LME approach usually finds the global minimum (up to a negligible numerical error) for polynomial optimization problems with a low relaxation order. However, it is still open that if the finite convergence is guaranteed theoretically or not, even for generic cases. Moreover, when the correlatively sparse polynomial optimization (1.2) is given by generic polynomials, its KKT ideal is zero-dimensional.

Thus the real variety given by equality constraints in (3.39) is a finite set. For the classical Moment-SOS relaxations, finite convergence is theoretically guaranteed when equality constraints of the polynomial optimization give a zero-dimensional real variety, as shown in [19]. So, it is interesting to ask whether the analogous is true for CS-SOS relaxations. Besides that, our numerical experiments indicate that the CS-LME approach can usually find the global minimum for polynomial optimization problems even if some $\mathrm{IQ}_{\mathcal{I}^{(i)}}(h^{(i)}, g^{(i)})$ is not archimedean. Therefore, an interesting question is whether the CS-LME approach has guaranteed asymptote or finite convergence without the archimedean condition for every $\mathrm{IQ}_{\mathcal{I}^{(i)}}(h^{(i)}, g^{(i)})$.

At last, we would like to remark that LMEs have broad applications in many polynomial defined problems. Therefore, a natural question is how to apply CS-LMEs to these applications. For example, when a saddle point problem is given by polynomials with correlative sparsity, can we apply CS-LMEs to construct polynomial optimization reformulation similar to the one in [31] for finding saddle points?

Appendix A. Computing LMEs and CS-LMEs. We introduce how to find LMEs and CS-LMEs for practical implementation. As mentioned in Subsection 2.4 and Section 3, finding LMEs (resp., CS-LMEs) is equivalent to finding matrices of polynomials $L(x)$, $D(x)$ (resp., $L^{(i)}(x)$, $D^{(i)}(x)$) such that (2.14) (resp. (3.3)) holds. Note that the matrices $G(x)$ and $G^{(i)}(x)$ only depend on constraints, and LMEs can be viewed as special cases of CS-LMEs that there only exists one block, i.e., $s = 1$. Here we only introduce how to get CS-LMEs, and the methodology for finding LMEs is similar.

Suppose the matrix of polynomial $G^{(i)}(x^{(i)})$ has full column rank over $\mathbb{C}^{n_i}$. In general, (3.3) gives a linear equation system. Denote $\hat{m}_i := m_i + \ell_i$, and

$$L^{(i)}(x^{(i)}) := \begin{bmatrix} L_{1,1}(x^{(i)}) & L_{1,2}(x^{(i)}) & \dots & L_{1,n_i}(x^{(i)}) \\ \vdots & \vdots & \vdots & \vdots \\ L_{\hat{m}_i,1}(x^{(i)}) & L_{\hat{m}_i,2}(x^{(i)}) & \dots & L_{\hat{m}_i,n_i}(x^{(i)}) \end{bmatrix},$$

$$D^{(i)}(x^{(i)}) := \begin{bmatrix} D_{1,1}(x^{(i)}) & D_{1,2}(x^{(i)}) & \dots & D_{1,\hat{m}_i}(x^{(i)}) \\ \vdots & \vdots & \vdots & \vdots \\ D_{\hat{m}_i,1}(x^{(i)}) & D_{\hat{m}_i,2}(x^{(i)}) & \dots & D_{\hat{m}_i,\hat{m}_i}(x^{(i)}) \end{bmatrix}.$$

Suppose all entries in $L^{(i)}(x^{(i)})$ and $D^{(i)}(x^{(i)})$ are polynomials whose degrees are not greater than d. For each j, k, let (here for the $\alpha = (\alpha_1, \dots, \alpha_{n_i}) \in \mathbb{N}_d^{n_i}$, we denote ${x^{(i)}}^{\alpha} := {x_1^{(i)}}^{\alpha_1} {x_2^{(i)}}^{\alpha_2} \dots {x_{n_i}^{(i)}}^{\alpha_{n_i}}$)

$$L_{j,k}(x^{(i)}) = \sum_{\alpha \in \mathbb{N}_d^{n_i}} L_{j,k,\alpha} \cdot {x^{(i)}}^{\alpha}, \quad D_{j,k}(x^{(i)}) = \sum_{\alpha \in \mathbb{N}_d^{n_i}} D_{j,k,\alpha} \cdot {x^{(i)}}^{\alpha}. \tag{A.1}$$

Then (3.3) can be written as the following linear equation system in variables $L_{j,k,\alpha}$ and $D_{j,k,\alpha}$:

$$\begin{aligned} &\sum_{l=1}^{n_i} \left(\sum_{\alpha \in \mathbb{N}_d^{n_i}} L_{j,l,\alpha} \cdot {x^{(i)}}^{\alpha} \right) \cdot \frac{\partial c_k^{(i)}}{\partial x_l^{(i)}}(x^{(i)}) + \left(\sum_{\alpha \in \mathbb{N}_d^{n_i}} D_{j,k,\alpha} \cdot {x^{(i)}}^{\alpha} \right) c_k^{(i)}(x^{(i)}) \\ &= \begin{cases} 1 & \text{if} \quad j = k, \\ 0 & \text{if} \quad j \neq k, \end{cases} \qquad (j \in [\hat{m}_i],\ k \in [\hat{m}_i]). \end{aligned} \tag{A.2}$$

We remark that in (A.2), the equality means that the polynomials on both sides are identically equaled. By [23, Proposition 5.2], since $G^{(i)}(x)$ has full column rank over $\mathbb{C}^{n_i}$, the system (A.2) must have solutions when d is large enough. Therefore, for each $i \in [s]$, we solve the linear system (A.2) for solutions with a given degree d. If we get a solution to (A.2), then we recover polynomial matrices $L^{(i)}(x^{(i)})$ and $D^{(i)}(x^{(i)})$ (hence CS-LMEs) using this solutions; otherwise, we let $d \leftarrow d+1$ and solve (A.2) with the updated degree d, until a solution is obtained.

Sometimes, one may get CS-LMEs without actually computing polynomial matrices $L^{(i)}(x^{(i)})$ and $D^{(i)}(x^{(i)})$. Instead, CS-LMEs can be directly obtained using the 'multiplication-cancellation' trick [6]. This is shown in the following example.

Example A.1. Consider the case that

$$g^{(i)}(x^{(i)}) = \left(1 - {x^{(i)}}^\top x^{(i)},\ x_1^{(i)}, \ldots, x_{n_i}^{(i)}\right).$$

Then the KKT-typed system (3.27) for the ith block implies that

$$F^{(i)}(z^{(i)}) = -2\lambda_1^{(i)} \cdot x^{(i)} + \sum\nolimits_{j=1}^{n_i} \lambda_{j+1}^{(i)} \cdot e_j, \tag{A.3}$$

$$\lambda_1^{(i)} \perp 1 - {x^{(i)}}^\top x^{(i)}, \quad \lambda_{j+1}^{(i)} \perp x_j^{(i)}\ (j \in [n_i]). \tag{A.4}$$

By multiplying ${x^{(i)}}^\top$ on both sides of (A.3), we get

$${x^{(i)}}^\top F^{(i)}(z^{(i)}) = -2\lambda_1^{(i)} \cdot {x^{(i)}}^\top x^{(i)} + \sum\nolimits_{j=1}^{n_i} \lambda_{j+1}^{(i)} \cdot x_j^{(i)}.$$

Note that (A.4) implies that $\lambda_1^{(i)} \cdot {x^{(i)}}^\top x^{(i)} = \lambda_1^{(i)}$ and $\lambda_{j+1}^{(i)} \cdot x_j^{(i)} = 0$. So we further have

$${x^{(i)}}^\top F^{(i)}(z^{(i)}) = -2\lambda_1^{(i)}.$$

Therefore, again by (A.3), we get CS-LMEs that

$$\lambda_1^{(i)} = -{x^{(i)}}^\top F^{(i)}(z^{(i)})/2, \quad \lambda_{j+1}^{(i)} = F_j^{(i)}(z^{(i)}) + 2\lambda_1^{(i)} \cdot x_j^{(i)}\ (j \in [n_i]).$$

We remark that though we do not get explicit expressions for $L^{(i)}(x^{(i)})$ and $D^{(i)}(x^{(i)})$, essentially, this trick is equivalent to finding solutions for (3.3). For instance, the step of multiplying ${x^{(i)}}^\top$ on both sides of (A.3) means that the first row of $L^{(i)}(x^{(i)})$ is ${x^{(i)}}^\top$. Besides that, for some commonly used constraints (e.g., box, ball, simplex, etc.), LMEs are explicitly given in [23], and they can be similarly applied to the construction of CS-LMEs.

Acknowledgments. The authors would like to thank the editor and anonymous reviewers for all their valuable comments and suggestions, which led to an improvement of the manuscript. We also thank Jiawang Nie and Jie Wang for their inspiring and helpful comments. Zheng Qu is partially supported by NSFC Young Scientist Fund grant No. 12001458 and Hong Kong Research Grants Council General Research Fund grant No. 17317122. Xindong Tang is partially supported by the Start-up Fund P0038976/BD7L from The Hong Kong Polytechnic University.

[6]This trick was introduced by Professor Jiawang Nie in his research group discussions. It is also mentioned in Section 6.3 of his new book *Moment and Polynomial Optimization* [24].

REFERENCES

[1] J. R. S. Blair and B. Peyton, *An introduction to chordal graphs and clique trees*, in Graph Theory and Sparse Matrix Computation, A. George, J. R. Gilbert, and J. W. H. Liu, eds., New York, NY, 1993, Springer New York, pp. 1–29.

[2] D. Cifuentes, C. Harris, and B. Sturmfels, *The geometry of SDP-exactness in quadratic optimization*, Mathematical programming, 182 (2020), pp. 399–428.

[3] E. De Klerk and M. Laurent, *On the Lasserre hierarchy of semidefinite programming relaxations of convex polynomial optimization problems*, SIAM Journal on Optimization, 21 (2011), pp. 824–832.

[4] J. Demmel, J. Nie, and V. Powers, *Representations of positive polynomials on noncompact semialgebraic sets via KKT ideals*, Journal of Pure and Applied Algebra, 209 (2007), pp. 189–200.

[5] D. Grimm, T. Netzer, and M. Schweighofer, *A note on the representation of positive polynomials with structured sparsity*, Archiv der Mathematik, 89 (2007), pp. 399–403.

[6] D. Henrion and J.-B. Lasserre, *Detecting global optimality and extracting solutions in GloptiPoly*, in Positive polynomials in control, Springer, 2005, pp. 293–310.

[7] D. Henrion, J.-B. Lasserre, and J. Löfberg, *GloptiPoly 3: moments, optimization and semidefinite programming*, Optimization Methods & Software, 24 (2009), pp. 761–779.

[8] Z. Hua and Z. Qu, *On the exactness of Lasserre's relaxation for polynomial optimization with equality constraints*, arXiv preprint arXiv:2110.13766, (2021).

[9] M. Kojima and M. Muramatsu, *A note on sparse SOS and SDP relaxations for polynomial optimization problems over symmetric cones*, Computational Optimization and Applications, 42 (2009), pp. 31–41.

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

[11] ———, *Convergent SDP-relaxations in polynomial optimization with sparsity*, SIAM Journal on Optimization, 17 (2006), pp. 822–843.

[12] J. B. Lasserre, *An introduction to polynomial and semi-algebraic optimization*, vol. 52, Cambridge University Press, 2015.

[13] J. B. Lasserre, *The moment-SOS hierarchy*, in Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018, World Scientific, 2018, pp. 3773–3794.

[14] M. Laurent, *Sums of squares, moment matrices and optimization over polynomials*, in Emerging applications of algebraic geometry, Springer, 2009, pp. 157–270.

[15] V. Magron and J. Wang, *TSSOS: a Julia library to exploit sparsity for large-scale polynomial optimization*, arXiv preprint arXiv:2103.00915, (2021).

[16] ———, *Sparse polynomial optimization: theory and practice*, World Scientific, 2023.

[17] M. Newton and A. Papachristodoulou, *Sparse polynomial optimisation for neural network verification*, arXiv preprint arXiv:2202.02241, (2022).

[18] J. Nie, *Certifying convergence of lasserre's hierarchy via flat truncation*, Mathematical Programming, 142 (2013), pp. 485–510.

[19] ———, *Polynomial optimization with real varieties*, SIAM Journal on Optimization, 23 (2013), pp. 1634–1646.

[20] ———, *The $\mathcal{A}$ truncated K-moment problem*, Foundations of Computational Mathematics, 14 (2014), pp. 1243–1276.

[21] ———, *Optimality conditions and finite convergence of Lasserre's hierarchy*, Mathematical Programming, 146 (2014), pp. 97–121.

[22] ———, *Linear optimization with cones of moments and nonnegative polynomials*, Mathematical Programming, 153 (2015), pp. 247–274.

[23] ———, *Tight relaxations for polynomial optimization and Lagrange multiplier expressions*, Mathematical Programming, 178 (2019), pp. 1–37.

[24] ———, *Moment and Polynomial Optimization*, SIAM, 2023.

[25] J. Nie and J. Demmel, *Sparse sos relaxations for minimizing functions that are summations of small polynomials*, SIAM Journal on Optimization, 19 (2009), pp. 1534–1558.

[26] J. Nie, J. Demmel, and B. Sturmfels, *Minimizing polynomials via sum of squares over the gradient ideal*, Mathematical Programming, 106 (2006), pp. 587–606.

[27] J. Nie and X. Tang, *Nash equilibrium problems of polynomials*, arXiv preprint arXiv:2006.09490, (2020).

[28] ———, *Convex generalized Nash equilibrium problems and polynomial optimization*, Mathematical Programming, (2021).

[29] J. Nie, L. Wang, J. J. Ye, and S. Zhong, *A Lagrange multiplier expression method for bilevel polynomial optimization*, SIAM Journal on Optimization, 31 (2021), pp. 2368–2395.

[30] J. Nie, Z. Yang, and X. Zhang, *A complete semidefinite algorithm for detecting copositive matrices and tensors*, SIAM Journal on Optimization, 28 (2018), pp. 2902–2921.
[31] J. Nie, Z. Yang, and G. Zhou, *The saddle point problem of polynomials*, Foundations of Computational Mathematics, (2021), pp. 1–37.
[32] M. Putinar, *Positive polynomials on compact semi-algebraic sets*, Indiana University Mathematics Journal, 42 (1993), pp. 969–984.
[33] B. Reznick, *Some concrete aspects of Hilbert's 17th problem*, Contemporary mathematics, 253 (2000), pp. 251–272.
[34] H. Waki, S. Kim, M. Kojima, and M. Muramatsu, *Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity*, SIAM Journal on Optimization, 17 (2006), pp. 218–242.
[35] J. Wang and V. Magron, *Certifying global optimality of AC-OPF solutions via the CS-TSSOS hierarchy*, arXiv preprint arXiv:2109.10005, (2021).
[36] J. Wang, V. Magron, and J.-B. Lasserre, *TSSOS: A Moment-SOS hierarchy that exploits term sparsity*, SIAM Journal on Optimization, 31 (2021), pp. 30–58.
[37] J. Wang, V. Magron, J. B. Lasserre, and N. H. A. Mai, *CS-TSSOS: Correlative and term sparsity for large-scale polynomial optimization*, ACM Transactions on Mathematical Software, 48 (2022), pp. 1–26.