Exploiting Constant Trace Property in Large-scale Polynomial Optimization

NGOC HOANG ANH MAI, Laboratory for Analysis and Architecture of Systems, France
J. B. LASSERRE and VICTOR MAGRON, Laboratory for Analysis and Architecture of Systems, Université de Toulouse, France
JIE WANG, Academy of Mathematics and Systems Science, China

We prove that every semidefinite moment relaxation of a polynomial optimization problem (POP) with a ball constraint can be reformulated as a semidefinite program involving a matrix with constant trace property (CTP). As a result, such moment relaxations can be solved efficiently by first-order methods that exploit CTP, e.g., the conditional gradient-based augmented Lagrangian method. We also extend this CTP-exploiting framework to large-scale POPs with different sparsity structures. The efficiency and scalability of our framework are illustrated on some moment relaxations for various randomly generated POPs, especially second-order moment relaxations for quadratically constrained quadratic programs.

CCS Concepts: • Computing methodologies → Optimization algorithms; Algebraical algorithms; • Mathematics of computing → Semidefinite programming;

Additional Key Words and Phrases: Polynomial optimization, moment-SOS hierarchy, conditional gradient-based augmented Lagrangian, constant trace property

ACM Reference format:
Ngoc Hoang Anh Mai, J. B. Lasserre, Victor Magron, and Jie Wang. 2022. Exploiting Constant Trace Property in Large-scale Polynomial Optimization. ACM Trans. Math. Softw. 48, 4, Article 40 (December 2022), 39 pages. https://doi.org/10.1145/3555309

1 INTRODUCTION

This article is in the line of recent efforts to promote first-order methods as a viable alternative to interior-point methods for solving large-scale conic optimization problems, in particular, large-scale semidefinite programming (SDP) relaxations of polynomial optimization problems.
(POPs). We show that a wide class of POPs have a nice property, namely, the constant trace property (CTP), and that this property can be exploited in combination with first-order methods to solve large-scale SDP relaxations associated with a POP. So far, this property has been exploited only in a few cases, the most prominent examples being the Shor relaxation of Max-Cut [49], in which the authors are able to handle SDP matrices of huge size, and equality-constrained POPs on the sphere [30].

Given polynomials \( f, g_i, h_j \), let us consider the following POP with \( n \) variables, \( m \) inequality constraints and \( l \) equality constraints:

\[
    f^* := \min \{ f(x) : \ g_i(x) \geq 0, \ i \in [m], \ h_j(x) = 0, \ j \in [l] \},
\]

where \([m] := \{1, \ldots, m\}\) and \([l] := \{1, \ldots, l\}\). In general POP (1.1) is non-convex, NP-hard. It is well known that under mild conditions, the optimal value \( f^* \) of POP (1.1) can be approximated as closely as desired by the so-called Moment-Sums of squares (Moment-SOS) hierarchy [24]. There are a lot of important applications of POP (1.1) as well as the Moment-SOS hierarchy; the interested reader is referred to the monograph [17] and references therein.

**Computational cost of moment relaxations.** The \( k \)th order moment relaxation for POP (1.1) can be rewritten in compact form as the following standard SDP:

\[
    \tau = \inf_{X \in S^+} \{ \langle C, X \rangle : \langle A_j, X \rangle = b_j, \ j \in [\zeta] \},
\]

where \( S^+ \) is the set of positive semidefinite (psd) matrices in a block diagonal form: \( X = \text{diag}(X_1, \ldots, X_\omega) \) with \( X_j \) being a block of size \( s(j) \), \( j \in [\omega] \) and \( \zeta \) is the number of affine constraints. We denote the largest block size by \( s^{\max} := \max_{j \in [\omega]} s(j) \).

Note that SDP-relaxation (1.2) of POP (1.1) at step \( k \) of the Moment-SOS hierarchy has \( \omega = m + 1 \) blocks whose largest size is \( s^{\max} \equiv \binom{n+k}{n} \) while the number of affine constraints is \( \zeta = \mathcal{O} \left( \binom{n+k}{n}^2 \right) \).

Thus, the computational cost for solving SDP (1.2) grows very rapidly with \( k \).

We say that SDP (1.2) has constant trace property (CTP) if there exists a positive real number \( a \) such that \( \text{trace}(X) = a \), for all feasible solution \( X \) of SDP (1.2). We also say that POP (1.1) has CTP when every moment relaxation of POP (1.1) has CTP.

Table 1 lists several available methods for solving SDP (1.2). In particular, observe that two of them, CGAL and SBM, are first-order methods that exploit CTP. In Reference [49], the authors combined CGAL with the Nyström sketch (named SketchyCGAL), which requires dramatically less storage than other methods and is very efficient for solving Shor’s relaxation of large-scale Max-Cut instances.

Fortunately, it is usually possible to reduce the size of this SDP relaxation by exploiting certain structures of POP (1.1). Table 2 lists some of these structures.

- **Correlative sparsity (CS), term sparsity (TS), and their combination (CS-TS)** are applied to POP (1.1) in case that the data \( f, g_i, h_j \) are sparse polynomials. The main idea of CS, TS, and CS-TS is to break the moment matrices and localizing matrices (which are the psd matrices in the moment relaxation) into a lot of blocks according to certain sparsity patterns derived from the POP. If the largest block size is relatively small (say, \( s^{\max} \leq 100 \)), then the corresponding SDP can be solved efficiently. But if the largest block size is still large (say, \( s^{\max} \geq 200 \)), then the corresponding SDP remains hard to solve.

- An alternative approach to handle large SDP relaxations arising from POPs is to work with sums of squares directly, as in References [33, 34]. This approach combines non-symmetric conic optimization with polynomial interpolation and happens to be particularly efficient for problems involving polynomials of high degree and few variables (say, \( n \leq 3 \)). Solving the
Table 1. Complexity Comparison (in Terms of Floating-point Operations) of Several Methods for Solving SDP

| Method                        | Software               | SDP type                      | Convergence rate | The most expensive part per iteration |
|-------------------------------|------------------------|-------------------------------|------------------|--------------------------------------|
| IP [16] (second-order)        | SDPT3 [35], Mosek [1]  | Arbitrary                     | \(O(\log(1/\varepsilon))\) [37] | System of linear equations solving with complexity \(O(\varepsilon^{max})\) [36, Table 1] |
| IP with nonsymmetric cone [33] (second-order) | alfonso [34]          | Arbitrary                     | \(O(\log(1/\varepsilon))\) | System of linear equations solving with complexity \(O(\eta^3)\) |
| ADMM [2] (first-order)        | SCS [32], COSMO [10]  | Arbitrary                     | \(O(\varepsilon^{-1})\) [18] | Positive definite system of linear equations solving by LDL\(^T\)-decomposition with complexity \(O(\varepsilon^{max})\) |
| SBM [15] (first-order)        | ConicBundle [14]      | with CTP                      | \(O(\log(1/\varepsilon)/\varepsilon)\) [8] | Positive definite linear system solving with complexity \(O(\varepsilon^{max})\) |
| CGAL [48] (first-order)       | SketchyCGAL [49]      | with CTP                      | \(O(\varepsilon^{-2})\) | Smallest eigenvalue computing by the Arnoldi iteration with complexity \(O(s^{max})\) [28] |

Table 2. Several Special Structures for Reducing Complexity of the Moment-SOS Relaxations

| Structure                         | Software      | POP type                                      |
|-----------------------------------|---------------|-----------------------------------------------|
| CS [25, 39]                       | SparsePOP [40] | \(f = \sum_{j \in [p]} f_j\) and \(f_j, (g_i)_{i \in J}, (h_i)_{i \in W_j}\) share the same variables for every \(j \in [p]\) and \(p > 1\) |
| TS [42, 45]                       | TSSOS         | \(f, g, h_j\) involve a few of terms          |
| CS-TS [46]                        | TSSOS         | Both CS and TS hold                           |
| Nonsymmetric conic optimization [33] | alfonso [34], Hypatia [7] | A small number of variables and possibly high degree |
| CTP [30]                          | SpectralPOP   | Equality constrained POPs on a sphere \((m = 0\) and \(h_1 := R - \|x\|^2_2\)) |

\(k\)th order moment relaxation (1.2) with the related primal-dual interior-point algorithm has a \(O((n+k)^{1/2})\) iteration complexity and a \(O((n+k)^{1/2}(n+2k)^3)\) time complexity. This method seems practically less efficient than the standard interior-point method when \(n\) increases, as shown in Reference [33, Table 7]. It is in contrast with sparsity exploiting techniques (CS and CS-TS), which is able to deal with POPs involving a very large number of variables.

- In the previous work [30], the first three authors exploited CTP for equality-constrained POPs on a sphere and converted the resulting SDP relaxations to spectral minimization problems that could be solved efficiently by Limited Memory Bundle Method (LMBM) of Karmitsa and Mäkelä in Reference [21]. This method can return approximate optimal values of SDP relaxations involving \(2,000 \times 2,000\) matrices for which Mosek encounters memory...
issues and SketchyCGAL is much less efficient. Importantly, the moment SDP-relaxation of an equality-constrained POP has a single psd matrix. In contrast, for a POP involving a ball constraint (with possibly other inequality constraints), the resulting moment SDP-relaxation includes several psd matrices. Unfortunately for such SDPs, LMBM usually returns inaccurate values even when CTP holds because of ill-conditioning issues. LMBM only updates the dual variables, and so it is hard to ensure that the KKT conditions hold. We can overcome the latter ill-conditioning issues by relying on a primal-dual algorithm such as CGAL. It turns out that CGAL (without sketching) is suitable for this type of SDPs. For an SDP involving a single matrix, SketchyCGAL stores updated matrices by means of Nyström sketch. In our experimental setting, we rather consider CGAL without sketching, which boils down to relying on implicit updated matrices. It turns out that this strategy is much faster than the one based on Nyström sketch, but does not provide the primal (matrix) solution.

**SDP relaxations of non-convex quadratically constrained quadratic programs.** A non-convex quadratically constrained quadratic program (QCQP) is a special instance of POP (1.1) for which the degree of the input polynomials is at most two. Famous instances of non-convex QCQPs include the Max-Cut problem and the optimal power flow (OPF) problem [20]; in addition, we recall that linearly constrained quadratic programs have an equivalent Max-Cut formulation [27]. They also have applications in deep learning, e.g., the computation of Lipschitz constants [6] and the stability analysis of recurrent neural networks [9]. On the one hand, local optima of non-convex QCQPs can be obtained by using local solvers; see, e.g., Reference [5]. On the other hand, the global optimum of QCQPs can be approximated as closely as desired by using moment (SDP) relaxations. In practice, non-convex QCQPs usually involve a large number of variables (say, $n \geq 1,000$) and their associated SDP relaxations (1.2) can be classified in two groups as follows:

- **The first-order relaxation:** $k = 1$ (also known as Shor’s relaxation in the literature). In this case the number of affine constraints in SDP (1.2) is typically not exceeding the largest block size, i.e., $\zeta \leq s_{\text{max}}$. It can be efficiently solved by most SDP solvers, in particular, with SketchyCGAL [49]. Nevertheless, the first-order relaxation may provide only a lower bound of the optimal value of POP (1.1). In this case, one needs to solve the second and perhaps even higher-order relaxations to obtain tighter bounds or the global optimum.

- **The second and higher-order relaxations:** $k \geq 2$. In this case, the number of affine constraints in SDP (1.2) is typically much larger than the largest block size ($\zeta \gg s_{\text{max}}$). Then, unfortunately, most SDP solvers cannot handle large-scale SDPs of this form. In our previous work [30], we proposed a remedy for the particular case of the second-order SDP relaxations of equality-constrained POPs on a sphere by relying on first-order solvers such as LMBM.

**Common issues of solving large-scale SDP relaxations.** When solving the second and higher-order SDP relaxations, SDP solvers often encounter the following issues:

- **Storage:** Interior-point methods are often chosen by users because of their highly accurate output. These methods are efficient for solving medium-scale SDPs. However, they frequently fail due to lack of memory when solving large-scale SDPs (say, $s_{\text{max}} > 500$ and $\zeta > 2 \times 10^5$ on a standard laptop). First-order methods (e.g., ADMM, SBM, CGAL) provide an alternative to interior-point methods to avoid the memory issue. This is due to the fact that the cost per iteration of first-order methods is much lower than that of interior-point methods. At the price of losing convexity, one can also rely on heuristic methods and replace the full matrix $X$ in SDP (1.2) by a simpler one to save memory. For instance, the Burer-Monteiro method [3] considers a low-rank factorization of $X$. However, to get correct results the
rank cannot be too low [41] and therefore this limitation makes it useless for the second and higher-order relaxations of POPs. Not suffering from such a limitation, CGAL not only maintains the convexity of SDP (1.2) but also possibly runs with implicit matrix $X$, as described in Remarks A.12 and A.17.

- **Accuracy**: Nevertheless, first-order methods have low convergence rates compared to the interior-point methods. Their performance depends heavily on the problem scaling and conditioning. As a result, in solving large-scale SDPs with first-order methods it is often difficult to obtain numerical results with high accuracy. Therefore, we do not expect the relative gap of the approximate value ($\text{val}_{\text{approx}}$) returned by first-order SDP solvers w.r.t. the exact value ($\text{val}_{\text{exact}}$), defined by

$$\frac{|\text{val}_{\text{approx}} - \text{val}_{\text{exact}}|}{|\text{val}_{\text{exact}}|},$$

(1.3)

to be smaller than $10^{-8}$ (as for interior-point methods) but at least to be smaller than 1%.

The goal of this article is to provide a method that returns the optimal value of the second-order moment SDP-relaxation and that is suitable for a class of large-scale non-convex QCQPs with CTP. Ideally (i) it should avoid the memory issue, and (ii) the resulting relative gap of the approximate value returned by this method w.r.t. the exact value should be less than 1%.

**Contribution.** We show that (i) a large class of POPs have the constant trace property and (ii) this property can be exploited for solving their associated semidefinite relaxations via appropriate first-order methods. More precisely, our contribution is threefold:

1. In Section 3.2, we show that if a positive real number belongs to the interior of every truncated quadratic module associated with the inequality constraints, which is defined later in (2.7), then the corresponding POP has CTP. Moreover, we prove that this condition always holds when a ball constraint is present.

2. In Section 3.3, we provide a numerical procedure to check whether a POP has CTP. It consists in solving a certain linear program (LP) of the form:

$$\inf_{y \in \mathbb{R}^q_+} \{ c^T y : Ay = b \},$$

(1.4)

where $\mathbb{R}_+ := [0, \infty)$, $c \in \mathbb{R}^q$, $A \in \mathbb{R}^{p \times q}$ and $b \in \mathbb{R}^p$. With this approach, we prove in Section 3.4 that several special classes of POPs (including POPs on a ball, annulus, simplex) have CTP.

3. Our final contribution, postponed in Appendices A.1 and A.2, is to handle sparse large-scale POPs by integrating sparsity-exploiting techniques into the CTP-exploiting framework.

For practical implementation, we provide a software library called ctpPOP. It models each moment SDP-relaxation of POPs as a standard SDP with CTP and then solves this SDP by CGAL or a spectral method (SM), based on nonsmooth optimization solvers (LMBM [21] or the Proximal Bundle Method [22]).

In Section 4 and Appendix A.3, we provide extensive numerical experiments to illustrate the efficiency and scalability of ctpPOP with the CGAL solver. In all our randomly generated POPs with different sparsity structures, the relative gap of the optimal value provided by CGAL w.r.t. the optimal value provided by Mosek is below 1%. Because of its cheap cost per iteration, CGAL is more suitable for particularly SDPs of form (1.2) with $\zeta = O(s^{\max}_\theta^2)$ (such as the second-order moment SDP-relaxations of POPs) than other solvers (e.g., COSMO).

For instance, for minimizing a dense quadratic polynomial on the unit ball with 100 variables, CGAL returns the optimal value of the second-order moment SDP relaxation within six hours on
a standard laptop while Mosek (considered as the state-of-the-art SDP solver using interior-point methods) runs out of memory. Similarly, for minimizing a sparse quadratic polynomial involving a thousand variables with a ball constraint on each clique of variables, CGAL spends around two thousand seconds to solve the second-order moment SDP-relaxation, while Mosek runs again out of memory. The largest clique of this POP involves 42 variables.

The classical OPF problem without constraints on current magnitudes (as in References [11, 19]) can be formulated as a POP with ball and annulus constraints. For many instances Shor’s relaxation provides the global optimum. However, for illustration purposes, we have compared CGAL and Mosek for solving the second-order CS-TS relaxation of the “case89_pegase_api” instance from the PGLib-OPF database. The largest block size and the number of equality constraints of this SDP are around 1.7 thousand and 8 million, respectively. While Mosek failed because of a memory issue, CGAL still returned the optimal value in two days, and with relative gap w.r.t. a local optimal value being less than 0.6%.

2 NOTATION AND PRELIMINARY RESULTS

With \( x = (x_1, \ldots, x_n) \), let \( \mathbb{R}[x] \) stand for the ring of real polynomials and let \( \Sigma[x] \subseteq \mathbb{R}[x] \) be the subset of sum of squares (SOS) polynomials. Their restrictions to polynomials of degree at most \( d \) and \( 2d \) are denoted by \( \mathbb{R}[x]_d \) and \( \Sigma[x]_d \), respectively. For \( \alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n \), let \( |\alpha| := \alpha_1 + \cdots + \alpha_n \). For \( d \in \mathbb{N} \), let \( \mathbb{N}^d := \{ \alpha \in \mathbb{N}^n : |\alpha| \leq d \} \) and \( \mathbb{N}^+ := \{ u \in \mathbb{N}^d : u \neq d \} \), for \( * \in \{\geq, \leq, >, <\} \). For \( n \in \mathbb{N}^+ \), let \( [n] := \{1, 2, \ldots, n\} \).

Let \( (x^\alpha)_{\alpha \in \mathbb{N}^n} \) be the canonical monomial basis of \( \mathbb{R}[x] \) (sorted w.r.t. the graded lexicographic order) and \( \nu_d(x) \) be the vector of monomials of degree up to \( d \), with length \( s(n, d) := \binom{n+d}{n} \). When it is clear from the context, we also write \( s(d) \) instead of \( s(n, d) \). A polynomial \( p \in \mathbb{R}[x]_d \) can be written as \( p(x) = \sum_{\alpha \in \mathbb{N}^d} p_\alpha x^\alpha = \mathbf{p}^\top \nu_d(x) \), where \( \mathbf{p} = (p_\alpha) \in \mathbb{R}^{s(d)} \) is the vector of coefficients in the canonical monomial basis. For \( p \in \mathbb{R}[x]_d \), let \( |p| := \deg(p)/2 \). The \( l_1 \)-norm of a polynomial \( p \) is given by the \( l_1 \)-norm of the vector of coefficients \( \mathbf{p} \), that is \( \|\mathbf{p}\|_1 := \sum |p|_\alpha \). Given \( a \in \mathbb{R}^n \), the \( l_2 \)-norm of \( a \) is \( \|a\|_2 := (a_1^2 + \cdots + a_n^2)^{1/2} \) and the maximum norm of \( a \) is \( \|a\|_{\infty} := \max |a_j| : j \in [n] \).

Given a subset \( S \) of real symmetric matrices, let \( S^+ := \{x \in S : x \geq 0\} \).

Polynomial optimization problem. A polynomial optimization problem (POP) is defined as

\[
\min f(x) \quad : \quad x \in S(g) \cap V(h),
\]

where \( S(g) \) and \( V(h) \) are a basic semialgebraic set and a real variety defined, respectively, by

\[
S(g) := \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i \in [m] \}
\]

\[
V(h) := \{ x \in \mathbb{R}^n : h_j(x) = 0, \ j \in [l] \}
\]

for some polynomials \( f, g_i, h_j \in \mathbb{R}[x] \) with \( g := \{g_i\}_{i \in [m]}, h := \{h_j\}_{j \in [l]} \). We will assume that POP (2.5) has at least one global minimizer.

Riesz linear functional. Given a real-valued sequence \( y = (y_\alpha)_{\alpha \in \mathbb{N}^n} \), define the Riesz linear functional \( L_y : \mathbb{R}[x] \to \mathbb{R}, f \mapsto L_y(f) := \sum_\alpha f_\alpha y_\alpha \). Let \( d \) be a positive integer. A real infinite (respectively, finite) sequence \( y_\alpha \) (respectively, \( y_\alpha \)) has a representing measure if there exists a finite Borel measure \( \mu \) such that \( y_\alpha = \int x^\alpha d\mu(x) \) for every \( \alpha \in \mathbb{N}^n \) (respectively, \( \alpha \in \mathbb{N}^n \)). In this case, \( y_\alpha \) is called the moment sequence of \( \mu \). We denote by \( \text{supp}(\mu) \) the support of a Borel measure \( \mu \).

\(^1\)https://github.com/power-grid-lib/pglib-opf.
Moment/Localizing matrix. The moment matrix of order \(d\) associated with a real-valued sequence \(y = (y_{\alpha})_{\alpha \in \mathbb{N}^n}\) and \(d \in \mathbb{N}^{>0}\), is the real symmetric matrix \(M_d(y)\) of size \(s(d)\), with entries \((y_{\alpha + \beta})_{\alpha, \beta \in \mathbb{N}^n}\). The localizing matrix of order \(d\) associated with \(y = (y_{\alpha})_{\alpha \in \mathbb{N}^n}\) and \(p = \sum y_{\alpha}x^\alpha \in \mathbb{R}[x]\), is the real symmetric matrix \(M_d(y)\) of size \(s(d)\) with entries \((\sum y_{\alpha + \beta}x_{\alpha + \beta})_{\alpha, \beta \in \mathbb{N}^n}\).

Quadratic module. Given \(g = \{g_i : i \in [m]\} \subseteq \mathbb{R}[x]\), the quadratic module associated with \(g\) is defined by \(Q(g) := \{k \geq 0 : \sigma_0 + \sum_{i \in [m]} \sigma_i g_i = 0, \sigma_i \in \Sigma(x), \sigma_0 \in \Sigma(x)\}\), and for a positive integer \(k\), the set

\[
Q_k(g) := \{k \geq 0 : \sigma_0 + \sum_{i \in [m]} \sigma_i g_i = 0, \sigma_0 \in \Sigma(x)_k, \sigma_i \in \Sigma(x)_{k-[g_i]}\}
\]

(2.7)

is the truncation of \(Q(g)\) of order \(k\).

Ideal. Given \(h = \{h_i : i \in [l]\} \subseteq \mathbb{R}[x]\), the set \(I(h) := \{\sum_{j \in [l]} \psi_j h_j : \psi_j \in \mathbb{R}[x]\}\) is the ideal generated by \(h\), and the set \(I_k(h) := \{\sum_{j \in [l]} \psi_j h_j : \psi_j \in \mathbb{R}[x]_{2(k-[h_j])}\}\) is the truncation of \(I(h)\) of order \(k\).

Archimedeanity. Assume that there exists \(R > 0\) such that \(R - \|x\|_2^2 \in Q(g) + I(h)\). As a consequence, \(S(g) \cap V(h) \subseteq B_R\), where \(B_R := \{x \in \mathbb{R}^n : \|x\|_2 \leq \sqrt{R}\}\). In this case, we say that \(Q(g) + I(h)\) is Archimedean [26].

The Moment-SOS hierarchy [24]. Given a POP \((Ax + b)\), consider the following associated hierarchy of SOS relaxations indexed by \(k \in \mathbb{N}^{\geq k_{\min}}\) with \(k_{\min} := \max\{\|f\|, \|g_i\|\} \in [m], \|h_i\| \in [l]\}:

\[
\rho_k := \sup \{\xi \in \mathbb{R} : f - \xi \in Q_k(g) + I_k(h)\}.\]

(2.8)

For each \(\sigma \in \Sigma(x)_d\), there exists \(G \geq 0\) such that \(\sigma = v_d^T G v_d\). Thus, for each \(k \in \mathbb{N}^{\geq k_{\min}}, (2.8)\) can be rewritten as an SDP:

\[
\rho_k = \sup_{\xi, G, v_d} \left\{ \xi : G_i \geq 0, f - \xi = v_d^T G v_d + \sum_{i \in [m]} g_i v_d^T G v_k-[g_i] + \sum_{j \in [l]} h_j v_d^T G v_{2(k-[h_j])} u_j \right\}.\]

(2.9)

For every \(k \in \mathbb{N}^{\geq k_{\min}}\), the dual of (2.9) reads as

\[
\tau_k := \inf_{y \in \mathbb{R}^{I(k)}} \left\{ L_y(f) : \begin{array}{l}
M_k(y) \geq 1, y_0 = 1 \\
M_k-[g_i](g_i y) \geq 0, i \in [m] \\
M_k-[h_j](h_j y) = 0, j \in [l] 
\end{array} \right\}.\]

(2.10)

If \(Q(g) + V(h)\) is Archimedean, then both \((\rho_k)_{k \in \mathbb{N}^{\geq k_{\min}}}\) and \((\tau_k)_{k \in \mathbb{N}^{\geq k_{\min}}}\) converge to \(f^*\). For more details on the Moment-SOS hierarchy and its various applications, the interested reader is referred to Reference [26].

3 EXPLOITING CTP FOR DENSE POPS

This section is devoted to developing a framework to exploit CTP for dense POPs. We provide a sufficient condition for a POP to have CTP, as well as a series of linear programs to check whether the sufficient condition holds. In addition, we show that several special classes of POPs have CTP.

3.1 CTP for Dense POPs

First, let us define CTP for a POP. To simplify notation, for every \(k \in \mathbb{N}^{\geq k_{\min}}\), denote by \(S_k\) the set of real symmetric matrices

- of size \(s_k := s(k) + \sum_{i \in [m]} s(k-[g_i])\),
- in a block diagonal form \(X = \text{diag}(X_0, \ldots, X_m)\), and such that
- \(X_0\) (respectively, \(X_i\)) is of size \(s(k)\) (respectively, \(s(k-[g_i])\)) for \(i \in [m]\).
Letting \( D_k(y) := \text{diag}(M_k(y), M_{k-[g_1]}(g_1y), \ldots, M_{k-[g_m]}(g_my)) \), SDP (2.10) can be rewritten in the form:

\[
\tau_k := \inf_{y \in \mathbb{R}^{(2k)}} \left\{ L_y(f) \mid D_k(y) \in S_k^+, y_0 = 1, M_{k-[h_i]}(h_iy) = 0, i \in [I] \right\}.
\] (3.11)

Definition 3.1 (CTP for a POP). We say that POP (2.5) has CTP if for every \( k \in \mathbb{N}^{\geq k_{\min}} \), there exists \( a_k > 0 \) and a positive definite matrix \( P_k \in S_k \) such that for all \( y \in \mathbb{R}^k \),

\[
M_{k-[h_i]}(h_iy) = 0, i \in [I], y_0 = 1 \quad \Rightarrow \quad \text{trace}(P_kD_k(y)P_k) = a_k.
\] (3.12)

In other words, we say that POP (2.5) has CTP if each moment relaxation (3.11) has an equivalent form involving a psd matrix whose trace is constant. In this case, we call \( a_k \) the constant trace and \( P_k \) the basis transformation matrix. In the next subsection, we provide a sufficient condition for POP (2.5) to have CTP.

Example 3.2 (CTP for Equality-constrained POPs on a Sphere [30]). If \( g = \emptyset \) and \( h_1 = R - \| \cdot \|_2^2 \) for some \( R > 0 \), then POP (2.5) has CTP with \( \alpha_k = (R + 1)^k \) and \( P_k := \text{diag}(\alpha_k^{-1/2}) \), where \((\theta_k, \alpha)_{\alpha \in \mathbb{N}^n_k} \subseteq \mathbb{R}^n_{\geq 0} \) satisfies \( (1 + \| \cdot \|_2^2)^k = \sum_{\alpha \in \mathbb{N}^n_k} \theta_{k, \alpha} x^{2\alpha} \), for all \( k \in \mathbb{N}^{\geq k_{\min}} \).

We now provide a general method to solve a POP with CTP. We first convert the kth order moment relaxation (3.11) of this POP to a standard primal SDP problem with CTP and then leverage appropriate first-order algorithms that exploit CTP to solve the resulting SDP problem.

Suppose POP (2.5) has CTP. For every \( k \in \mathbb{N}^{\geq k_{\min}} \), letting \( X = P_k D_k(y) P_k \), (3.11) can be rewritten as

\[
\tau_k = \inf_{X \in S_k^+} \{ (C_k, X) : A_k X = b_k \},
\] (3.13)

where \( A_k : S_k \to \mathbb{R}^{\xi_k} \) is a linear operator such that \( A_k X = ((A_{k,1}, X), \ldots, (A_{k,\xi_k}, X)) \) with \( A_{k,i} \in S_k, i \in [\xi_k], C_k \in S_k \) and \( b_k \in \mathbb{R}^{\xi_k} \). Appendix A.6.1 describes how to convert SDP (3.11) to the form (3.13).

The dual of SDP (3.13) reads

\[
\rho_k = \sup_{z \in \mathbb{R}^{\xi_k}} \{ b_k^\top z : A_k^\top z - C_k \in S_k^+ \},
\] (3.14)

where \( A_k^\top : \mathbb{R}^{\xi_k} \to S_k \) is the adjoint operator of \( A_k \), i.e., \( A_k^\top z = \sum_{i \in [\xi_k]} z_i A_{k,i} \).

After replacing \((A_k, A_{k,i}, b_k, C_k, S_k, \xi_k, s, \tau_k, \rho_k, a_k)\) by \((A, A_i, b, C, S, \xi, s, r, \rho, a)\), the primal-dual (3.13)–(3.14) has an equivalent formulation as the primal-dual (1.58)–(1.59); see also Appendix A.4.1 with \( \omega = m + 1 \) and \( s_{\max} = s(k) \).

Then two first-order algorithms (CGAL and SM) are leveraged for solving the primal-dual (1.58)–(1.59); see Appendix A.4.1 and Appendix A.5.1.

3.2 A Sufficient Condition for a POP to Have CTP

In this section, we provide a sufficient condition for POP (2.5) to have CTP.

For \( k \in \mathbb{N}^{\geq k_{\min}} \), let \( Q_k^\circ(g) \) be the interior of the truncated quadratic module \( Q_k(g) \), i.e., \( Q_k^\circ(g) := \{ v_k^\top G_0 v_k + \sum_{i \in [m]} g_i v_k^[i-g_i] G_i v_k^[i-g_i] : G_i > 0, i \in [0] \cup [m] \} \).

Theorem 3.3. The following statements hold:

1. If one of the following equivalent conditions hold for all \( k \in \mathbb{N}^{\geq k_{\min}} \):

\[
\mathbb{R}^{\geq 0} \subseteq Q_k^\circ(g) + I_k(h) \quad \Leftrightarrow \quad \forall \delta > 0, \delta \in Q_k^\circ(g) + I_k(h)
\]

\[
\Leftrightarrow \quad 1 \in Q_k^\circ(g) + I_k(h),
\] (3.15)

then POP (2.5) has CTP, as in Definition 3.1.
(2) Assume that $h = \emptyset$ and $S(g)$ has nonempty interior. Then POP (2.5) has CTP if and only if
\[ \mathbb{R}^{>0} \subseteq Q^o_S(g) \land k \in \mathbb{N}^{\geq k_{\text{min}}} . \]

**Proof.** (1) Let $k \in \mathbb{N}^{\geq k_{\text{min}}}$ and assume that $\mathbb{R}^{>0} \subseteq Q^o_S(g) + I_k(h)$. Then there exists $a_k > 0$ such that
\[ a_k = v_k^T G_k v_k + \sum_{i \in [m]} g_i v_k^T [g_i] G_i v_k - [g_i] + \sum_{j \in [l]} h_j v_{2(k-[h_j])}^T u_j , \]
for some $G_i > 0$, $i \in \{0\} \cup [m]$ and real vector $u_j$, $j \in [l]$. We denote by $G_i^{1/2}$ the square root of $G_i$, $i \in \{0\} \cup [m]$. Then $G_i^{1/2}$ is well-defined and $G_i^{1/2} > 0$. Set $P_k = \text{diag}(G_0^{1/2}, \ldots, G_m^{1/2})$. Let $y \in \mathbb{R}^{s(2k)}$ such that $M_k(h_j y) = 0$, $j \in [l]$, and $y_0 = 1$. Then
\[ L_y \left( \sum_{j \in [l]} h_j v_{2(k-[h_j])}^T u_j \right) = \sum_{j \in [l]} \sum_{\alpha \in \mathbb{N}^{\geq k-[h_j]}} u_j, \alpha L_y(h_j x^\alpha) = 0 . \]
From this and (3.17),
\[ a_k = L_y \left( v_k^T G_k v_k + \sum_{i \in [m]} g_i v_k^T [g_i] G_i v_k - [g_i] \right) = \text{trace}(M_k(y) G_k) + \sum_{i \in [m]} \text{trace}(M_k(y) G_i) \]
\[ = \text{trace} \left( G_0^{1/2} M_k(y) G_0^{1/2} \right) + \sum_{i \in [m]} \text{trace} \left( G_i^{1/2} M_k(y) G_i^{1/2} \right) = \text{trace}(P_k D_k(y) P_k), \]
yielding the first statement.

(2) The “if” part comes from the first statement. Let us prove the “only if” part. Assume that POP (2.5) has CTP (Definition 3.1). Let $a \in S(g)$, $y = (y_{a})_{a \in \mathbb{N}^{\geq k_{\text{min}}}}$ be the moment sequence of the Dirac measure $\delta_{a}$. Let $k \in \mathbb{N}^{\geq k_{\text{min}}}$ be fixed. Since $P_k \in S_k$, $P_k = \text{diag}(W_0, \ldots, W_m)$. Then $W_i^k > 0$, $i \in \{0\} \cup [m]$, since $P_k > 0$. Let us define the polynomial $w := v_k^T W_k^o v_k + \sum_{i \in [m]} g_i v_k^T [g_i] W_i^o v_k - [g_i]$. By assumption,
\[ a_k = \text{trace} \left( P_k D_k(y) P_k \right) = \text{trace}(W_0 M_k(y) W_0) + \sum_{i \in [m]} \text{trace}(W_i M_k-1(g_i y) W_i) \]
\[ = \text{trace}(M_k(y) W_0^2) + \sum_{i \in [m]} \text{trace}(M_k-1(g_i y) W_i^2) = L_y \left( v_k^T W_0^2 v_k + \sum_{i \in [m]} g_i v_k^T [g_i] W_i^2 v_k - [g_i] \right) = \int_{\mathbb{R}^n} w d\delta_{a}(x) = w(a). \]
It implies that $w - a_k$ vanishes on $S(g)$. Since $S(g)$ has nonempty interior, $w = a_k$, yielding the second statement.

The following lemma will be used later on:

**Lemma 3.4.** Let $R > 0$. For all $k \in \mathbb{N}^{\geq 1}$, one has
\[ (R + 1)^k = (1 + \|x\|_2^2)^k + (R - \|x\|_2^2) \sum_{j=0}^{k-1} (R + 1)^j (1 + \|x\|_2^2)^{k-j-1} . \]

**Proof.** Let $k \in \mathbb{N}^{\geq 1}$. Letting $a = R + 1$ and $b = 1 + \|x\|_2^2$, the desired equality follows from
\[ a^k - b^k = (a - b) \sum_{j=0}^{k-1} a^j b^{k-1-j} . \]

The next result states that the sufficient condition in Theorem 3.3 holds whenever a ball constraint is present in the POP’s description. For a real symmetric matrix $A$, denote the largest eigenvalue of $A$ by $\lambda_{\text{max}}(A)$.
Theorem 3.5. If $R - ||x||_2^2 \in g$ for some $R > 0$, then the inclusions (3.16) hold and therefore POP (2.5) has CTP.

Proof. Without loss of generality, set $g_m := R - ||x||_2^2$ and let $k \in \mathbb{N}^{\geq k_{\min}}$ be fixed. By Lemma 3.4, $(R + 1)^k = \Theta + g_m \Lambda$, where $\Theta := (1 + ||x||_2^2)^k$ and $\Lambda := \sum_{j=0}^{k-1} (R + 1)^j (1 + ||x||_2^2)^{k-j-1}$. Note that

\begin{itemize}
  \item $\Theta = \sum_{a \in \mathbb{N}^n} \theta_a x^{2a} = v_k^T G_0 v_k$ for some $(\theta_a)_{a \in \mathbb{N}^n} \subseteq \mathbb{R}^>$;
  \item $\Lambda = \sum_{a \in \mathbb{N}^n} \lambda_a x^{2a} = v_k^T G_m v_{k-1}$ for some $(\lambda_a)_{a \in \mathbb{N}^n} \subseteq \mathbb{R}^>$.
\end{itemize}

Here, $G_0 = \text{diag}((\theta_a)_{a \in \mathbb{N}^n})$ and $G_m = \text{diag}((\lambda_a)_{a \in \mathbb{N}^n})$ are both positive definite. Then, we have $(R + 1)^k = v_k^T G_0 v_k + g_m v_{k-1}^T G_m v_{k-1}$. Denote by $I_t$ the identity matrix of size $s(t)$ for $t \in \mathbb{N}$.

Let $W$ be a real symmetric matrix such that $\sum_{i \in [m-1]} g_i v_{k-\lceil g_i \rceil}^T I_{k-\lceil g_i \rceil} v_{k-\lceil g_i \rceil} = v_k^T W v_k$. Since $G_0 > 0$, there exists $\delta > 0$ such that $G_0 - \delta W > 0$. Indeed, $G_0 - \delta W > 0 \iff I_k > \delta G_0^{1/2} W G_0^{-1/2} \iff 1 > \delta \max_{\mathbb{R}^>}(G_0^{1/2} W G_0^{-1/2})$, yielding the selection $\delta = 1/(\max_{\mathbb{R}^>}(G_0^{1/2} W G_0^{-1/2}) + 1)$. Then

$$(R + 1)^k = v_k^T (G_0 - \delta W) v_k + \delta \sum_{i \in [m-1]} g_i v_{k-\lceil g_i \rceil}^T I_{k-\lceil g_i \rceil} v_{k-\lceil g_i \rceil} + g_m v_{k-1}^T G_m v_{k-1},$$

which implies $(R + 1)^k \in Q_k^\circ(g)$, which in turn yields the desired conclusion. \hfill \Box

The next result is a consequence of Theorem 3.5. It states that if a POP has a ball constraint then the corresponding SOS relaxations satisfy Slater’s condition.

Corollary 3.6. Assume that $R - ||x||_2^2 \in g$ for some $R > 0$. Then Slater’s condition holds for SDP (2.9) for all $k \geq k_{\min}$. As a consequence, strong duality holds for the primal-dual (2.9)–(2.10) for all $k \geq k_{\min}$.

Proof. It suffices to prove that SDP (2.9) has a strictly feasible solution for all $k \geq k_{\min}$. Let $k \geq k_{\min}$ be fixed. By Reference [31, Proposition 5.8], there exist $\sigma_0 \in \Sigma[x]_k$, $\sigma \in \Sigma[x]_{k-1}$ and $\lambda \in \mathbb{R}$ such that $f + \lambda = \sigma_0 + (R - ||x||_2^2) \sigma$. Thus, $f + \lambda \in Q_k(g)$. By Theorem 3.5, $1 \in Q_k^\circ(g)$ and therefore $f + 1 + \lambda \in Q_k(g)$, which yields the desired conclusion. \hfill \Box

Remark 3.7. From the proofs of Theorems 3.5 and 3.3, the constant trace $a_k$ and the basis transformation matrix $P_k$ (Definition 3.1) can be taken as

$$a_k = (R + 1)^k \quad \text{and} \quad P_k = \text{diag}((G_0 - \delta W)^{1/2}, \sqrt{\delta} I_{k-\lceil g_1 \rceil}, \ldots, \sqrt{\delta} I_{k-\lceil g_{m-1} \rceil}, G_m^{1/2}).$$

However, this choice leads to poor numerical properties. In the next section, we provide a series of linear programs inspired from the inclusion in (3.15) to obtain a constant trace $a_k$ and a basis transformation matrix $P_k$ that achieve better numerical performance.

3.3 Verifying CTP for POPs by Solving Linear Programs

For any $k \in \mathbb{N}^{\geq k_{\min}}$, let $\hat{S}_k$ be the set of real diagonal matrices of size $s(k)$ and consider the following LP:

$$\inf_{\xi, G_i, u_j} \left\{ \xi \mid G_0 - I_0 \in \hat{S}_k^+, \ G_i - I_i \in \hat{S}_k^{\lceil g_i \rceil}, \ i \in [m], \right.$$  
$$\left. \xi = v_k^T G_0 v_k + \sum_{i \in [m]} g_i v_{k-\lceil g_i \rceil}^T I_{k-\lceil g_i \rceil} v_{k-\lceil g_i \rceil} + \sum_{j \in [l]} h_j v_{2(k-\lceil h_j \rceil)}^T u_j \right\},$$

where $I_i$ is the identity matrix for $i \in \{0\} \cup [m]$.

Lemma 3.8. If LP (3.21) has a feasible solution $(\xi_k, G_{i,k}, u_{j,k})$ for every $k \in \mathbb{N}^{\geq k_{\min}}$, then POP (2.5) has CTP with $a_k = \xi_k$ and $P_k = \text{diag}(G_{0,k}^{1/2}, \ldots, G_{m,k}^{1/2})$.
The proof of Lemma 3.8 is similar to that of Theorem 3.3 with $a_k = \xi_k$ and $G_i = G_{i,k}$, $i \in \{0\} \cup \{m\}$.

Since small constant traces are highly desirable for efficiency of first-order algorithms (e.g., CGAL), we search for an optimal solution of LP (3.21) instead of just a feasible solution.

**Remark 3.9.** One can extend the classes of diagonal matrices $\hat{S}_k$, $\hat{S}_{k-[g]}$ in (3.21) to obtain a smaller constant trace. For instance, one can define $\hat{S}_k$, $\hat{S}_{k-[g]}$ to be the class of symmetric block diagonal matrices with block size two. As shown in Reference [42, Lemma 4.3], (3.21) then becomes a second-order cone program that can be also efficiently solved.

### 3.4 Special Classes of POPs with CTP

In this section, we identify two classes of POPs whose CTP can be verified by LP (3.21).

For $I \subseteq [n]$, let $x(I) := \{x_j : j \in I\}$. For matrices $A$ and $B$ of the same size, the Hadamard product of $A$ and $B$, denoted by $A \odot B$, is the matrix with entries $[A \odot B]_{i,j} = A_{i,j}B_{i,j}$.

#### 3.4.1 POPs with Ball or Annulus Constraints on Subsets of Variables

Consider the following assumption on the inequality constraints of POP (2.5):

**Assumption 3.10.** There exists a nonnegative integer $r \leq m/2$ and

- $\overline{R}_j > R_j > 0$, $T_i \subseteq [n]$ for $i \in [r]$;
- $\overline{R}_j > 0$, $T_j \subseteq [n]$ for $j \in [m]\backslash[2r]$

such that

1. $(\cup_{i \in [r]} T_i) \cup (\cup_{j \in [m]\backslash[2r]} T_j) = [n]$;
2. $g_i := \|x(T_i)\|_2^2 - \overline{R}_i - \|x(T_i)\|_2^2$ for $i \in [r]$;
3. $g_j := \overline{R}_j - \|x(T_j)\|_2^2$ for $j \in [m]\backslash[2r]$.

Notice that if Assumption 3.10 holds, then POP (2.5) has $r$ annulus constraints and $(m-2r)$ ball constraints on subsets of variables. Moreover, $Q(g) + I(h)$ is Archimedean due to (1–3) in Assumption 3.10.

**Example 3.11.** Assumption 3.10 holds in the following cases:

1. $m = 1, r = 0$ and $g_1 := \overline{R}_1 - \|x\|_2^2$, i.e., $S(g)$ is a ball;
2. $m = n, r = 0$ and $g_i := \overline{R}_i - x_i^2$ for $i \in [n]$, i.e., $S(g)$ is a box;
3. $m = 2, r = 1$ and $g_1 := \|x\|_2^2 - \overline{R}_1, g_2 := \overline{R}_2 - \|x\|_2^2 (\overline{R}_1 > \overline{R}_2 > 0)$, i.e., $S(g)$ is an annulus.

**Proposition 3.12.** If Assumption 3.10 holds, then LP (3.21) has a feasible solution for every $k \in \mathbb{N}^{\geq m}$, and therefore POP (2.5) has CTP.

**Proof.** Let Assumption 3.10 hold. It is sufficient to show that (3.21) has a feasible solution for every $k \in \mathbb{N}^{\geq m}$.

Let $u = (u_j)_{j \in [n]} \subseteq \mathbb{N}^{\leq m}$ be defined by

$$u_j := \|x(T_j)\|_2^2 - \sum_{i \in [r]} \|x(T_i)\|_2^2 + \sum_{i \in [m]\backslash[2r]} \|x(T_i)\|_2^2. \quad (3.23)$$

Since $(\cup_{i \in [r]} T_i) \cup (\cup_{j \in [m]\backslash[2r]} T_j) = [n]$, one has $u_j \in \mathbb{N}^{\geq 1}, j \in [n]$. Moreover,
With $R := \sum_{i \in [r]} (R_i + \bar{R}_i) + \sum_{i \in [m] \setminus [2r]} \bar{R}_i$, by replacing $x$ by $u \circ x$ in Lemma 3.4, one obtains that for all $k \in \mathbb{N}^{\geq k_{\min}}$,

$$\begin{equation}
(R + 1)^k = (1 + \|u \circ x\|_2^2)^k + \Lambda_{k-1} \sum_{i \in [m]} \delta_i g_i, \tag{3.24}
\end{equation}$$

where $\Lambda_{k-1} := \sum_{j=0}^{k-1} (R + 1)^j (1 + \|u \circ x\|_2^2)^{k-j-1}$ and

$$\delta_i := \frac{R_i}{\bar{R}_i - R_i}, \delta_{i+r} := \frac{\bar{R}_i}{\bar{R}_i - R_i}, \text{ for } i \in [r], \text{ and } \delta_q = 1, q \in [m] \setminus [2r]. \tag{3.25}$$

It is due to the fact that

$$\begin{equation}
R - \|u \circ x\|_2^2 = \sum_{i \in [r]} (R_i + \bar{R}_i - \|x(T_i)\|_2^2) + \sum_{i \in [m] \setminus [2r]} (\bar{R}_i - \|x(T_i)\|_2^2), \tag{3.26}
\end{equation}$$

and $R_i + \bar{R}_i - \|x(T_i)\|_2^2 = \delta_i g_i + \delta_{i+r} g_{i+r}$, for all $i \in [r]$. For each $k \in \mathbb{N}^{\geq k_{\min}}$, let $(\theta_{k,\alpha})_{\alpha \in \mathbb{N}_k^m} \subseteq \mathbb{R}^{>0}$ and $(\eta_{k-1,\alpha})_{\alpha \in \mathbb{N}_k^{m-1}} \subseteq \mathbb{R}^{>0}$ be such that

$$\begin{equation}
(1 + \|u \circ x\|_2^2)^k = \sum_{\alpha \in \mathbb{N}_k^m} \theta_{k,\alpha} x^{2\alpha} \text{ and } \Lambda_{k-1} = \sum_{\alpha \in \mathbb{N}_k^{m-1}} \eta_{k-1,\alpha} x^{2\alpha}, \tag{3.27}
\end{equation}$$

and define the diagonal matrices

$$\begin{equation}
G_{k}^{(0)} := \text{diag}((\theta_{k,\alpha})_{\alpha \in \mathbb{N}_k^m}) \text{ and } G_{k-1}^{(i)} := \text{diag}((\delta_i \eta_{k-1,\alpha})_{\alpha \in \mathbb{N}_k^{m-1}}), i \in [m]. \tag{3.27}
\end{equation}$$

Then (3.24) yields that for every $k \in \mathbb{N}^{\geq k_{\min}}$,

$$\begin{equation}
(R + 1)^k = v_k^T G_{k}^{(0)} v_k + \sum_{i \in [m]} g_i v_k^T G_{k-1}^{(i)} v_{k-1}. \tag{3.27}
\end{equation}$$

Hence, $((R + 1)^k, G_{k}^{(i)}, 0)$ is a feasible solution of (3.21), for every $k \in \mathbb{N}^{\geq k_{\min}}$. \hfill \Box

3.4.2 POPs with Inequality Constraints of Equivalent Degree. We say that polynomials $p_1, \ldots, p_t$ are of equivalent degree if $[p_1] = \cdots = [p_t]$.

Assumption 3.13. Let $m \geq 3$ and $(g_j)_{j \in [m-2]}$ be of equivalent degree. $L > 0$ and $R > 0$ are such that $g_{m-1} = L - \sum_{j \in [m-2]} g_j$ and $g_m = R - \|x\|_2^2$.

Proposition 3.14. If Assumption 3.13 holds, then LP (3.21) has a feasible solution for every $k \in \mathbb{N}^{\geq k_{\min}}$, and therefore POP (2.5) has CTP.

Proof. Let Assumption 3.13 hold with $u := [g_i]$, $i \in [n + 1]$. For every $k \in \mathbb{N}^{\geq k_{\min}}$, letting $\Lambda_{k-1} := \sum_{j=0}^{k-1} (R + 1)^j (1 + \|x\|_2^2)^{k-j-1}$ and $\Theta_{t} := (1 + \|x\|_2^2)^{t}$, for $t \in \mathbb{N}$, Lemma 3.4 yields $\sum_{j=0}^{k-1} (R + 1)^j (1 + \|x\|_2^2)^{k-j-1}$. It implies that for every $k \in \mathbb{N}^{\geq k_{\min}}$,

$$\begin{equation}
(R + 1)^k = \left(\Theta_k - \frac{L}{L + 1} \Theta_{k-u}\right) + \frac{1}{L + 1} \Theta_{k-u} \sum_{i \in [m-1]} g_i + g_m \Lambda_{k-1}. \tag{3.28}
\end{equation}$$

It is due to the fact that $\sum_{i \in [m-1]} g_i = L$. For each $k \in \mathbb{N}^{\geq k_{\min}}$, let us consider the following sequences:

- $(v_{k,\alpha})_{\alpha \in \mathbb{N}_k^m} \subseteq \mathbb{R}^{>0}$ such that $\Theta_k - \frac{L}{L + 1} \Theta_{k-u} = \sum_{\alpha \in \mathbb{N}_k^m} v_{k,\alpha} x^{2\alpha}$;
- $(\theta_{k-u,\alpha})_{\alpha \in \mathbb{N}_k^{m-1}} \subseteq \mathbb{R}^{>0}$ such that $\frac{1}{L + 1} \Theta_{k-u} = \sum_{\alpha \in \mathbb{N}_k^{m-1}} \theta_{k-u,\alpha} x^{2\alpha}$;
- $(\eta_{k-1,\alpha})_{\alpha \in \mathbb{N}_k^{m-1}} \subseteq \mathbb{R}^{>0}$ such that $\Lambda_{k-1} = \sum_{\alpha \in \mathbb{N}_k^{m-1}} \eta_{k-1,\alpha} x^{2\alpha}$.

ACM Transactions on Mathematical Software, Vol. 48, No. 4, Article 40. Publication date: December 2022.
Exploiting Constant Trace Property in Large-scale Polynomial Optimization

For each $k \in \mathbb{N}^{\geq k_{\text{min}}}$, define the diagonal matrices $G^{(0)}_k := \text{diag}((v_{k,a})_{a \in \mathbb{N}_n^k})$, $G^{(1)}_{k-u} := \text{diag}((\theta_{k-u,a})_{a \in \mathbb{N}_n^{k-u}})$, and $G^{(2)}_{k-1} := \text{diag}((\eta_{k-1,a})_{a \in \mathbb{N}_n^{k-1}})$.

Then (3.28) yields that for every $k \in \mathbb{N}^{\geq k_{\text{min}}}$,

$$\textstyle (R + 1)^k = v_k^T \sum_{i \in [m-1]} g_i + v_{k-u}^T G^{(1)}_{k-u} v_{k-u} \sum_{i \in [m-1]} g_i + v_{k-1}^T G^{(2)}_{k-1} v_{k-1} g_m. \quad (3.29)$$

Hence, $(R + 1)^k, G^{(i)}_k, 0$ is a feasible solution of (3.21), for every $k \in \mathbb{N}^{\geq k_{\text{min}}}$. By using Lemma 3.8, the conclusion follows:

**Example 3.15.** Let $R, L > 0$ satisfy $R \geq L^2$ and

$$\textstyle m = n + 2, g_i = x_i \text{ for } i \in [n], g_{n+1} = L - \sum_{i \in [n]} x_i \text{ and } g_{n+2} = R - \|x\|_2^2. \quad (3.30)$$

Then Assumption 3.13 holds and $S(g)$ is a simplex.

When $S(g)$ is compact, we can always reformulate POP (2.5) such that Assumption 3.13 holds. Suppose $S(g) \subseteq \mathcal{B}_R$ for some $R$. Let $u = \max_{i \in [m]} |g_i|$. Set $\tilde{g}_i := g_i (1 + \|x\|_2^2)^u - [g_i]$ for $i \in [m]$. Let $L$ be a positive number such that $\sum_{i \in [m]} \tilde{g}_i \leq L$ on $S(g)$. Set $\tilde{g}_{m+1} := L - \sum_{i \in [m]} \tilde{g}_i$ and $\tilde{g}_{m+2} := R - \|x\|_2^2$.

**Remark 3.16.** For the latter case, one can choose any positive number $L \geq (R + 1)^u \sum_{i \in [m]} \|g_i\|_1$. Indeed, for any $z \in S(g)$, and, since $\|z\|_2^2 \leq R$,

$$\textstyle |z|^{\alpha} = \prod_{i \in [n]} |z_i|^{\alpha_i} \leq \prod_{i \in [n]} (1 + \|z\|_2^2)^{\alpha_i/2} = (1 + \|z\|_2^2)^{|\alpha|/2} \leq (1 + R)^{|\alpha|/2}, \forall \alpha \in \mathbb{N}_{2n}^n.$$

This implies that for every $i \in [m]$,

$$\textstyle \tilde{g}_i(z) \leq (1 + R)^u - [g_i] \sum_{\alpha \in \mathbb{N}_{2n}^n} |g_{\alpha}| |z|^{\alpha} \leq (1 + R)^u - [g_i] (R + 1)^u |g_i| \|g_i\|_1 = (1 + R)^u \|g_i\|_1.$$

Thus, we have $\sum_{i \in [m]} \tilde{g}_i \leq (1 + R)^u \sum_{i \in [m]} \|g_i\|_1$ on $S(g)$.

**Corollary 3.17.** With the above notation, $S(g \cup \{\tilde{g}_{m+1}, \tilde{g}_{m+2}\}) = S(g)$ and LP (3.21) has a feasible solution when replacing $g$ by $g \cup \{\tilde{g}_{m+1}, \tilde{g}_{m+2}\}$ for each $k \in \mathbb{N}^{\geq k_{\text{min}}}$. As a result, POP (2.5) is equivalent to the following POP:

$$\textstyle f^* := \inf \{ f(x) : x \in S(g \cup \{\tilde{g}_{m+1}, \tilde{g}_{m+2}\}) \cap V(h) \}, \quad (3.31)$$

which has CTP.

**Proof.** Let $\tilde{g} := \{\tilde{g}_i\}_{i \in [m+2]}$. Then $\{\tilde{g}_i\}_{i \in [m]}$ are of equivalent degree, i.e., there exists $u \in \mathbb{N}$ such that $\|\tilde{g}_i\| = u$, for all $i \in [m]$. Thus, Assumption 3.13 holds with $g \leftarrow \tilde{g}, m \leftarrow m + 2$. By Proposition 3.14, (3.21) has a feasible solution with $g \leftarrow \tilde{g}$ for every order $k \in \mathbb{N}^{\geq k_{\text{min}}}$. It implies that for every $k \in \mathbb{N}^{\geq k_{\text{min}}}$, there exist $u^{(l)}_k \in \mathbb{R}^{(2(k-\lceil h_j \rceil))}$, $j \in [l]$, and

$$\textstyle \eta^{(0)}_{k,a} \in \mathbb{R}^{\geq 0}, \quad \eta^{(l)}_{k-u,a} \in \mathbb{R}^{\geq 0}, \quad i \in [m + 1], \quad \eta^{(m+2)}_{k-1,a} \in \mathbb{R}^{\geq 0}, \quad \text{such that}$$

$$\textstyle 1 = v_k^T \text{diag} \left( (\eta^{(0)}_{k,a})_{a \in \mathbb{N}_n^k} \right) v_k + \sum_{i \in [m+1]} \tilde{g}_i v_{k-u}^T \text{diag} \left( (\eta^{(l)}_{k-u,a})_{a \in \mathbb{N}_n^{k-u}} \right) v_{k-u}$$

$$\textstyle + \tilde{g}_{m+2} v_{k-1}^T \text{diag} \left( (\eta^{(m+2)}_{k-1,a})_{a \in \mathbb{N}_n^{k-1}} \right) v_{k-1} + \sum_{j \in [l]} h_j v_{2(k-\lceil h_j \rceil)} u^{(l)}_k.$$
Let $k \in \mathbb{N}^{\geq k_{\min}}$ be fixed. We define the following polynomials:

- $\sigma_0 := \mathbf{v}_k^T \text{diag}((\eta_{k,\alpha}^{(0)})_{\alpha \in \mathbb{N}^n}) \mathbf{v}_k = \sum_{\alpha \in \mathbb{N}^n} \eta_{k,\alpha}^{(0)} x^{2\alpha}$,
- $\sigma_i := \mathbf{v}_{k-u}^T \text{diag}((\eta_{k-u,\alpha}^{(i)})_{\alpha \in \mathbb{N}^n}) \mathbf{v}_{k-u} = \sum_{\alpha \in \mathbb{N}^n} \eta_{k-u,\alpha}^{(i)} x^{2\alpha}$, $i \in [m + 1]$,
- $\sigma_{m+2} := \mathbf{v}_{k-1}^T \text{diag}((\eta_{k-1,\alpha}^{(m+2)})_{\alpha \in \mathbb{N}^n}) \mathbf{v}_{k-1} = \sum_{\alpha \in \mathbb{N}^n} \eta_{k-1,\alpha}^{(m+2)} x^{2\alpha}$,
- $\psi_j := \mathbf{v}_{2(k-\lfloor h_j \rfloor)}^T u_{k}^{(j)}$, $j \in [l]$.

From this and since $\tilde{g}_i := g_i (1 + \|x\|^2) - g_i$, for $i \in [m]$, one has

$$1 = \sigma_0 + \sum_{i \in [m]} \sigma_i \tilde{g}_i + \sum_{j \in [l]} \psi_j h_j = \sigma_0 + \sum_{i \in [m]} \sigma_i (1 + \|x\|^2) - g_i + \tilde{g}_{m+1} \sigma_{m+1} + \tilde{g}_{m+2} \sigma_{m+2} + \sum_{j \in [l]} \psi_j h_j,$$

(3.32)

Then there exist $(\theta_{k-\lfloor g_i \rfloor,\alpha}^{(i)})_{\alpha \in \mathbb{N}^n_{k-\lfloor g_i \rfloor}} \subseteq \mathbb{R}^{\geq 0}$, $i \in [m]$, such that

$$\sigma_i (1 + \|x\|^2) - g_i = \sum_{\alpha \in \mathbb{N}^n_{k-\lfloor g_i \rfloor}} \theta_{k-\lfloor g_i \rfloor,\alpha}^{(i)} x^{2\alpha}, \quad i \in [m].$$

(3.33)

Thus, (3.32) becomes

$$1 = \mathbf{v}_k^T \text{diag}((\eta_{k,\alpha}^{(0)})_{\alpha \in \mathbb{N}^n_k}) \mathbf{v}_k + \sum_{i \in [m]} \tilde{g}_i \mathbf{v}_{k-\lfloor g_i \rfloor}^T \text{diag}((\theta_{k-\lfloor g_i \rfloor,\alpha}^{(i)})_{\alpha \in \mathbb{N}^n_{k-\lfloor g_i \rfloor}}) \mathbf{v}_{k-\lfloor g_i \rfloor} + \tilde{g}_{m+1} \mathbf{v}_{k-u}^T \text{diag}((\eta_{k-u,\alpha}^{(m+1)})_{\alpha \in \mathbb{N}^n_{k-u}}) \mathbf{v}_{k-u} + \tilde{g}_{m+2} \mathbf{v}_{k-1}^T \text{diag}((\eta_{k-1,\alpha}^{(m+2)})_{\alpha \in \mathbb{N}^n_{k-1}}) \mathbf{v}_{k-1} + \sum_{j \in [l]} h_j \mathbf{v}_{2(k-\lfloor h_j \rfloor)}^T u_{k}^{(j)} + I_k(h),$$

since

- $\text{diag}((\eta_{k,\alpha}^{(0)})_{\alpha \in \mathbb{N}^n_k}) > 0$, $\text{diag}((\theta_{k-\lfloor g_i \rfloor,\alpha}^{(i)})_{\alpha \in \mathbb{N}^n_{k-\lfloor g_i \rfloor}}) > 0$, $i \in [m]$,
- $\text{diag}((\eta_{k-u,\alpha}^{(m+1)})_{\alpha \in \mathbb{N}^n_{k-u}}) > 0$, and $\text{diag}((\eta_{k-1,\alpha}^{(m+2)})_{\alpha \in \mathbb{N}^n_{k-1}}) > 0$.

It yields that (3.21) has a feasible solution with $g \leftarrow g \cup \{\tilde{g}_{m+1}, \tilde{g}_{m+2}\}$, for every order $k \in \mathbb{N}^{\geq k_{\min}}$. \hfill $\square$

In case that POP (2.5) does not have CTP and $S(g)$ is compact, Corollary 3.17 provides a way to construct an equivalent POP by including two additional redundant constraints. Then CTP of this new POP can be verified by LP.

### 3.5 Main Algorithm

Algorithm 1 below solves POP (2.5) whose CTP can be verified by LP.

**ALGORITHM 1:** Approximating the optimal value of a dense POP with CTP

**Input:** POP (2.5) and a relaxation order $k \in \mathbb{N}^{\geq k_{\min}}$

**Output:** The optimal value $\tau_k$ of SDP (3.13)

1. Solve LP (3.21) with an optimal solution $(\xi_k, \mathbf{G}_{i,k}, \mathbf{u}_{j,k})$;
2. Let $a_k = \xi_k$ and $P_k = \text{diag}(G_{0,k}^{1/2}, \ldots, G_{m,k}^{1/2})$;
3. Compute the optimal value $\tau_k$ of SDP (3.13) by running an algorithm based on first-order method and that exploits CTP.

Examples of algorithms based on first-order methods and that exploit CTP are CGAL (Algorithm 3 in Appendix A.4.1) or SM (Algorithm 5 in Appendix A.5.1).

ACM Transactions on Mathematical Software, Vol. 48, No. 4, Article 40. Publication date: December 2022.
Table 3. Notation

| Symbol | Description |
|--------|-------------|
| $n$    | the number of variables of a POP |
| $m$    | the number of inequality constraints of a POP |
| $l$    | the number of equality constraints of a POP |
| $u^{\text{max}}$ | the largest size of variable cliques of a sparse POP |
| $p$    | the number of variable cliques of a sparse POP |
| $k$    | the relaxation order of the Moment-SOS hierarchy |
| $t$    | the sparse order of the sparsity adapted Moment-SOS hierarchy (for TS and CS-TS) |
| $\omega$ | the number of psd blocks of an SDP |
| $s^{\text{max}}$ | the largest size of psd blocks of an SDP |
| $\zeta$ | the number of affine equality constraints of an SDP |
| $a^{\text{max}}$ | the largest constant trace |
| **Mosek** | the SDP relaxation modeled by SumOfSquares (for dense POPs) or TSSOS (for sparse POPs) and solved by Mosek 9.1 |
| **CGAL** | the SDP relaxation modeled by our CTP-exploiting method and solved by the CGAL algorithm |
| **LMBM** | the SDP relaxation modeled by our CTP-exploiting method and solved by the SM algorithm with the LMBM solver |
| **val** | the optimal value of the SDP relaxation |
| **gap** | the relative optimality gap w.r.t. the value returned by Mosek, i.e., $\text{gap} = \frac{|\text{val} - \text{val}(\text{Mosek})|}{|\text{val}(\text{Mosek})|}$ |
| **time** | the running time in seconds (including modeling and solving time) |
| $-$    | the calculation runs out of space |

4 Numerical Experiments for Dense Pops

In this section, we report results of numerical experiments obtained by solving the second-order Moment-SOS relaxation of various randomly generated instances of QCQPs with CTP. The experiments were performed in Julia 1.3.1 with the following software packages:

- **SumOfSquares** [47] is a modeling library for solving the Moment-SOS relaxations of dense POPs, based on JuMP (with Mosek 9.1 as SDP solver).
- **TSSOS** [44–46] is a modeling library for solving Moment-SOS relaxations of sparse POPs based on JuMP (with Mosek 9.1 as SDP solver).
- **LMBM** solves unconstrained non-smooth optimization with the limited-memory bundle method by Haarala et al. [12, 13] and calls Kormitsa’s Fortran implementation of the LMBM algorithm [21].
- **Arpack** [29] is used to compute the smallest eigenvalues and the corresponding eigenvectors of real symmetric matrices of (potentially) large size, which is based on the implicitly restarted Arnoldi method.

The implementation of Algorithms 1 and 2 is available online via the link: https://github.com/maihoanganh/ctpPOP.

We use a desktop computer with an Intel(R) Core(TM) i7-8665U CPU @ 1.9 GHz × 8 and 31.2 GB of RAM. The notation for the numerical results is given in Table 3.

For the examples tested in this article, the modeling time of SumOfSquares, TSSOS, and ctpPOP is typically negligible compared to the solving time of the packages Mosek, CGAL, and LMBM.
Table 4. Numerical Results for Minimizing a Dense Quadratic Polynomial on a Unit Ball

| POP size | SDP size | Mosek | CGAL | LMBM |
|----------|----------|-------|------|------|
| n        | s_{max}  | \zeta | val  | time | val  | time | val  | time |
| 10       | 66       | 1,277 | -2.2181 | 0.3 | -2.2170 | 0.2 | -2.2187 | 0.3 |
| 20       | 231      | 16,402 | -3.7973 | 4 | -3.7947 | 0.6 | -3.7906 | 7 |
| 30       | 496      | 77,377 | -3.6876 | 3474 | -3.6858 | 104 | -3.8530 | 59 |
| 40       | 861      | 236,202 | - | - | -4.1718 | 33 | -4.7730 | 179 |
| 50       | 1,326    | 564,877 | - | - | -6.3107 | 1,007 | -7.3874 | 139 |
| 60       | 1,891    | 1,155,402 | - | - | -6.5326 | 1,085 | -7.4733 | 674 |
| 70       | 2,556    | 2,119,777 | - | - | -7.3739 | 1,262 | -9.5223 | 1,486 |
| 80       | 3,321    | 3,590,002 | - | - | -7.9559 | 4,988 | -10.0260 | 1,241 |
| 90       | 4,186    | 5,718,077 | - | - | -7.3425 | 5,187 | -9.4477 | 5,313 |
| 100      | 5,151    | 8,676,002 | - | - | -7.7374 | 2,245 | -10.684 | 5,355 |

POP size: m = 1, l = 0; Relaxation order: k = 2; SDP size: \( \omega = 2, a_{\text{max}} = 3 \).

Table 5. Numerical Results for Randomly Generated Dense QCQPs with a Ball Constraint

| POP size | SDP size | Mosek | CGAL | LMBM |
|----------|----------|-------|------|------|
| n        | l        | s_{max}  | \zeta | val  | time | val  | time | val  | time |
| 10       | 3        | 66      | 1,475 | -2.0686 | 1.7 | -2.0674 | 0.8 | -2.0874 | 0.3 |
| 20       | 5        | 231     | 17,557 | -3.0103 | 61 | -3.0075 | 7 | -3.0750 | 18 |
| 30       | 8        | 496     | 81,345 | -3.3293 | 4,573 | -3.3249 | 80 | -3.683 | 123 |
| 40       | 10       | 861     | 244,812 | - | - | -4.6977 | 194 | -5.3488 | 488 |
| 50       | 13       | 1,326   | 582,115 | - | - | -4.2394 | 951 | -6.1325 | 837 |
| 60       | 15       | 1,891   | 1,183,767 | - | - | -5.7793 | 1,387 | -7.5718 | 3,781 |
| 70       | 18       | 2,556   | 2,165,785 | - | - | -6.1278 | 4,335 | -8.1181 | 15,854 |

POP size: m = 1, l = \lceil n/4 \rceil; Relaxation order: k = 2; SDP size: \( \omega = 2, a_{\text{max}} = 3 \).

Hence, the total running time mainly depends on the solvers and we compare their performances below. As mentioned in the introduction, the current framework differs from our previous work [30], where we exploited CTP for equality-constrained POPs on a sphere, which could be solved by LMBM efficiently. The reason is that the SDP relaxations of such equality-constrained POPs involve a single psd matrix. For the benchmarks of this section, we consider POPs involving ball/annulus constraints, and so the resulting relaxations include several psd matrices. Our numerical experiments confirm that for such SDPs, LMBM returns inaccurate values, since the gap w.r.t. the value of Mosek is typically larger than 1%, while CGAL (without sketching) performs better for this type of SDPs in terms of accuracy. As shown in Section 4.1, the last columns of Tables 4 and 5 illustrate how inaccurate LMBM can be for large problems (n \geq 20), thus, we do not report LMBM results in the other experiments.

4.1 Randomly Generated Dense QCQPs with a Ball Constraint

**Test problems:** We construct randomly generated dense QCQPs with a ball constraint as follows:

1. Generate a dense quadratic polynomial objective function \( f \) with random coefficients following the uniform probability distribution on \((-1, 1)\);
2. Let \( m = 1 \) and \( g_i := 1 - \|x\|_2^2 \);
3. Take a random point \( a \) in \( S(g) \) w.r.t. the uniform distribution;
Table 6. Numerical Results for Minimizing a Dense Quadratic Polynomial on an Annulus

| POP size | SDP size | Mosek | CGAL |
|----------|----------|-------|------|
|          | $s_{\text{max}}$ | $\zeta$ | val | time | val | time |
| n |  | | | | | |
| 10 | 66 | 1,343 | −3.0295 | 0.5 | −3.0278 | 1 |
| 20 | 231 | 16,633 | −3.6468 | 69 | −3.6458 | 5 |
| 30 | 496 | 77,873 | −3.9108 | 2,546 | −3.9079 | 9 |
| 40 | 861 | 237,063 | − | − | −4.7469 | 28 |
| 50 | 1,326 | 566,203 | − | − | −6.4170 | 112 |
| 60 | 1,891 | 1,157,293 | − | − | −7.9325 | 130 |
| 70 | 2,556 | 2,122,333 | − | − | −7.6164 | 226 |
| 80 | 3,321 | 3,593,323 | − | − | −7.6164 | 226 |
| 90 | 4,186 | 5,722,263 | − | − | −8.1900 | 3,563 |

POP size: $m = 2$, $l = 0$; Relaxation order: $k = 2$; SDP size: $\omega = 3$, $a_{\text{max}} = 4$.

Table 7. Numerical Results for Randomly Generated Dense QCQPs with Annulus Constraints

| POP size | SDP size | Mosek | CGAL |
|----------|----------|-------|------|
|          | $s_{\text{max}}$ | $\zeta$ | val | time | val | time |
| n |  | | | | | |
| 10 | 3 | 66 | 1,541 | −2.7950 | 0.5 | −2.7934 | 2 |
| 20 | 5 | 231 | 17,788 | −3.5048 | 95 | −3.5027 | 10 |
| 30 | 8 | 496 | 81,841 | −3.3964 | 4,237 | −3.3937 | 45 |
| 40 | 10 | 861 | 245,673 | − | − | −6.573 | 140 |
| 50 | 13 | 1,326 | 583,441 | − | − | −3.8236 | 437 |
| 60 | 15 | 1,891 | 1,185,658 | − | − | −4.5246 | 1,076 |
| 70 | 18 | 2,556 | 2,168,341 | − | − | −6.2924 | 4,783 |

POP size: $m = 2$, $l = \lceil n/4 \rceil$; Relaxation order: $k = 2$; SDP size: $\omega = 3$, $a_{\text{max}} = 4$.

(4) For every $j \in [l]$, generate a dense quadratic polynomial $h_j$ by
   (i) for each $\alpha \in \mathbb{N}_n^2 \setminus \{0\}$, taking a random coefficient $h_{j, \alpha}$ for $h_j$ in $(-1, 1)$ w.r.t. the uniform distribution;
   (ii) setting $h_{j, 0} := -\sum_{\alpha \in \mathbb{N}_n^2 \setminus \{0\}} h_{j, \alpha} a^\alpha$.
Then $a$ is a feasible solution of POP (2.5).

The numerical results are displayed in Tables 4 and 5.

Discussion: As one can see from Tables 4 and 5, CGAL is typically the fastest solver and returns an optimal value of gap within 1% w.r.t. the one returned by Mosek when $n \leq 30$. Mosek runs out of memory when $n \geq 40$, while CGAL works well up to $n = 100$. We should point out that LMBM is less accurate or even fails to converge to the optimal value when $n \geq 20$. The reason might be that LMBM only solves the dual problem and hence loses information of the primal problem.

4.2 Randomly Generated Dense QCQPs with Annulus Constraints

Test problems: We construct randomly generated dense QCQPs as in Section 4.1, where the ball constraint is now replaced by annulus constraints. Namely, in Step 2, we take $m = 2$, $g_1 := \|x\|_2^2 - 1/2$ and $g_2 := 1 - \|x\|_2^2$. The numerical results are displayed in Tables 6 and 7.
4.3 Randomly Generated Dense QCQPs with Box Constraints

Test problems: We construct randomly generated dense QCQPs as in Section 4.1, where the ball constraint is now replaced by box constraints. Namely, in Step 2, we take \( m = n, g_j := -x_j^2 + 1/n, j \in [n] \).

The numerical results are displayed in Tables 8 and 9.

Discussion: We observe similar behaviors of the solvers as in Section 4.1. One can also see that solving a QCQP with box constraints is less efficient than solving the same one with ball constraints. This is because the efficiency of CGAL depends on the number of psd blocks involved in an SDP. For instance, when \( n = 50 \), CGAL takes around 1,000 seconds to solve the second-order moment relaxation of a QCQP with a ball constraint, while it takes around 2,100 seconds to solve this relaxation for a QCQP with box constraints.

4.4 Randomly Generated Dense QCQPs with Simplex Constraints

Test problems: We construct randomly generated dense QCQPs as in Section 4.1, where the ball constraint is now replaced by simplex constraints. Namely, in Step 2, we take \( g \) such that (3.30) holds with \( L = R = 1 \). The numerical results are displayed in Tables 10 and 11.

Discussion: Again, we observe a behavior of the solvers similar to that in Section 4.1. One can also see that solving a QCQP with simplex constraints by CGAL is significantly slower than solving the same one with box constraints. For instance, when \( n = 50 \), CGAL takes 2,100 seconds to solve
Table 10. Numerical Results for Minimizing a Dense Quadratic Polynomials on a Simplex

| POP size | SDP size | Mosek | CGAL |
|----------|---------|-------|------|
|          |         | val   | time | val   | time |
| 10       | 66      | 2,003 | −1.9954 | 0.3 | −1.9950 | 7 |
| 20       | 231     | 21,253 | −1.5078 | 58 | −1.5055 | 116 |
| 30       | 496     | 92,753 | −2.0537 | 2,804 | −2.0480 | 377 |
| 40       | 861     | 271,503 | − | − | −2.3034 | 950 |
| 50       | 1,326   | 632,503 | − | − | −1.8366 | 9,539 |

POP size: \( m = n + 2, l = 0 \); Relaxation order: \( k = 2 \); SDP size: \( \omega = n + 3, a_{\text{max}} = 5 \).

Table 11. Numerical Results for Randomly Generated Dense QCQPs with Simplex Constraints

| POP size | SDP size | Mosek | CGAL |
|----------|---------|-------|------|
|          |         | val   | time | val   | time |
| 10       | 2       | 66    | 2,135 | −1.0605 | 0.4 | −1.0606 | 176 |
| 20       | 3       | 231   | 21,946 | −1.6629 | 72 | −1.6628 | 512 |
| 30       | 5       | 496   | 95,233 | −1.0091 | 6,206 | −1.0249 | 1,089 |
| 40       | 6       | 861   | 276,669 | − | − | −0.3256 | 2,314 |
| 50       | 8       | 1,326 | 643,111 | − | − | −1.4200 | 10,035 |

POP size: \( m = n + 2, l = \lceil n/7 \rceil \); Relaxation order: \( k = 2 \); SDP size: \( \omega = n + 3, a_{\text{max}} = 5 \).

Table 12. Numerical Comparison with ADMM (COSMO) on Randomly Generated Dense QCQPs with a Ball Constraint

| POP size | SDP size | Mosek | CGAL | COSMO |
|----------|---------|-------|------|-------|
|          |         | val   | time | val   | time | val   | time |
| 10       | 3       | 66    | 1,475 | −2.3153 | 0.7 | −2.3134 | 0.1 | −2.3125 | 0.2 |
| 20       | 5       | 231   | 17,557 | −3.6585 | 57 | −3.6562 | 7 | −3.6582 | 5 |
| 30       | 8       | 496   | 81,345 | −4.6221 | 4,670 | −4.6177 | 69 | −4.6230 | 91 |
| 40       | 10      | 861   | 244,812 | − | − | −4.9932 | 173 | −4.9989 | 532 |
| 50       | 13      | 1,326 | 582,115 | − | − | −5.0394 | 524 | −5.0418 | 2,468 |
| 60       | 15      | 1,891 | 1,183,767 | − | − | −5.3537 | 735 | −5.3548 | 6,176 |

POP size: \( m = 1, l = \lceil n/4 \rceil \); Relaxation order: \( k = 2 \); SDP size: \( \omega = 2, a_{\text{max}} = 3 \).

the second-order moment relaxation for a QCQP with box constraints, while it takes 9,500 seconds with simplex constraints.

4.5 Numerical Comparison between CGAL and ADMM

In Table 12, we make a numerical comparison between CGAL (with our CTP-exploiting method) and COSMO, an SDP solver based on ADMM (see Table 1) on some randomly generated dense QCQPs with a ball constraint (as in Section 4.1).

Discussion: Table 12 indicates that both CGAL and COSMO provide approximate values with gap within 1% w.r.t. the ones returned by Mosek when \( n \leq 30 \). In addition, COSMO is slightly more accurate for \( n \in \{20, 30\} \) while CGAL offers an increasing speedup when \( n \geq 30 \).
Table 13. Numerical Results for Randomly Generated Dense POPs with a Ball Constraint

| POP size | SDP size | Mosek | CGAL |
|----------|----------|-------|------|
| n        | l        | d     | k    | a^{max} | s^{max} | ω      | val | time | val | time |
| 15       | 4        | 3     | 2    | 3       | 136     | 5,581   | −3.0127 | 5   | −3.0089 | 1   |
|          |          |       | 3    | 4       | 816     | 288,933 | −       | −   | −3.0021 | 290 |
| 10       | 3        | 4     | 2    | 3       | 66      | 1,280   | −2.0327 | 0.3 | −2.0194 | 0.6 |
|          |          |       | 3    | 4       | 286     | 35,443  | −1.9337 | 41  | −1.9310 | 16  |

POP size: m = 1, l = ⌈n/4⌉; SDP size: ω = 2.

4.6 Dense POPs with a Ball Constraint

We construct randomly generated dense POPs as in Section 4.1, with input data of degree $d \in \{3, 4\}$. The numerical results, displayed in Table 13, indicate that CGAL returns an optimal value with gap within 1% w.r.t. the one of Mosek, and is faster when the largest block size increases.

5 CONCLUSION

In this article, we have proposed a general framework for exploiting the constant trace property in solving large-scale SDPs, typically SDP-relaxations arising from the Moment-SOS hierarchy for POPs. Extensive numerical experiments strongly suggest that with this CTP formulation, the CGAL solver based on first-order methods is more efficient and more scalable than Mosek without exploiting CTP, especially when the block size is large. In addition, the optimal value returned by CGAL is typically within 1% w.r.t. the one returned by Mosek.

We have also integrated sparsity-exploiting techniques into the CTP-exploiting framework to handle large-scale POPs. For SDP-relaxations of large-scale POPs with a term and/or correlative sparsity pattern, and in applications for which only a medium accuracy of optimal solutions is enough, we believe that our framework should be very useful.

As a topic of further investigation, we would like to improve the LP-based formulation for verifying CTP, for instance by relying on more general second-order cone programming. We also would like to generalize the CTP-exploiting framework to noncommutative POPs [4, 23, 43], which have attracted a lot of attention in the quantum information community. Another line of research would be to investigate whether CTP could be efficiently exploited by interior-point solvers.

A APPENDIX

A.1 Exploiting CTP for POPs with CS

In this section, we extend the CTP-exploiting framework to POPs with sparsity. For clarity of exposition, we only consider correlative sparsity (CS). However, in Appendix A.2, we also treat term sparsity (TS) [45] as well as correlative-term sparsity (CS-TS) [46]. Since the methodology is very similar to that in the dense case described earlier, we omit details and only present the main results.

To begin with, we recall some basic facts on exploiting CS for POP (2.5) initially proposed in Reference [38] by Waki et al.

For $α \in \mathbb{N}_n$, let $\text{supp}(α) := \{j \in [n] : α_j > 0\}$. For $I \subseteq [n]$, let $x(I) := \{x_j : j \in I\}$ and $\mathbb{N}_d^n = \{α \in \mathbb{N}_n : \text{supp}(α) \subseteq I\}$. Assume $I \subseteq [n]$. Given $y = (y_α)_α\in\mathbb{N}_d^n$, the moment (respectively, localizing) submatrix associated with $I$ of order $d$ is defined by $M_d(y, I) := (y_{α+β})_α,β\in\mathbb{N}_d^n$ (respectively, $M_d(qy, I) := (\sum_γ q_γ y_{α+β+γ})_α,β\in\mathbb{N}_d^n$ for $q ∈ \mathbb{R}[x(I)]$). Let $y_d^{I_l} := (x^α)_α\in\mathbb{N}_d^n$ with length $s(I_l, d) := \binom{|I_l|}{n}^d$. For matrices $A$ and $B$ of same sizes, the Hadamard product of $A$ and $B$, denoted by $A \circ B$, is the matrix with entries $(A \circ B)_{i,j} = A_{i,j}B_{i,j}$.
A.1.1 POPs with CS. Assume that \(|L_j| \in [p]\) (with \(n_j := |L_j|\)) are the maximal cliques of (a chordal extension of) the correlative sparsity pattern (csp) graph associated with POP (2.5), as defined in Reference [38].

Let \(|L_j| \in [p]\) (respectively, \(|W_j| \in [p]\)) be a partition of \([m]\) (respectively, \([l]\)) such that for all \(i \in J_j, g_i \in \mathbb{R}[x(L_j)]\) (respectively, \(i \in W_j, h_i \in \mathbb{R}[x(L_j)]\)), \(j \in [p]\). For each \(j \in [p]\), let \(m_j := |J_j|, l_j := |W_j|\) and \(g_j := \{g_i : i \in J_j\}, h_W := \{h_i : i \in W_j\}\). Then \(Q(g_j)\) (respectively, \(I(h_W)\)) is a quadratic module (respectively, an ideal) in \(\mathbb{R}[x(L_j)]\), for \(j \in [p]\).

For each \(k \in \mathbb{N}^{\geq k_{\text{min}}}\), consider the following sparse SOS relaxation:

\[
\rho_k^{\text{cs}} := \sup \left\{ \xi : f - \xi \in \sum_{j \in [p]} \left( Q_k(g_j) + I_k(h_W) \right) \right\}. \tag{1.34}
\]

It is equivalent to the SDP:

\[
\rho_k^{\text{cs}} = \sup_{\xi, G_l^{(j)}, u_l^{(j)}} \left\{ \xi : \begin{aligned}
G_l^{(j)} &\geq 0, i \in [0] \cup J_j, j \in [p], \\
\sum_{j \in [p]} &\left( (v_k^{(j)} G_0^{(j)} v_k^{(j)})^T + \sum_{i \in J_j} g_i (v_k^{(j)} G_0^{(j)} v_k^{(j)})^T + \sum_{i \in W_j} h_i (v_k^{(j)} G_0^{(j)} v_k^{(j)})^T \right) W_j^{(j)} \right\}. \tag{1.35}
\]

The dual of (1.35) reads

\[
\tau_k^{\text{cs}} := \inf_{y \in \mathbb{R}^{n(2k)}} \left\{ L_y(f) : \begin{aligned}
M_k(y, l_j) &\geq 0, j \in [p], y_0 = 1, \\
M_{k-l_j}(g_i y, l_j) &\geq 0, i \in J_j, j \in [p], \\
M_{k-h_j}(h_i y, l_j) &\geq 0, i \in W_j, j \in [p].
\end{aligned} \right\}. \tag{1.36}
\]

It is shown in Reference [25, Theorem 3.6] that convergence of the primal-dual (1.35)–(1.36) to \(f^*\) is guaranteed if there are additional ball constraints on each clique of variables.

A.1.2 Exploiting CTP for POPs with CS. Consider POP (2.5) with CS described in Section A.1.1. For every \(j \in [p]\) and for every \(k \in \mathbb{N}^{\geq k_{\text{min}}}\), letting \(D_k(y, l_j) := \text{diag}(M_k(y, l_j), (M_{k-h_j}(g_i y, l_j))_{i \in J_j})\) for \(y \in \mathbb{R}^{n(2k)}\), SDP (1.36) can be rewritten as

\[
\tau_k^{\text{cs}} := \inf_{y \in \mathbb{R}^{n(2k)}} \left\{ L_y(f) : \begin{aligned}
D_k(y, l_j) &\geq 0, j \in [p], y_0 = 1, \\
M_{k-h_j}(h_i y, l_j) &\geq 0, i \in W_j, j \in [p].
\end{aligned} \right\}. \tag{1.37}
\]

We define CTP for POP with CS as follows:

**Definition A.1 (CTP for a POP with CS).** We say that POP (2.5) with CS has CTP if for every \(k \in \mathbb{N}^{\geq k_{\text{min}}}\) and for every \(j \in [p]\), there exists a positive number \(a_k^{(j)}\) and a positive definite matrix \(P_k^{(j)} \in S_k\) such that for all \(y \in \mathbb{R}^{n(2k)},\)

\[
M_{k-h_j}(h_i y, l_j) = 0, i \in W_j, y_0 = 1 \implies \text{trace}\left( P_k^{(j)} D_k(y, l_j) P_k^{(j)} \right) = a_k^{(j)}. \tag{1.38}
\]

The following result provides a sufficient condition for a POP with CS to have CTP:

**Theorem A.2.** Assume that there is a ball constraint on each clique of variables, i.e.,

\[
\forall j \in [p], R_j - \|x(L_j)\|_2^2 \in g \text{ for some } R_j > 0. \tag{1.39}
\]

Then one has \(\mathbb{R}^{>0} \subseteq Q_k^{o}(g_j)\), for all \(k \in \mathbb{N}^{\geq k_{\text{min}}}\) and all \(j \in [p]\). As a consequence, POP (2.5) has CTP.
The proof of Theorem A.2 being very similar to that of Theorem 3.5 by considering each clique of variables, is omitted.

Again, by considering each clique of variables, the following result can be obtained from Theorem A.2 in the same way Corollary 3.6 was obtained.

**Corollary A.3.** If (1.39) holds, then Slater’s condition for SDP (1.35) holds for all $k \in \mathbb{N}^{\geq k_{\text{min}}}$.

We are now in position to provide a general method to solve POPs with CS that have SDP.

Consider POP (2.5) with CS described in Section A.1.1. Assume that POP (2.5) has CTP and let $k \in \mathbb{N}^{\geq k_{\text{min}}}$ be fixed. For every $j \in [p]$, we denote by $S_{j,k}$ the set of real symmetric matrices of size $s(k,n_j) + \sum_{i \in J_j} s(k-[g_i], n_j)$ in a block diagonal form: $X = \text{diag}(X_0, (X_i)_{i \in J_j})$ such that $X_0$ is a block of size $s(k,n_j)$ and $X_i$ is a block of size $s(k-[g_i], n_j)$ for $i \in J_j$.

Letting

$$X_j = P_k^{(j)} D_k(y, I_j) P_k^{(j)}, \quad j \in [p],$$

SDP (1.37) can be rewritten as

$$\tau^*_k = \inf_{X_j \in S_{j,k}} \left\{ \sum_{j \in [p]} \langle C_{j,k}, X_j \rangle : \sum_{j \in [p]} \mathcal{A}_{j,k} X_j = b_k, \ j \in [p] \right\},$$

where for every $j \in [p]$, $\mathcal{A}_{j,k} : S_{j,k} \rightarrow \mathbb{R}^+$ is a linear operator of the form $\mathcal{A}_{j,k} X = (\langle A_{j,k,1}, X \rangle, \ldots, \langle A_{j,k,\xi_k}, X \rangle)$ with $A_{j,k,i} \in S_{j,k}, i \in [\xi_k], C_{j,k} \in S_{j,k}, j \in [p]$ and $b_k \in \mathbb{R}^\xi_k$. See Appendix A.6.2 for the conversion of SDP (1.37) to the form (1.41).

The dual of SDP (1.41) reads

$$\rho^*_k = \sup_{y \in \mathbb{R}^\xi_k} \left\{ b_k^T y : \mathcal{A}^T_{j,k} y - C_{j,k} \in S_{j,k}^+, \ j \in [p] \right\} ,$$

where $\mathcal{A}^T_{j,k} : \mathbb{R}^\xi_k \rightarrow S_{j,k}$ is the adjoint operator of $\mathcal{A}_{j,k}$, i.e., $\mathcal{A}^T_{j,k} z = \sum_{i \in [\xi_j]} z_i A_{j,k, i}, \ j \in [p]$. By Definition A.1, it holds that for every $k \in \mathbb{N}^{\geq k_{\text{min}}},$

$$\forall X_j \in S_{j,k}, \ j \in [p], \quad \sum_{j \in [p]} \mathcal{A}_{j,k} X_j = b_k \Rightarrow \text{trace}(X_j) = a_k^{(j)}, \ j \in [p].$$

After replacing $(\mathcal{A}_{j,k}, A_{j,k,i}, b_k, C_{j,k}, S_{j,k}, \xi_k, \tau^*_k, a_k^{(j)})$ by $(\mathcal{A}_{j,k}, A_{j,k,i}, b, C_j, S_j, \xi, \tau, a_j)$, SDP (1.41) then becomes SDP (1.61); see Appendix A.4.2 with $\omega_j = m_j + 1$ and $s_{\text{max}} = \max_{j \in [p]} s(k,n_j)$.

If there is a ball constraint on each clique of variables, then by Corollary A.3, strong duality holds for the pair (1.41)–(1.42), for every $k \in \mathbb{N}^{\geq k_{\text{min}}}$. The two algorithms (CGAL and SM) based on first-order methods are then leveraged to solve the primal-dual (1.41)–(1.42); see Appendix A.4.2 and Appendix A.5.2.

### A.1.3 Verifying CTP for POPs with CS Via LP.

As in the dense case, we can verify CTP for a POP with CS via a series of LPs.

For every $k \in \mathbb{N}^{\geq k_{\text{min}}}$ and for every $j \in [p]$, let $\hat{S}_{k,j}$ be the set of real diagonal matrices of size $s(k,n_j)$ and consider the following LP:

$$\inf_{\xi, G_i, u_i} \left\{ \xi : \begin{array}{l} G_0 - I_0 \in \hat{S}^+_{k,j}, \ G_i - I_i \in \hat{S}^+_{k-[g_i],j}, \ i \in J_j, \\
\xi = (v_k^j)^T G_0 v_k^j + \sum_{i \in J_j} g_i (v_k^j)^T G_i v_k^j + \sum_{i \in W_j} h_i (v_k^j)^T u_i \end{array} \right\} ,$$

where $I_i$ is the identity matrix, for every $i \in \{0\} \cup J_j$. 

ACM Transactions on Mathematical Software, Vol. 48, No. 4, Article 40. Publication date: December 2022.
Lemma A.4. Let POP (2.5) with CS be described in Section A.1.1. If LP (1.44) has a feasible solution \( \min \{ R^j + 1 : \| u^j \|_2 \} \), for every \( k \in \mathbb{N}^{\geq \text{min}} \) and for every \( j \in [p] \), then POP (2.5) has CTP with \( \bar{F}_k = \text{diag}(G_{0,k}^{1/2}, G_{1,k}^{1/2}, \ldots, G_{q,k}^{1/2}) \) and \( a_k^{(j)} = \bar{v}_k^{(j)} \), for \( k \in \mathbb{N}^{\geq \text{min}} \) and for \( j \in [p] \).

The proof of Lemma A.4 is similar to that of Lemma 3.8.

For instance, for POPs with ball or annulus constraints on subsets of each clique of variables, CTP can be verified by LP.

Proposition A.5. Let POP (2.5) with CS be described in Section A.1.1. Let \( (T_j)_{j \in \mathbb{N}^{\geq \text{min}}} \) be as in Assumption 3.10 and further assume that for every \( j \in [p], (\cup_{q \in J_j \cap [r]} T_q) \cup (\cup_{q \in J_j \cap [r]} T_q) = I_j \). Then LP (1.44) has a feasible solution for every \( k \in \mathbb{N}^{\geq \text{min}} \), and therefore POP (2.5) has CTP.

Proof. To prove that POP (2.5) has CTP on each clique of variables, it is sufficient to show that (1.44) has a feasible solution, for every \( k \in \mathbb{N}^{\geq \text{min}} \) and for every \( j \in [p] \) due to Lemma A.4.

For every \( j \in [p] \), let \( u^{(j)} \) be defined by

\[
u^{(j)}_i = |\{ q \in J_j \cap [r] : i \in T_q \}| + |\{ q \in J_j \cap [2r] : i \in T_q \}|, \quad i \in I_j.
\]

For every \( j \in [p] \), one has \( u^{(j)}_i \in \mathbb{N}^{\geq 1} \), \( i \in I_j \), according to \( (\cup_{q \in J_j \cap [r]} T_q) \cup (\cup_{q \in J_j \cap [2r]} T_q) = I_j \).

Moreover,

\[
\| u^{(j)} \circ x(I_j) \|_2^2 = \sum_{i \in J_j \cap [r]} \| x(T_i) \|_2^2 + \sum_{i \in J_j \cap [2r]} \| x(T_i) \|_2^2, \quad \forall j \in [p].
\]

For every \( j \in [p] \), with \( R^{(j)} := \sum_{i \in J_j \cap [r]} (R_i + \bar{R}_i) + \sum_{i \in J_j \cap [2r]} R_i \), by replacing \( x \) (respectively, \( R \)) by \( u^{(j)} \circ x(I_j) \) (respectively, \( R^{(j)} \)) in Lemma 3.4, we obtain

\[
(R^{(j)} + 1)^k = (1 + \| u^{(j)} \circ x(I_j) \|_2^2)^k + \Lambda^{(j)}_{k-1} \sum_{i \in I_j} \delta_i g_i, \quad \forall j \in [p], \forall k \in \mathbb{N}^{\geq \text{min}},
\]

where \( \Lambda_{k-1} := \sum_{r=0}^{k-1} (R^{(j)} + 1)^r (1 + \| u^{(j)} \circ x(I_j) \|_2^2)^{k-r-1} \) and

\[
\delta_i := \frac{R_i}{R_i - \bar{R}_i}, \quad \delta_{i+r} := \frac{\bar{R}_i}{\bar{R}_i - \bar{R}_i}, \quad i \in J_j \cap [r] \text{ and } \delta_q = 1, q \in J_j \cap [2r].
\]

It is due to the fact that

\[
R^{(j)} - \| u^{(j)} \circ x(I_j) \|_2^2 = \sum_{i \in J_j \cap [r]} (R_i + \bar{R}_i - \| x(T_i) \|_2^2) + \sum_{i \in J_j \cap [2r]} (\bar{R}_i - \| x(T_i) \|_2^2),
\]

and \( R_i + \bar{R}_i - \| x(T_i) \|_2^2 = \delta_i g_i + \delta_{i+r} g_{i+r}, i \in J_j \cap [r] \). For every \( j \in [p] \), for each \( k \in \mathbb{N}^{\geq \text{min}} \), let \( (\theta^{(j)}_{k,a})_{a \in I^f_j} \subseteq \mathbb{R}^{\geq 0} \) and \( (\eta^{(j)}_{k-1,a})_{a \in I^f_{k-1}} \subseteq \mathbb{R}^{\geq 0} \) be such that

\[
(1 + \| u^{(j)} \circ x(I_j) \|_2^2)^k = \sum_{a \in I^f_j} \theta^{(j)}_{k,a} x^{2a} \quad \text{and} \quad \Lambda_{k-1}^{(j)} = \sum_{a \in I^f_{k-1}, a \in I^f_j} \eta^{(j)}_{k-1,a} x^{2a},
\]

and define the diagonal matrices

\[
G_{k}^{(j),0} := \text{diag}(\theta^{(j)}_{k,a})_{a \in I^f_j} \quad \text{and} \quad G_{k-1}^{(j),i} := \text{diag}(\delta_i \eta^{(j)}_{k-1,a})_{a \in I^f_{k-1}}, i \in J_j.
\]

For every \( j \in [p] \), (1.47) yields that for every \( k \in \mathbb{N}^{\geq \text{min}} \),

\[
(R^{(j)} + 1)^k = (v_k^{(j)})^T G_{k}^{(j),0} v_k^{(j)} + \sum_{i \in J_j} g_i (v_k^{(j)})^T G_{k-1}^{(j),i} v_k^{(j)},
\]
Hence, \((R^{(j)} + 1)^k, G^{(j, l)}_k, 0\) is a feasible solution of \((1.44)\), for every \(k \in \mathbb{N}^{\geq k_{\text{min}}}\) and for every \(j \in [p]\). \(\Box\)

A.1.4 Main Algorithm. Algorithm 2 below solves POP (2.5) with CS and whose CTP can be verified by LP.

**Algorithm 2: Approximating the optimal value of a POP with CS and CTP**

**Input:** POP (2.5) with CS and a relaxation order \(k \in \mathbb{N}^{\geq k_{\text{min}}}\)

**Output:** The optimal value \(\tau^{\text{CS}}_k\) of SDP (1.41)

1: for \(j \in [p]\) do
2: Solve LP (1.44) to obtain an optimal solution \((\xi_k^{(j)}, G_k^{(j)}, u_k^{(j)})\);
3: Let \(d_k^{(j)} = \xi_k^{(j)}\) and \(F_k^{(j)} = \text{diag}((G_0^{(j)})^{1/2}, \ldots, (G_m^{(j)})^{1/2})\);
4: Compute the optimal value \(\tau^{\text{CS}}_k\) of SDP (1.41) by running an algorithm based on first-order methods and that exploits CTP.

In Step 4 of Algorithm 2 the two algorithms CGAL (Algorithm 4 in Appendix A.4.2 or SM (Algorithm 6 in Appendix A.5.2) are good candidates.

A.2 Exploiting CTP for POPs with TS and CS-TS

In this section, we restate the main results of TS in Reference [45] and CS-TS in Reference [46]. Similarly to dense POPs and POPs with CS, one can easily exploit CTP for POPs with TS and CS-TS. The central reason is that the diagonal of each moment/localizing matrix in a given moment relaxation of a dense POP (respectively, POP with CS) does not change when TS (respectively, CS-TS) is exploited.

A.2.1 Term Sparsity (TS). Fix a relaxation order \(k \in \mathbb{N}^{\geq k_{\text{min}}}\) and a sparse order \(t \in \mathbb{N} \setminus \{0\}\). We compute as in Reference [45, Section 5] the following block diagonal (up to permutation) \((0, 1)\)-binary matrices: \(G^{(0)}_{k, t}\) of size \(s(k)\); \(G^{(i)}_{k, t}\) of size \(s(k - [g_i])\), \(i \in [m]\); \(H^{(i)}_{k, t}\) of size \(s(k - [h_i])\), \(i \in [l]\). Then, we consider the following sparse moment relaxation of POP (2.5):

\[
\tau^{\text{TS}}_{k, t} := \inf_{y \in \mathbb{R}^{\langle 2^k \rangle}} \left\{ L_y(f) \left| \begin{array}{l}
G^{(0)}_{k, t} \circ M_k(y) \geq 0, y_0 = 1, \\
G^{(i)}_{k, t} \circ M_{k-[g_i]}(y) \geq 0, i \in [m], \\
H^{(i)}_{k, t} \circ M_{k-[h_i]}(y) = 0, i \in [l] 
\end{array} \right. \right\}. \tag{1.52}
\]

One has \(\tau^{\text{TS}}_{k, t-1} \leq \tau^{\text{TS}}_{k, t} \leq \tau_k \leq f^*, \) for all \((k, t)\). Moreover, we have the following theorem:

**Theorem A.6** (Wang et al. [45, Theorem 5.1]). For each \(k \in \mathbb{N}^{\geq k_{\text{min}}}\), the sequence \((\tau^{\text{TS}}_{k, t})_{t \in \mathbb{N} \setminus \{0\}}\) converges to \(\tau_k\) (the optimal value of SDP (2.10)) in finitely many steps.

The dual of (1.52) reads

\[
\rho^{\text{TS}}_{k, t} = \sup_{\xi_0, Q, U} \left\{ \left. \xi \right| \begin{array}{l}
\bar{Q}_t = G^{(i)}_{k, t} \circ \bar{Q}_i \geq 0, i \in [0] \cup [m], \\
\bar{U}_t = H^{(i)}_{k, t} \circ \bar{U}_i, i \in [l], \\
f - \xi = v_k^T Q_0 v_k + \sum_{i \in [m]} g_i v_k^T [-g_i] Q_i v_k^T [-g_i] + \sum_{i \in [l]} h_i v_k^T [-h_i] U_i v_k^T [-h_i] \right. \right\}. \tag{1.53}
\]

A.2.2 Correlative-Term Sparsity (CS-TS). The basic idea of correlative-term sparsity is to exploit term sparsity for each clique. The clique structure of the initial set of variables is derived from correlative sparsity (Section A.1.1).
Fix a relaxation order \( k \in \mathbb{N} \geq k_{\min} \). For every sparse order \( t \in \mathbb{N} \setminus \{0\} \) and for every \( j \in [p] \), we compute the following block diagonal (up to permutation) \((0, 1)\)-binary matrices (see Reference [46]):
\[
G_{k,t,j}^{(i)} \text{ of size } s(n_j, k); \quad H_{k,t,j}^{(i)} \text{ of size } s(n_j, k - [g_i]), \quad i \in J_j; \quad I_{k,t,j}^{(i)} \text{ of size } s(n_j, k - [h_i]), \quad i \in W_j.
\]

Then let us consider the following CS-TS moment relaxation:
\[
\tau_{k,t}^{cs-ts} := \inf_{y \in \mathbb{R}^{n(k)}(k)} \left\{ \mathcal{L}_y(f) \middle| G_{k,t,j}^{(i)} \circ M_k(y, l_j) \geq 0, \quad j \in [p], \quad y_0 = 1, \right. \quad \left. G_{k,t,j}^{(i)} \circ M_k - [g_i](y_j, l_j) \geq 0, \quad i \in J_j, \quad j \in [p], \right. \quad \left. H_{k,t,j}^{(i)} \circ M_k - [h_i](y_j, l_j) = 0, \quad i \in W_j, \quad j \in [p] \right\}.
\tag{1.54}
\]

One has \( \tau_{k,t-1}^{cs-ts} \leq \tau_{k,t}^{cs-ts} \leq \tau_k \leq f^* \), for all \((k, t)\). Moreover, we have the following theorem:

**Theorem A.7 (Wang et al. [46])**. For each \( k \in \mathbb{N} \geq k_{\min} \), the sequence \( (\tau_{k,t}^{cs})_{t \in \mathbb{N} \setminus \{0\}} \) converges to \( \tau_k^{ci} \) (the optimal value of SDP (1.36)) in finitely many steps.

The dual of (1.54) reads
\[
\rho_{k,t}^{cs-ts} = \sup_{\xi : Q_t^{(j)}(y) \leq 0, i \in [p]} \left\{ \xi \left| \begin{align*}
Q_t^{(j)} &= G_{k,t,j}^{(i)} \circ Q_t^{(j)} \geq 0, \quad i \in [0] \cup J_j, \quad j \in [p], \\
U_t^{(j)} &= H_{k,t,j}^{(i)} \circ U_t^{(j)} \geq 0, \quad i \in W_j, \quad j \in [p], \\
f - \xi &= \sum_{i \in [p]} \left( v^{(i)}_k \right)^{\top} Q_t^{(j)} v^{(i)}_k \\
&\quad + \sum_{i \in J_j} g_i (v^{(i)}_{k-[g_i]})^{\top} Q_t^{(j)} v^{(i)}_{k-[g_i]} \\
&\quad + \sum_{i \in W_j} h_i (v^{(i)}_{k-[h_i]})^{\top} Q_t^{(j)} v^{(i)}_{k-[h_i]} \end{align*} \right. \right\}.
\tag{1.55}
\]

### A.3 Numerical Experiments for Sparse POPs

In this section, we report results of numerical experiments for sparse POPs with the same settings and notations as in Section 4.

#### A.3.1 Randomly Generated QCQPs with TS and Ball Constraints.

**Test problems**: We construct randomly generated QCQPs with TS and a ball constraint as follows:

1. Generate a quadratic polynomial objective function \( f \) such that for \( \alpha \in \mathbb{N}^p \) with \(|\alpha| \neq 2\), \( f_\alpha = 0 \) and for \( \alpha \in \mathbb{N}_2^p \) with \(|\alpha| = 2\), the coefficient \( f_\alpha \) is randomly generated in \((-1, 1)\) w.r.t. the uniform distribution;
2. Take \( m = 1 \) and \( g_1 := 1 - \|x\|_2^2 \);
3. Take a random point \( a \) in \( S(g) \) w.r.t. the uniform distribution;
4. For every \( j \in [l] \), generate a quadratic polynomial \( h_j \) by
   - (i) setting \( h_{j,a} = 0 \) for each \( \alpha \in \mathbb{N}_2^p \setminus \{0\} \) with \(|\alpha| \neq 2\);
   - (ii) for each \( \alpha \in \mathbb{N}_2^p \setminus \{0\} \) with \(|\alpha| = 2\), taking a random coefficient \( h_{j,a} \) for \( h_j \) in \((-1, 1)\) w.r.t. the uniform distribution;
   - (iii) setting \( h_{j,0} := -\sum_{\alpha \in \mathbb{N}_2^p \setminus \{0\}} h_{j,a} a^\alpha \).

Then \( a \) is a feasible solution of POP (2.5).

The numerical results are displayed in Tables 14 and 15.

**Discussion**: The behavior of solvers is similar to that in the dense case.

#### A.3.2 Randomly Generated QCQPs with TS and Box Constraints.

**Test problems**: We construct randomly generated QCQPs with TS as in Section A.3.1, where the ball constraint is now replaced by box constraints. The numerical results are displayed in Tables 16 and 17.
### Table 14. Numerical Results for Minimizing a Random Quadratic Polynomial with TS on the Unit Ball

| POP size | SDP size | Mosek | CGAL |
|----------|----------|-------|------|
|          |          |       |      |
| $n$ | $s_{\text{max}}$ | $\zeta$ | val | time | val | time |
| 10 | 56 | 937 | -1.5681 | 4 | -1.5527 | 0.7 |
| 20 | 211 | 13,722 | -2.4275 | 36 | -2.3996 | 1 |
| 30 | 466 | 68,357 | -3.0748 | 1,930 | -3.0577 | 8 |
| 40 | 821 | 214,842 | - | - | -3.6999 | 20 |
| 50 | 1,276 | 523,177 | - | - | -4.1603 | 128 |
| 60 | 1,831 | 1,083,362 | - | - | -4.1914 | 655 |
| 70 | 2,486 | 2,005,397 | - | - | -4.9578 | 1,461 |
| 80 | 3,241 | 3,419,282 | - | - | -5.6452 | 7,253 |

POP size: $m = 1, l = 0$; Relaxation order: $k = 2$; Sparse order: $t = 1$; SDP size: $\omega = 4, a_{\text{max}}^\omega = 3.$

### Table 15. Numerical Results for Randomly Generated QCQPs with TS and a Ball Constraint

| POP size | SDP size | Mosek | CGAL |
|----------|----------|-------|------|
|          |          |       |      |
| $n$ | $l$ | $s_{\text{max}}$ | $\zeta$ | val | time | val | time |
| 10 | 3 | 56 | 1,105 | -0.60612 | 0.7 | -0.60550 | 2 |
| 20 | 5 | 211 | 14,777 | -2.3115 | 47 | -2.3097 | 17 |
| 30 | 8 | 466 | 72,085 | -2.8344 | 3,102 | -2.8321 | 112 |
| 40 | 10 | 821 | 223,052 | - | - | -3.4081 | 476 |
| 50 | 13 | 1,276 | 539,765 | - | - | -3.3552 | 1,845 |
| 60 | 15 | 1,831 | 1,110,827 | - | - | -3.5620 | 2,992 |

POP size: $m = 1, l = \lceil n/4 \rceil$; Relaxation order: $k = 2$; Sparse order: $t = 1$; SDP size: $\omega = 4, a_{\text{max}}^\omega = 3.$

### Table 16. Numerical Results for Minimizing a Random Quadratic Polynomial with TS on a Box

| POP size | SDP size | Mosek | CGAL |
|----------|----------|-------|------|
|          |          |       |      |
| $n$ | $\omega$ | $s_{\text{max}}$ | $\zeta$ | val | time | val | time |
| 10 | 22 | 56 | 1,441 | -1.0539 | 3 | -1.0519 | 14 |
| 20 | 42 | 211 | 17,731 | -1.3925 | 93 | -1.3802 | 161 |
| 30 | 62 | 466 | 81,871 | -2.2301 | 4392 | -2.2128 | 567 |
| 40 | 82 | 821 | 246,861 | - | - | -2.5209 | 1,602 |
| 50 | 102 | 1,276 | 585,701 | - | - | -3.0282 | 2,583 |
| 60 | 122 | 1,831 | 1,191,391 | - | - | -3.0470 | 10,858 |

POP size: $m = n, l = 0$; Relaxation order: $k = 2$; Sparse order: $t = 1$; SDP size: $a_{\text{max}} = 3.$

**Discussion:** Again, the behavior of solvers is similar to that in the dense case.

#### A.3.3 Randomly Generated QCQPs with CS and Ball Constraints on Each Clique of Variables.

**Test problems:** We construct randomly generated QCQPs with CS and ball constraints on each clique of variables as follows:

ACM Transactions on Mathematical Software, Vol. 48, No. 4, Article 40. Publication date: December 2022.
(1) Take a positive integer \( u, p := \lceil n/u \rceil + 1 \) and let

\[
I_j = \begin{cases} 
[u], & \text{if } j = 1, \\
[u(j - 1), \ldots, u], & \text{if } j \in \{2, \ldots, p - 1\}, \\
(u(p - 1), \ldots, n), & \text{if } j = p;
\end{cases}
\]

(1.56)

(2) Generate a quadratic polynomial objective function \( f = \sum_{j \in [p]} f_j \) such that for each \( j \in [p] \), \( f_j \in \mathbb{R}[x(I_j)]_2 \), and the coefficient \( f_{j, \alpha}, \alpha \in \mathbb{N}_2^{I_j} \) of \( f_j \) is randomly generated in \((-1, 1)\) w.r.t. the uniform distribution;

(3) Take \( m = p \) and \( g_j := -\|x(I_j)\|_2^2 + 1, j \in [m] \);

(4) Take a random point \( a \) in \( S(g) \) w.r.t. the uniform distribution;

(5) Let \( r := \lfloor l/p \rfloor \) and

\[
W_j := \begin{cases} 
((j - 1)r + 1, \ldots, (j)r), & \text{if } j \in [p - 1], \\
((p - 1)r + 1, \ldots, l), & \text{if } j = p.
\end{cases}
\]

(1.57)

For every \( j \in [p] \) and every \( i \in W_j \), generate a quadratic polynomial \( h_i \in \mathbb{R}[x(I_j)]_2 \) by

(a) for each \( \alpha \in \mathbb{N}_2^{I_j} \setminus \{0\} \), taking a random coefficient \( h_{i, \alpha} \) of \( h_i \) in \((-1, 1)\) w.r.t. the uniform distribution;

(b) setting \( h_{i, 0} := -\sum_{\alpha \in \mathbb{N}_2^{I_j} \setminus \{0\}} h_{j, \alpha} a^\alpha \).

Then \( a \) is a feasible solution of POP (2.5).

The numerical results are displayed in Tables 18 and 19.
Table 19. Numerical Results for Randomly Generated QCQPs with CS and Ball Constraints on Each Clique of Variables

| POP size | SDP size | Mosek | CGAL |
|----------|----------|-------|------|
| u        | p        | ω     | s^max | ζ     |
| 11       | 91       | 182   | 91    | 235,023 | -224.15 | 163 | -224.09 | 204 |
| 16       | 63       | 126   | 171   | 572,905 | -192.45 | 1,830 | -192.30 | 335 |
| 21       | 48       | 96    | 276   | 1,139,460 | -- | -- | -162.79 | 537 |
| 26       | 39       | 78    | 406   | 2,005,124 | -- | -- | -148.77 | 1,014 |
| 31       | 33       | 66    | 561   | 3,239,573 | -- | -- | -142.38 | 2,115 |
| 36       | 28       | 56    | 741   | 4,862,292 | -- | -- | -124.97 | 5,304 |

POP size: n = m = p = 1,000, l = 143, u^max = u + 1; Relaxation order: k = 2; SDP size: \( ω = 2p \), \( a^\text{max} = 3 \).

Table 20. Numerical Results for Minimizing a Random Quadratic Polynomial with CS and Box Constraints on Each Clique of Variables

| POP size | SDP size | Mosek | CGAL |
|----------|----------|-------|------|
| u        | p        | ω     | s^max | ζ     |
| 11       | 91       | 1,181 | 91    | 313,361 | -204.89 | 443 | -204.69 | 753 |
| 16       | 63       | 1,125 | 171   | 720,323 | -163.11 | 3,082 | -162.88 | 3,059 |
| 21       | 48       | 1,095 | 276   | 1,380,918 | -- | -- | -147.92 | 5,655 |
| 26       | 39       | 1,077 | 406   | 2,357,161 | -- | -- | -131.00 | 8,889 |

POP size: n = m = 1,000, l = 0, u^max = u + 1; Relaxation order: k = 2; Constant trace: \( a^\text{max} \in [3, 4] \).

Table 21. Numerical Results for QCQPs with CS and Box Constraints on Each Clique of Variables

| POP size | SDP size | Mosek | CGAL |
|----------|----------|-------|------|
| u        | p        | ω     | s^max | ζ     |
| 11       | 91       | 1,181 | 91    | 325,672 | -187.01 | 402 | -186.98 | 1,915 |
| 16       | 63       | 1,125 | 171   | 742,536 | -142.16 | 4,323 | -142.27 | 4,126 |
| 21       | 48       | 1,095 | 276   | 1,412,696 | -- | -- | -131.14 | 5,334 |
| 26       | 39       | 1,077 | 406   | 2,406,406 | -- | -- | -113.44 | 8,037 |

POP size: n = m = 1,000, l = 143, u^max = u + 1; Relaxation order: k = 2; Constant trace: \( a^\text{max} \in [3, 4] \).

Discussion: The number of variables is fixed as \( n = 1,000 \). We increase the clique size \( u \) so the number of variable cliques \( p \) decreases accordingly. Again, results in Tables 18 and 19 show that CGAL is faster and returns an optimal value of gap within 1% w.r.t. the one returned by Mosek (for \( u \leq 16 \)). Moreover, Mosek runs out of memory when \( u \geq 21 \).

A.3.4 Randomly Generated QCQPs with CS and Box Constraints on Each Clique of Variables.

Test problems: We construct randomly generated QCQPs with CS as in Section A.3.3, where ball constraints are now replaced by box constraints. Namely, in Step 3, we take \( m = n, g_j := -x_j^2 + 1/u, j \in [n] \).

The numerical results are displayed in Tables 20 and 21.

Discussion: The number of variables is fixed as \( n = 1,000 \). We increase the clique size \( u \) so the number of variable cliques \( p \) decreases accordingly. From results in Tables 18 and 19, one observes that when the largest size of variable cliques is relatively small (say, \( u \leq 11 \)), Mosek is the fastest
The behavior of solvers is similar to that in Section A.3.4. Here, we also emphasize that our framework is less efficient than interior-point methods for most benchmarks presented in Reference [46]. The two underlying reasons are that (1) the block size of the resulting SDP relaxations is small, in which case Mosek performs more efficiently, e.g., for the benchmarks from CGAL solver. However, when the largest size of variable cliques is relatively large (say, \( u \geq 21 \)), Mosek runs out of memory while CGAL still works well.

A.3.5 Randomly Generated QCQPs with CS-TS and Ball Constraints on Each Clique of Variables.

Test problems: We construct randomly generated QCQPs with CS-TS and ball constraints on each clique of variables as follows:

1. Take a positive integer \( u, p := \lceil n/u \rceil + 1 \) and let \((I_j)_{j \in [p]}\) be defined as in (1.56);
2. Generate a quadratic polynomial objective function \( f = \sum_{j \in [p]} f_j \) such that for each \( j \in [p], f_j \in \mathbb{R}[x(I_j)]_2 \) and the nonzero coefficient \( f_{j, \alpha} \) with \( \alpha \in \mathbb{N}_2^{|I_j|} \) and \(|\alpha| = 2\) is randomly generated in \((-1, 1)\) w.r.t. the uniform distribution;
3. Take \( m = p \) and \( g_j := -\|x(I_j)\|_2^2 + 1, j \in [m] \);
4. Take a random point \( a \) in \( S(g) \) w.r.t. the uniform distribution;
5. Let \( r := |I(p)\| \) and \((W_j)_{j \in [p]}\) be as in (1.57). For every \( j \in [p] \) and every \( i \in W_j \), generate a quadratic polynomial \( h_{i,j} \in \mathbb{R}[x(I_j)]_2 \) by

   a. for each \( \alpha \in \mathbb{N}_2^{|I_j|} \setminus \{0\} \) with \(|\alpha| \neq 2\), taking \( h_{i,j,\alpha} = 0 \);
   b. for each \( \alpha \in \mathbb{N}_2^{|I_j|} \) with \(|\alpha| = 2\), taking a random coefficient \( h_{i,j,\alpha} \) of \( h_{i,j} \) in \((-1, 1)\) w.r.t. the uniform distribution;
   c. setting \( h_{i,0,j} := -\sum_{\alpha \in \mathbb{N}_2^{|I_j|} \setminus \{0\}} h_{i,j,\alpha} a^\alpha \).

Then \( a \) is a feasible solution of POP (2.5).

The numerical results are displayed in Tables 22 and 23.

| Table 22. Numerical Results for Minimizing a Random Quadratic Polynomial with CS-TS and Ball Constraints on Each Clique of Variables |
| --- |
| POP size | SDP size | Mosek | CGAL |
| \( u \) | \( p \) | \( \omega \) | \( s^{\text{max}} \) | \( \zeta \) | val | time | val | time |
| 11 | 91 | 364 | 79 | 169,654 | -160.05 | 163 | -160.01 | 498 |
| 16 | 63 | 252 | 154 | 448,354 | -135.78 | 1,422 | -135.74 | 768 |
| 21 | 48 | 192 | 254 | 939,619 | - | - | -117.17 | 1,605 |
| 26 | 39 | 156 | 379 | 1,705,763 | - | - | -106.26 | 3,150 |

POP size: \( n = 1,000, m = p, l = 0, u^{\text{max}} = u + 1 \); Relaxation order: \( k = 2 \); Sparse order: \( t = 1 \); SDP size: \( a^{\text{max}} = 3 \).

| Table 23. Numerical Results for QCQPs with CS-TS and Ball Constraints on Each Clique of Variables |
| --- |
| POP size | SDP size | Mosek | CGAL |
| \( u \) | \( p \) | \( \omega \) | \( s^{\text{max}} \) | \( \zeta \) | val | time | val | time |
| 11 | 91 | 364 | 79 | 180,303 | -155.91 | 158 | -155.87 | 604 |
| 16 | 63 | 252 | 154 | 468,290 | 127.42 | 1,707 | -127.36 | 1,053 |
| 21 | 48 | 192 | 254 | 939,619 | - | - | -114.85 | 2,877 |
| 26 | 39 | 156 | 379 | 1,751,556 | - | - | -102.30 | 6,878 |

POP size: \( n = 1,000, m = p, l = 143, u^{\text{max}} = u + 1 \); Relaxation order: \( k = 2 \); Sparse order: \( t = 1 \); SDP size: \( a^{\text{max}} = 3 \).
Table 24. Numerical Results for Minimizing a Random Quadratic Polynomial with CS-TS and Box Constraints on Each Clique of Variables

| POP size | SDP size | Mosek | CGAL |
|----------|----------|-------|------|
|         |          | val   | time | val   | time |
| $u$  | $p$  | $\omega$ | $s^{\max}$ | $\xi$ |       |       |
| 11    | 91    | 2,362 | 79 | 248,335 | −126.15 | 151 | −126.04 | 1,982 |
| 16    | 63    | 2,250 | 154 | 601,081 | −100.75 | 2,225 | −100.64 | 7,323 |
| 21    | 48    | 2,190 | 254 | 1,191,001 | − | − | −87.804 | 10,734 |
| 26    | 39    | 2,154 | 379 | 2,080,265 | − | − | −81.908 | 20,294 |

POP size: $n = m = 1,000$, $l = 0$, $u^{\max} = u + 1$; Relaxation order: $k = 2$; Sparse order: $t = 1$; Constant trace: $a^{\max} \in [3, 4]$.

Table 25. Numerical Results for QCQPs with CS-TS and Box Constraints on Each Clique of Variables

| POP size | SDP size | Mosek | CGAL |
|----------|----------|-------|------|
|         |          | val   | time | val   | time |
| $u$  | $p$  | $\omega$ | $s^{\max}$ | $\xi$ |       |       |
| 11    | 91    | 2,362 | 79 | 258,984 | −114.53 | 325 | −114.27 | 482 |
| 16    | 63    | 2,250 | 154 | 621,017 | −96.199 | 4,450 | −96.079 | 1,245 |
| 21    | 48    | 2,190 | 254 | 1,220,027 | − | − | −83.013 | 8,204 |
| 26    | 39    | 2,154 | 379 | 2,126,058 | − | − | −74.532 | 27,600 |

POP size: $n = m = 1,000$, $l = 143$, $u^{\max} = u + 1$; Relaxation order: $k = 2$; Sparse order: $t = 1$; Constant trace: $a^{\max} \in [3, 4]$.

[46, Section 5.2], and (2) it is harder to find the constant trace, e.g., for the benchmarks from Reference [46, Section 5.4]. Thus, our proposed method complements that in Reference [46] when the block size of the SDP relaxations is large and/or when CTP can be efficiently verified.

A.3.6 Randomly Generated QCQPs with CS-TS and Box Constraints on Each Clique of Variables.

Test problems: We construct randomly generated QCQPs with CS-TS as in Section A.3.5, where ball constraints are now replaced by box constraints. Namely, in Step 3, we take $m = n$, $g_j := −x_j^2 + 1/u$, $j \in [n]$. The numerical results are displayed in Tables 24 and 25.

Discussion: The behavior of solvers is similar to that in Section A.3.4.

A.4 Conditional Gradient-based Augmented Lagrangian

A.4.1 SDP with CTP. Let $s, l, s^{(j)} \in \mathbb{N}^\Xi$, $j \in [\omega]$ be fixed such that $s = \sum_{j=1}^{\omega} s^{(j)}$. Let $S$ be the set of real symmetric matrices of size $s$ in a block diagonal form: $X = \text{diag}(X_1, \ldots, X_\omega)$, such that $X_j$ is a block of size $s^{(j)}$, $j \in [\omega]$. Let $s^{\max} := \max_{j \in [\omega]} s^{(j)}$. Let $S^+$ be the set of all $X \in S$ such that $X \succeq 0$, i.e., $X$ has only nonnegative eigenvalues. Then $S$ is a Hilbert space with scalar product $\langle A, B \rangle = \text{trace}(B^TA)$ and $S^+$ is a self-dual cone.

Let us consider the following SDP:

$$
\tau = \inf_{X \in S^+} \{ \langle C, X \rangle : \mathcal{A}X = b \},
$$

(1.58)

where $\mathcal{A} : S \to \mathbb{R}^\xi$ is a linear operator of the form $\mathcal{A}X = [\langle A_1, X \rangle, \ldots, \langle A_\xi, X \rangle]$, with $A_i \in S$, $i \in [\xi]$, $C \in S$ is the cost matrix and $b \in \mathbb{R}^\xi$ is a vector.

The dual of SDP (1.58) reads

$$
\rho = \sup_{y \in \mathbb{R}^\xi} \{ b^T y : \mathcal{A}^T y - C \in S^+ \},
$$

(1.59)

where $\mathcal{A}^T : \mathbb{R}^\xi \to S$ is the adjoint operator of $\mathcal{A}$, i.e., $\mathcal{A}^T y = \sum_{i \in [\xi]} y_i A_i$.  

ACM Transactions on Mathematical Software, Vol. 48, No. 4, Article 40. Publication date: December 2022.
The following assumption will be used later on:

**ASSUMPTION A.8.** Consider the following conditions:

1. Strong duality of primal–dual (1.58)–(1.59) holds, i.e., \( \rho = \tau \) and \( \rho \in \mathbb{R} \).
2. Constant trace property (CTP): \( \exists \alpha > 0 : \forall X \in S, AX = b \Rightarrow \text{trace}(X) = a \).

For \( X \in S \), the Frobenius norm of \( X \) is defined by \( \|X\|_F = \sqrt{\text{trace}(XX^\top)} \). We denote by \( \|A\| \) the operator norm of \( A \), i.e., \( \|A\| := \max_{X \in S} \|AX\|_2/\|X\|_F \). The smallest eigenvalue of a real symmetric matrix \( D \) is denoted by \( \lambda_{\min}(D) \).

**Algorithm.** In Reference [49], Yurtsever et al. stated Algorithm 3 (see below) to solve SDP (1.58) with CTP. This procedure is based on the augmented Lagrangian paradigm combined together with the conditional gradient method.

### ALGORITHM 3: CGAL-SDP-CTP

**Input:** SDP (1.58) such that Assumption A.8 holds; Parameter \( K > 0 \).

**Output:** \((X_t)_{t \in \mathbb{N}}\) in.

1. Set \( X_0 := 0_S \) and \( y_0 := 0_{2\epsilon} \).
2. for \( t \in \mathbb{N} \) do
3. \( \beta_t := \sqrt{t + 1} + \eta_t := 2/(t + 1) \);
4. Take an eigenvector \( u_t \) corresponding to \( \lambda_{\min}(C + A^\top(y_{t-1} + \eta_t(AX_{t-1} - b))) \);
5. Set \( X_t := (1 - \eta_t)X_{t-1} + \eta_tau_t^u_t \);
6. Select \( y_t \) as the largest \( y \in [0, 1] \) such that:

\[
|\|AX_t - b\|_2^2 - \beta_t \eta_t a^2 \|AX\|^2| + \|y_{t-1} + y(AX_t - b)\|_2 \leq K;
\]
7. Set \( y_t = y_{t-1} + \gamma_t(AX_t - b) \).

The convergence of the sequence \((X_t)_{t \in \mathbb{N}}\) in Algorithm 3 to the set of optimal solutions of SDP (1.58) is guaranteed as follows:

**Theorem A.9 ([49, Fact 3.1]).** Consider SDP (1.58) such that Assumption A.8 holds. Let \((X_t)_{t \in \mathbb{N}}\) be in the output of Algorithm 3. Then \( X_t \geq 0 \), for all \( t \in \mathbb{N} \) and \( \|AX_t - b\|_2 \to 0 \), \( |\langle C, X_t \rangle - \tau| \to 0 \) as \( t \to \infty \), with the rate of order \( O(\sqrt{t}) \).

**Remark A.10.** To achieve the best convergence rate for Algorithm 3, we scale the problem's input as follows: \( \|C\|_F = \|A\| = a = 1 \) and \( \|A_1\|_F = \cdots = \|A_K\|_F \).

**Remark A.11.** Given \( \epsilon > 0 \), the for loop in Algorithm 3 terminates when:

\[
\frac{|\langle C, X_{t-1} \rangle - (a\lambda_{\min}(C + A^\top(y_{t-1} + \eta_t(AX_{t-1} - b))) - b^\top y_{t-1})|}{1 + \max\{|\langle C, X_{t-1} \rangle|, |a\lambda_{\min}(C + A^\top(y_{t-1} + \eta_t(AX_{t-1} - b))) - b^\top y_{t-1}|\}} \leq \epsilon \tag{1.60}
\]

and \( \|AX_{t-1} - b\|_2/\max\{1, \|b\|_2\} \leq \epsilon \). In our experiments, we choose \( \epsilon = 10^{-3} \). Note that the left-hand side in (1.60) is the relative gap between the primal and dual approximate values obtained at each iteration.

**Remark A.12.** To save memory at each iteration, we can run Algorithm 3 with an implicit \( X_t \) by setting \( w_t := AX_t - b \). In this case, Step 5 becomes \( w_t := (1 - \eta_t)w_{t-1} + \eta_t[A(au_tu_t^\top) - b] \). Thus, we only obtain an approximate dual solution \( y_t \) of SDP (1.58) when Algorithm 3 terminates.

In Appendix A.4.2, we provide an analogous method to solve an SDP with CTP on each subset of blocks.
A.4.2 SDP with CTP on Each Subset of Blocks. Let \( p \in \mathbb{N}^{\geq 1}, s_j, \omega_j \in \mathbb{N}, j \in [p], \) and \( s^{(i,j)} \in \mathbb{N}^{\geq 1}, i \in [\omega_p], j \in [p], \) be fixed such that \( s_j = \sum_{i \in [\omega_j]} s^{(i,j)}, j \in [p]. \) For every \( j \in [p], \) let \( S_j \) be the set of real symmetric matrices of size \( s_j \) in a block diagonal form: \( X_j = \text{diag}(X_{1,j}, \ldots, X_{\omega_j,j}), \) such that \( X_{i,j} \) is a block of size \( s^{(i,j)}, i \in [\omega_p]. \) Let \( s^\text{max} := \max_{i \in [\omega_p], j \in [p]} s^{(i,j)}. \) For every \( j \in [p], \) let \( S_j^+ \) be the set of all \( X_j \in S_j \) such that \( X_j \succeq 0. \) Then for every \( j \in [p], S_j \) is a Hilbert space with scalar product \( \langle A, B \rangle = \text{trace}(B^\top A) \) and \( S_j^+ \) is a self-dual cone.

Let us consider the following SDP:

\[
\tau = \inf_{X_j \in S_j^+} \left\{ \sum_{j \in [p]} \langle C_j, X_j \rangle : \sum_{j \in [p]} A_j X_j = b \right\},
\]  

(1.61)

where \( A_j : S_j \to \mathbb{R}^{\xi_j} \) is a linear operator of the form \( A_j X = [\langle A_{1,j}, X \rangle, \ldots, \langle A_{\xi_j,j}, X \rangle], \) with \( A_{i,j} \in S_j, i \in [\xi_j], C_j \in S_j, j \in [p], \) and \( b \in \mathbb{R}^{\xi_j}. \)

The dual of SDP (1.61) reads

\[
\rho = \sup_{y \in \mathbb{R}^{\xi_j}} \left\{ b^\top y : A_j^\top y - C_j \in S_j^+, j \in [p] \right\},
\]  

(1.62)

where \( A_j^\top : S_j \to S_j^+ \) is the adjoint operator of \( A_j, \) i.e., \( A_j^\top z = \sum_{i \in [\xi_j]} z_i A_{i,j}, j \in [p]. \)

The following assumption will be used later on:

**Assumption A.13. Consider the following conditions:**

1. **Strong duality of primal-dual (1.61)–(1.62)** holds, i.e., \( \rho = \tau \) and \( \rho \in \mathbb{R}. \)
2. **Constant trace property (CTP):** there exist \( a_j > 0 \) and \( j \in [p], \) such that

\[
\forall X_j \in S_j, j \in [p], \quad \sum_{j \in [p]} A_j X_j = b \quad \Rightarrow \quad \text{trace}(X_j) = a_j, j \in [p].
\]  

(1.63)

Recall that \( \lambda_{\min}(D) \) stands for the smallest eigenvalue of a real symmetric matrix \( D. \) We denote by \( \prod_{j \in [p]} S_j \) the set of all \( X = \text{diag}(X_j) \in [p] \) such that \( X_j \in S_j, j \in [p]. \) Let \( C := \text{diag}(C_j) \in [p] \) and let \( A : \prod_{j \in [p]} S_j \to \mathbb{R}^{\xi_j} \) be a linear operator of the form \( A X = \sum_{j \in [p]} A_j X_j, \) for all \( X = \text{diag}(X_j) \in \prod_{j \in [p]} S_j. \) Then for every \( X = \text{diag}(X_j) \in \prod_{j \in [p]} S_j, \) we have \( \langle C, X \rangle = \sum_{j \in [p]} \langle C_j, X_j \rangle \) and \( A X = [\langle A^{(1)}, X \rangle, \ldots, \langle A^{(\xi_j)}, X \rangle], \) where \( A^{(i)} := \text{diag}((A_{i,j})_{j \in [p]}), \) for \( i \in [\xi]. \)

SDP (1.61) can be rewritten as \( \tau = \inf_{X \in \prod_{j \in [p]} S_j} \left\{ \langle C, X \rangle : A X = b \right\}. \)

The dual operator \( A^\top : \mathbb{R}^{\xi_j} \to \prod_{j \in [p]} S_j \) of \( A \) reads \( A^\top z = \text{diag}((A_j^\top z_j)_{j \in [p]}). \) Note \( \Delta_j := \{ X_j \in S_j^+ : \text{trace}(X_j) = a_j \}, j \in [p]. \)

**Algorithm.** To solve SDP (1.61) with CTP on each subset of blocks, we use Reference [48, Algorithm 1] due to Yurtsever et al. to describe Algorithm 4 with the following setting: \( X \leftarrow \Delta := \prod_{j \in [p]} \Delta_j, \quad K \leftarrow \{ b \}, \quad p \leftarrow \xi, \quad A_{x} \leftarrow A X, \quad f(x) \leftarrow \langle C, X \rangle, \quad \lambda_0 \leftarrow 1, \quad \beta_k \leftarrow \sigma_k \leftarrow \gamma_k, \quad D_{\hat{x}_{k+1}} \leftarrow K, \quad L_f \leftarrow 0, \quad r_{k+1} \leftarrow b, \quad D_X^2 \leftarrow 2 \sum_{j \in [p]} a_j^2, \quad v_k \leftarrow C + A^\top z_k, \quad \text{arg min}_{x \in D}(v_k, x) \leftarrow \arg \min_{x \in \Delta}(C + A^\top z_k, X). \)

With fixed \( z_k, \) we have:

\[
\min_{X \in \Delta}(C + A^\top z_k, X) = \min_{\text{diag}((X_j)_{j \in [p]}) \in \prod_{j \in [p]} \Delta_j, j \in [p]} \sum_{j \in [p]} \langle C_j + A_j^\top z_k, X_j \rangle = \sum_{j \in [p]} \min_{X_j \in \Delta_j} \langle C_j + A_j^\top z_k, X_j \rangle = \sum_{j \in [p]} a_j\lambda_{\min}(C_j + A_j^\top z_k).
\]
Let \( u_k^{(j)} \) be a uniform eigenvector corresponding to \( \lambda_{\min}(C_j + A_j^T z_k) \), for \( j \in [p] \). Then one has \( \text{diag}((a_j u_k^{(j)} (u_k^{(j)})^T)_{j \in [p]}) \in \arg\min_{X \succeq 0} \langle C + A_j^T z_k, X \rangle \). Thus, we can set \( s_k \leftarrow \text{diag}((a_j u_k^{(j)} (u_k^{(j)})^T)_{j \in [p]}) \) in Reference [48, Algorithm 1].

**Algorithm 4: CGAL-SDP-CTP-Blocks**

**Input:** SDP (1.61) such that Assumption A.13 holds; Parameter \( K > 0 \).

**Output:** \((X_j^{(t)})_{j \in [p]} \) for each \( t \in \mathbb{N} \).

1. Set \((X_j^{(0)})_{j \in [p]} := (0_{S_j})_{j \in [p]} \) and \( y_0 := 0_{R^e} \).
2. for \( t \in \mathbb{N} \) do
3. Set \( \beta_t := \sqrt{t + 1} \) and \( \eta_t := 2/(t + 1) \);
4. Set \( z_t := y_{t-1} + \eta_t \sum_{j \in [p]} A_j X_j^{(t-1)} - b \);
5. for \( j \in [p] \) do
6. Take a uniform eigenvector \( u_j^{(t)} \) corresponding to \( \lambda_{\min}(C_j + A_j^T z_t) \);
7. Set \( X_j^{(t)} := (1 - \eta_t) X_j^{(t-1)} + \eta_t a_j u_j^{(t)} (u_j^{(t)})^T \);
8. Select \( \gamma_j \) as the largest \( \gamma \in [0, 1] \) such that:
9. \( \| \sum_{j \in [p]} A_j X_j^{(t)} - b \|_2^2 \leq \beta_t \sum_{j \in [p]} \| A_j \|_2^2 \| X_j^{(t)} - b \|_2^2 \leq K \);
10. Set \( y_t := y_{t-1} + \gamma_t \sum_{j \in [p]} A_j X_j^{(t)} - b \).

Relying on Reference [48, Theorem 3.1], we guarantee the convergence of the sequence \((X_j^{(t)})_{j \in [p]} \) in Algorithm 4 to the set of optimal solutions of SDP (1.61) in the following theorem:

**Theorem A.14.** Consider SDP (1.61) such that Assumption A.13 holds. Let \((X_j^{(t)})_{j \in [p]} \) be the output of Algorithm 4. Then \( X_j^{(t)} \succeq 0 \), for all \( j \in [p] \) and for all \( t \in \mathbb{N} \) and \( \| \sum_{j \in [p]} A_j X_j^{(t)} - b \|_2 \to 0 \) and \( | \sum_{j \in [p]} (C_j, X_j^{(t)}) - \tau | \to 0 \) as \( t \to \infty \) with the rate \( O(\sqrt{t}) \).

**Remark A.15.** Before running Algorithm 4, we scale the problem’s input as follows:
\[ \| C \|_F = \| A \| = a_1 = \cdots = a_p = 1 \text{ and } \| A \|^2 \| C \|_F = \cdots = \| A \|^{(n)} \| F. \]

**Remark A.16.** Given \( \varepsilon > 0 \), the for loop in Algorithm 4 terminates when:
\[ \frac{1}{1 + \max(| \sum_{j \in [p]} (C_j, X_j^{(t-1)}) |, | \sum_{j \in [p]} (a_j \lambda_{\min}(C_j + A_j^T z_t) - b^T y_{t-1}) |)} \leq \varepsilon \]
and \( \| \sum_{j \in [p]} A_j X_j^{(t-1)} - b \|_2 / \max(1, \| b \|_2) \leq \varepsilon \). In our experiments, we choose \( \varepsilon = 10^{-2} \).

**Remark A.17.** To save memory at each iteration, we can run Algorithm 4 with implicit \( X_j^{(t)} \), \( j \in [p] \), by setting \( w_t := \sum_{j \in [p]} A_j X_j^{(t)} - b \). In this case, Step 7 becomes \( w_t := (1 - \eta_t) w_{t-1} + \eta_t \sum_{j \in [p]} A_j (a_j u_j^{(t)} (u_j^{(t)})^T - b) \). Thus, we only obtain an approximate dual solution \( y_t \) of SDP (1.61) when Algorithm 4 terminates.

### A.5 Spectral Method

**A.5.1 SDP with CTP.** Consider SDP with CTP described in Appendix A.4. The following assumption will be used later on.

**Assumption A.18.** Dual attainability: SDP (1.59) has an optimal solution.
Lemma A.19. Let Assumption A.8 hold and let \( \varphi : \mathbb{R}^\mathcal{X} \to \mathbb{R} \) be a function defined by \( y \mapsto \varphi(y) := a\lambda_{\text{min}}(C - \mathcal{A}^T y) + b^T y \). Then,
\[
\tau = \sup_{y \in \mathbb{R}^\mathcal{X}} \varphi(y).
\] (1.64)
Moreover, if Assumption A.18 holds, then problem (1.64) has an optimal solution.

Notice that \( \varphi \) in Lemma A.19 is concave and continuous but not differentiable in general. The subdifferential of \( \varphi \) at \( y \) reads:
\[
\partial \varphi(y) = \{ b - a\mathcal{A}U : U \in \text{conv}(\Gamma(C - \mathcal{A}^T y)) \},
\]
where for each \( A \in \mathcal{S} \),
\[
\Gamma(A) := \{ uu^T : Au = \lambda_{\text{min}}(A)u, \|u\|_2 = 1 \}.
\]

Next, we describe Algorithm 5 to solve SDP (1.58), which is based on nonsmooth first-order optimization methods (e.g., LMBM [13, Algorithm 1]).

**ALGORITHM 5:** Spectral-SDP-CTP

**Input:** SDP (1.58) with unknown optimal value and optimal solution; method (T) for solving convex nonsmooth unconstrained optimization problems (NSOP).

**Output:** the optimal value \( \tau \) of SDP (1.58).

1. Compute the optimal value \( \tau \) and an optimal solution \( \tilde{y} \) of the NSOP (1.64) by using method (T).

Corollary A.20. Let Assumption A.8 hold. Assume that the method (T) is globally convergent for NSOP (1.64) (e.g., (T) is LMBM). Then output \( \tau \) of Algorithm 5 is well-defined. Moreover, if Assumption A.18 holds, then the vector \( \tilde{y} \) mentioned at Step 1 of Algorithm 5 exists.

A.5.2 SDP with CTP on Each Subset of Blocks. Consider SDP with CTP on each subset of blocks described in Appendix A.4.2.

The following assumption will be used later on:

**Assumption A.21.** Dual attainability: SDP (1.62) has an optimal solution.

Lemma A.22. Let Assumption A.13 hold and let \( \psi : \mathbb{R}^\mathcal{X} \to \mathbb{R} \) be a function defined by \( y \mapsto \psi(y) := b^T y + \sum_{j \in [p]} a_j \lambda_{\text{min}}(C_j - \mathcal{A}_j^T y) \). Then,
\[
\tau = \sup_{y \in \mathbb{R}^\mathcal{X}} \psi(y).
\] (1.65)
Moreover, if Assumption A.21 holds, then problem (1.65) has an optimal solution.

Proof. From (1.61) and Condition 4 of Assumption A.13,
\[
\tau = \inf_{X_j \in S_j^+} \left\{ \sum_{j \in [p]} \langle C_j, X_j \rangle \left| \begin{array}{c}
\sum_{j \in [p]} \mathcal{A}_j X_j = b, \\
\langle I_j, X_j \rangle = a_j, j \in [p]
\end{array} \right. \right\},
\] (1.66)
where \( I_j \in S_j \) is the identity matrix, for \( j \in [p] \). Note that \( \langle I_j, X_j \rangle = \text{trace}(X_j) \), for \( X_j \in S_j, j \in [p] \).

The dual of this SDP reads
\[
\rho = \sup_{(\xi, y) \in \mathbb{R}^{p+\mathcal{X}}} \left\{ \sum_{j \in [p]} a_j \xi_j + b^T y : C_j - \mathcal{A}_j^T y - \xi I_j \in S_j^+, j \in [p] \right\}.
\] (1.67)
It implies that \( \rho = \sup_{\xi, y} \{ \sum_{j \in [p]} a_j \xi_j + b^T y : \xi_j \leq \lambda_{\text{min}}(C_j - \mathcal{A}_j^T y), j \in [p] \} \). From this, the result follows, since \( \rho = \tau \).

Proposition A.23. The function \( \psi \) in Lemma A.22 has the following properties:
(1) $\psi$ is concave and continuous but not differentiable in general.

(2) The subdifferential of $\psi$ at $y$ satisfies: $\partial \psi(y) = b + \sum_{j \in [p]} a_j \partial \psi_j(y)$, where for every $j \in [p]$, $\psi_j : \mathbb{R}^r \to \mathbb{R}$ is a function defined by $\psi_j(y) = \lambda_{\min}(C_j - A^T_j y)$ and $\partial \psi_j(y) = \{ -A_j U : U \in \text{conv}(\Gamma(C_j - A^T_j y)) \}$.

Proof. It is not hard to prove the first statement. Indeed, $\psi$ is a positive combination of $z \mapsto h^T z$, $\psi_j, j \in [p]$, which are convex, continuous functions. The second statement follows by applying the subdifferential sum rule and notice that the domains of $z \mapsto h^T z$, $\psi_j, j \in [p]$, are both $\mathbb{R}^n$. $\Box$

Next, we describe Algorithm 6 to solve SDP (1.61), which is based on nonsmooth first-order optimization methods (e.g., LMBM [13, Algorithm 1]).

**Algorithm 6:** Spectral-SDP-CTP-Blocks

**Input:** SDP (1.61) with unknown optimal value and optimal solution; method (T) for solving NSOP.

**Output:** the optimal value $\rho$ of SDP (1.61).

1. Compute the optimal value $\tau$ and an optimal solution $\bar{y}$ of the NSOP (1.65) by using method (T).

The fact that Algorithm 6 is well-defined under certain conditions is a corollary of Lemma A.22 and Reference [30, Lemma A.2].

**Corollary A.24.** Let Assumption A.13 hold. Assume that the method (T) is globally convergent for NSOP (1.65) (e.g., (T) is LMBM). Then output $\tau$ of Algorithm 6 is well-defined. Moreover, if Assumption A.21 holds, then the vector $\bar{y}$ involved at Step 1 of Algorithm 6 exists.

**A.6 Converting the Moment Relaxation to the Standard SDP**

**A.6.1 The Dense Case.** Let $k \in \mathbb{N}_{\geq k_{\text{min}}}$ be fixed. We will present a way to transform SDP (3.11) to the form (3.13). By adding slack variables $y^{(i)} \in \mathbb{R}^{s(2k - [g_i])}$, $i \in [m]$, SDP (3.11) is equivalent to

$$
\tau_k := \inf_{y^{(i)}} \{ L_k(f) \mid \begin{align*}
W_k(y,y^{(1)},\ldots,y^{(m)}) &\in S_k^+, \\
M_{k-[g_i]}(y^{(i)}) &\in M_{k-[g_i]}(g_i y), i \in [m], \\
M_{k-[h_i]}(h_i y) &\in \mathbb{R}^{s(2k-[g_i])}, i \in [l] \}
\}.
$$

(1.68)

where $W_k(y,y^{(1)},\ldots,y^{(m)}) := \text{diag}(M_k(y), M_{k-[g_i]}(y^{(i)}), \ldots, M_{k-[g_m]}(y^{(m)}))$.

Let $V = \{M_k(z) : z \in \mathbb{R}^{s(2k)} \}$ and $V_i = \{M_{k-[g_i]}(z) : z \in \mathbb{R}^{s(2k-[g_i])} \}, i \in [m]$. Then $V$ and $V_i, i \in [m]$ are the linear subspaces of the matrices of size $s(k)$ and $s(k-[g_i]), i \in [m]$, respectively.

Denote by $V^\perp$, $V_i^\perp, i \in [m]$ the orthogonal complements of $V$, $V_i$, $i \in [m]$, respectively. In Reference [30, Appendix A.2], we show how to take a basis $\{A_j\}_{j \in [r]}$ of $V^\perp$. Similarly, we can take a basis $\{\hat{A}_j\}_{j \in [r]}$ of $V_i^\perp, i \in [m]$. Here, $r = \dim(V^\perp)$ and $r_i = \dim(V_i^\perp), i \in [m]$.

Notice that if $X_0$ is a real symmetric matrix of size $s(k)$, then $X_0 = M_k(y)$ for some $y \in \mathbb{R}^{s(2k)}$ if and only if $\langle A_j, X_0 \rangle = 0, j \in [r]$. It implies that if $X = \text{diag}(X_0, \ldots, X_m) \in S_k$, then there exist $y$ and $y^{(i)}, i \in [m]$, such that $X = W_k(y,y^{(1)},\ldots,y^{(m)}) \iff \langle A, X \rangle = 0, A \in B_1$, where $B_1$ involves matrices $A$ defined as

- $A = \text{diag}(A_j, 0, \ldots, 0)$ for some $j \in [r];$
- $\hat{A} = \text{diag}(0, \hat{A}_j^{(1)}, \ldots, 0)$ for some $j \in [r_1];$
- $\ldots$
- $\hat{A} = \text{diag}(0, 0, \ldots, \hat{A}_j^{(m)})$ for some $j \in [r_m].$
Notice that
\[
|B_1| = r + \sum_{i \in [m]} r_i = \frac{s(k)(s(k) + 1)}{2} - s(2k) + \sum_{i \in [m]} \left( \frac{s(k - [gi])(s(k - [gi]) + 1)}{2} - s(2(k - [gi])) \right).
\] (1.69)

The constraints \( M_{k-[gi]}(y^{(i)}) = M_{k-[gi]}(g_i y), \) \( i \in [m], \) of SDP (1.68) are equivalent to \( y_\alpha = \sum_{y \in \mathbb{N}^n_{2k-[gi]}} g_i y_{\alpha + y}, \) \( \alpha \in \mathbb{N}^n_{2(k-[gi])}, \) \( i \in [m]. \) They can be written as \( \langle \tilde{A}, W_k(y, y^{(1)}, \ldots, y^{(m)}) \rangle = 0, \) for \( \tilde{A} \in B_2, \) where \( B_2 \) involves matrices \( \tilde{A} \) defined by \( \tilde{A} = \text{diag}(\tilde{A}, 0, \ldots, 0, \tilde{A}^{(i)}, 0, \ldots, 0), \) with \( \tilde{A} = (\tilde{A}_{\mu, \nu})_{\mu, \nu \in \mathbb{N}^n_{k}} \) being defined as follows:
\[
\tilde{A}_{\mu, \nu} = \begin{cases} 
\frac{1}{2} g_{i, \nu} & \text{if } \mu = \nu, \mu + \nu = \alpha + \gamma, \\
\frac{1}{2} & \text{if } \mu \neq \nu, (\mu, \nu) \in \{(\mu_1, \nu_1), (\nu_1, \mu_1)\} \\
0 & \text{otherwise}
\end{cases}
\] (1.70)

and \( \tilde{A}^{(i)} = (\tilde{A}_{\mu, \nu}^{(i)})_{\mu, \nu \in \mathbb{N}^n_{k-[gi]}} \) being defined as follows:
\[
\tilde{A}_{\mu, \nu}^{(i)} = \begin{cases} 
\frac{1}{2} g_{i, \nu} & \text{if } \mu = \nu, \mu + \nu = \alpha, \\
\frac{1}{2} & \text{if } \mu \neq \nu, (\mu, \nu) \in \{(\mu_1, \nu_1), (\nu_1, \mu_1)\} \\
0 & \text{otherwise}
\end{cases}
\] (1.71)

for some \( \alpha \in \mathbb{N}^n_{2(k-[gi])} \) and \( i \in [m]. \) Notice that \( |B_2| = \sum_{i \in [m]} 2(k - [gi]). \) Here, minimal(\( T \)) is the minimal element of \( T, \) for every \( T \subseteq \mathbb{N}^{2n} \) with respect to the graded lexicographic order.

The constraints \( M_{k-[hi]}(h_j y) = 0, j \in [l], \) can be simplified as \( \sum_{y \in \mathbb{N}^n_{2(h_j)}} h_{j, \gamma} y_{\alpha + \gamma} = 0, \) \( \alpha \in \mathbb{N}^n_{2(k-[hi])}, \) \( j \in [l]. \) They are equivalent to the following trace equality constraints:
\( \langle \tilde{A}, W_k(y, y^{(1)}, \ldots, y^{(m)}) \rangle = 0, \) \( \tilde{A} \in B_3, \) where \( B_3 \) involves matrices \( \tilde{A} = \text{diag}(\tilde{A}, 0, \ldots, 0), \) with \( \tilde{A} = (\tilde{A}_{\mu, \nu})_{\mu, \nu \in \mathbb{N}^n_{k}} \) being defined as follows:
\[
\tilde{A}_{\mu, \nu} = \begin{cases} 
h_{j, \nu} & \text{if } \mu = \nu, \mu + \nu = \alpha + \gamma, \\
\frac{1}{2} h_{j, \nu} & \text{if } \mu \neq \nu, (\mu, \nu) \in \{(\mu_1, \nu_1), (\nu_1, \mu_1)\} \\
0 & \text{otherwise}
\end{cases}
\]

Notice that \( |B_3| = \sum_{j \in [l]} 2(k - [h_j]). \) Let \( \cup_{j \in [3]} B_j = (A_i)_{i \in [\zeta_k - 1]}, \) where
\[
\zeta_k = 1 + \sum_{j \in [3]} |B_j| = 1 + \frac{s(k)(s(k) + 1)}{2} - s(2k) + \sum_{i \in [m]} \left( \frac{s(k - [gi])(s(k - [gi]) + 1)}{2} - s(2(k - [gi])) \right).
\]

The final constraint \( y_0 = 1 \) can be rewritten as \( \langle \tilde{A}_{\zeta_k}, W_k(y, y^{(1)}, \ldots, y^{(m)}) \rangle = 1 \) with \( \tilde{A}_{\zeta_k} \in S_k \) having zero entries except the top left one \( [A_{\zeta_k}]_{0,0} = 1. \) Thus, we select real vector \( b_k \) of length \( t_k \) such that all entries of \( b_k \) are zeros except the final one \( b_{\zeta_k} = 1. \)
The function \( L_y(f) = \sum_y f_y y \) is equal to \( \langle C, W_k(y, y^{(1)}, \ldots, y^{(m)}) \rangle \) with \( C := \text{diag}(C, 0, \ldots, 0) \), where \( C = (C_{\mu, \nu})_{\mu, \nu \in \mathbb{N}_k^m} \) is defined by

\[
C_{\mu, \nu} = \begin{cases} f_y & \text{if } \mu = \nu, \mu + v = y, \\
\frac{1}{2} f_y & \text{if } \mu \neq \nu, (\mu, \nu) \in \{(\mu_1, \nu_1), (\nu_1, \mu_1)\} \\
0 & \text{otherwise.}
\end{cases}
\]

By writing \( X = W_k(y, y^{(1)}, \ldots, y^{(m)}) \), SDP (1.68) has the standard form

\[
\tau_k = \inf_{X \in S_k^L} \{ \langle C, X \rangle : \bar{A}X = b_k \},
\]

where \( \bar{A} : S_k \rightarrow \mathbb{R}^{\tilde{\mu}_k} \) is a linear operator of the form \( \bar{A}X = \left[ \langle \bar{A}_1, X \rangle, \ldots, \langle \bar{A}_k, X \rangle \right] \). Since \( (U, V) = \langle P_k^{-1} U P_k^{-1}, P_k V P_k \rangle \), for all \( U, V \in S_k \), by noting \( X = P_k X P_k \), SDP (1.72) can be written as (3.13) with \( A_k,i = P_k^{-1} A P_k^{-1}, i \in [\tilde{\mu}_k], \) and \( C_k = P_k^{-1} \bar{C} P_k^{-1} \).

### A.6.2 The Sparse Case.

Let \( k \in \mathbb{N}^{\geq k_{\text{min}}} \) be fixed. We will present a way to transform SDP (1.37) to the form (1.41). Doing a similar process as in Appendix A.6.1 on every clique, by noting (1.40), for every \( j \in [p] \), the constraints

\[
\begin{align*}
D_k(y, I_j) &\geq 0, \quad y_0 = 1, \\
M_{k-[h_k]}(h_k y, I_j) & = 0, \quad i \in W_j,
\end{align*}
\]

(1.73) become \( \tilde{A}_jX_j = \tilde{b}_j \) for some linear operator \( \tilde{A}_j : S_{j,k} \rightarrow \mathbb{R}^{\tilde{\mu}_j} \) and vector \( \tilde{b}_j \in \mathbb{R}^{\tilde{\mu}_j} \). Moreover, \( L_y(f_j) = \langle C_j, X_j \rangle \) for some matrix \( C_j \in S_{j,k} \), since \( f_j \in \mathbb{R}[x(I_j)] \), for every \( j \in [p] \). Then from (1.40), the objective function of SDP (1.37) is \( L_y(f) = \sum_{j \in [p]} \langle C_j, X_j \rangle \).

Next, we describe the constraints depending on common moments on cliques. For every \( \alpha \in \cup_{j \in [p]} \mathbb{N}_k^{l_j} \), note \( T(\alpha) := \{ j \in [p] : \alpha \in \mathbb{N}_k^{l_j} \} \). In other words, \( T(\alpha) \) indices the cliques sharing the same moment \( y_\alpha \). For \( \alpha \in \cup_{j \in [p]} \mathbb{N}_k^{l_j} \) such that \( |T(\alpha)| \geq 2 \), for every \( j \in T(\alpha) \), let \( \tilde{A}_j^{(\alpha)} \in S_{j,k} \) be such that \( \langle \tilde{A}_j^{(\alpha)}, X_j \rangle = y_\alpha \). It implies the constraints \( \langle \tilde{A}_j^{(\alpha)}, X_j \rangle = \langle \tilde{A}_j^{(\alpha)}, X_j \rangle = 0, \quad i \in T(\alpha) \setminus \{ j_0 \}, \)

for every \( \alpha \in \cup_{j \in [p]} \mathbb{N}_k^{l_j} \) such that \( |T(\alpha)| \geq 2 \), for some \( j_0 \in T(\alpha) \). We denote by \( \tilde{A}X = 0_{\mathbb{R}^\tilde{\mu}} \) all these constraints with \( X = \text{diag}(X_j) \).

Set \( \zeta := \sum_{j \in [p]} \| \tilde{\mu}_j + \tilde{\mu} \| \) and \( b = [(\tilde{b}_j)]_{j \in [p]}, 0_{\mathbb{R}^\tilde{\mu}} \in \mathbb{R}^\tilde{\mu} \). Define the linear operator \( \tilde{A} : \prod_{j \in [p]} S_{j,k} \rightarrow \mathbb{R}^{\tilde{\mu}} \) such that \( \tilde{A}X = [\langle \tilde{A}_jX_j \rangle]_{j \in [p]} \), for all \( X = \text{diag}(X_j)_{j \in [p]} \in \prod_{j \in [p]} S_j \). From (1.40), the affine constraints of SDP (1.37) are now equivalent to \( \tilde{A}X = b \).

Let \( A^{(i)} := \text{diag}((A_{i,j})_{j \in [p]}) \in \prod_{j \in [p]} S_j, i \in [\tilde{\mu}], \) be such that

\[
\tilde{A}X = [\langle A^{(i)}, X \rangle, \ldots, \langle A^{(\tilde{\mu})}, X \rangle],
\]

for all \( X = \text{diag}(X_j)_{j \in [p]} \in \prod_{j \in [p]} S_j \). For every \( j \in [p] \), define \( A_j : S_j \rightarrow \mathbb{R}^{\tilde{\mu}} \) as a linear operator of the form \( A_jX := [(A_1, X), \ldots, (A_{\tilde{\mu}}, X)] \). Then \( \tilde{A}X = \sum_{j \in [p]} \bar{A}_jX_j \), for all \( X = \text{diag}(X_j)_{j \in [p]} \in \prod_{j \in [p]} S_j \). Hence, we obtain the data \( (C_{j,k}, A_{j,k}, b_k, \zeta_k) = (C_j, A_j, b, \zeta) \) of the standard form (1.41) by plugging \( k \).

### REFERENCES

[1] MOSEK ApS. 2019. The MOSEK Optimization Toolbox. Version 9.1. Retrieved from https://www.mosek.com/documentation.

[2] Stephen Boyd, Neal Parikh, and Eric Chu. 2011. Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers. Now Publishers Inc.

ACM Transactions on Mathematical Software, Vol. 48, No. 4, Article 40. Publication date: December 2022.
[3] Samuel Burer and Renato D. C. Monteiro. 2005. Local minima and convergence in low-rank semidefinite programming. *Math. Program.* 103, 3 (2005), 427–444.

[4] Sabine Burgdorf, Igor Klep, and Janez Povh. 2016. *Optimization of Polynomials in Non-commuting Variables.* Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-33338-0

[5] Yair Carmon and John C Duchi. 2020. First-order methods for nonconvex quadratic minimization. *SIAM Rev.* 62, 2 (2020), 395–436.

[6] Tong Chen, Jean B. Lasserre, Victor Magron, and Edouard Pauwels. 2020. Semialgebraic optimization for Lipschitz constants of ReLU networks. *Adv. Neural Inf. Process. Syst.* 33 (2020).

[7] Chris Coey, Lea Kapelevich, and Juan Pablo Vielma. 2020. Towards practical generic conic optimization. *arXiv preprint arXiv:2005.01136* (2020).

[8] Lijun Ding and Benjamin Grimmer. 2020. Revisit of spectral bundle methods: Primal-dual (sub) linear convergence rates. *arXiv preprint arXiv:2008.07067* (2020).

[9] Yoshio Ebihara, Hayato Waki, Victor Magron, Ngoc Hoang Anh Mai, Dimitri Peaucelle, and Sophie Tarbouriech. 2021. L2 induced norm analysis of discrete-time LTI systems for nonnegative input signals and its application to stability analysis of recurrent neural networks. *Eur. J. Contr.* 62 (2021), 99–104.

[10] Michael Garstka, Mark Cannon, and Paul Goulart. 2021. COSMO: A conic operator splitting method for convex conic problems. *J. Optim. Theor. Appl.* 190, 3 (2021), 779–810.

[11] Hadrien Godard, Sourour Elloumi, Amélie Lambert, Jean Maeght, and Manuel Ruiz. 2019. Novel approach towards global optimality of optimal power flow using quadratic convex optimization. In *6th International Conference on Control, Decision and Information Technologies (CoDIT).* IEEE, 1227–1232.

[12] Marjo Haarala, Kaisa Miettinen, and Marko M. Mäkelä. 2004. New limited memory bundle method for large-scale nonsmooth optimization. *Optim. Meth. Softw.* 19, 6 (2004), 673–692.

[13] Napsu Haarala, Kaisa Miettinen, and Marko M. Mäkelä. 2007. Globally convergent limited memory bundle method for large-scale nonsmooth optimization. *Math. Program.* 109, 1 (2007), 181–205.

[14] Christoph Helmberg, Michael L. Overton, and Franz Rendl. 2014. The spectral bundle method with second-order information. *Optim. Meth. Softw.* 29, 4 (2014), 855–876.

[15] Christoph Helmberg and Franz Rendl. 2000. A spectral bundle method for semidefinite programming. *SIAM J. Optim.* 10, 3 (2000), 673–696.

[16] Christoph Helmberg, Franz Rendl, Robert J. Vanderbei, and Henry Wolkowicz. 1996. An interior-point method for semidefinite programming. *SIAM J. Optim.* 6, 2 (1996), 342–361.

[17] Didier Henrion, Milan Korda, and Jean Bernard Lasserre. 2020. *The Moment-SOS Hierarchy: Lectures in Probability, Statistics, Computational Geometry, Control and Nonlinear PDEs.* Vol. 4. World Scientific.

[18] Mingyi Hong and Zhi-Quan Luo. 2017. On the linear convergence of the alternating direction method of multipliers. *Math. Program.* 162, 1–2 (2017), 165–199.

[19] Cédric Josz, Stéphane Fliscounakis, Jean Maeght, and Patrick Panciatici. 2016. AC power flow data in MATPOWER and QCQP format: iTesla, RTE snapshots, and PEGASE. *arXiv preprint arXiv:1603.01533* (2016).

[20] Cédric Josz and Daniel K. Molzahn. 2018. Lasserre hierarchy for large scale polynomial optimization in real and complex variables. *SIAM J. Optim.* 28, 2 (2018), 1017–1048.

[21] Napsu Karmitsa. 2007. LMBM–FORTRAN subroutines for large-scale nonsmooth minimization: User’s manual. *TUCS Tech. Rep.* 77, 856 (2007).

[22] Krzysztof C. Kiwiel. 1991. A tilted cutting plane proximal bundle method for convex nondifferentiable optimization. *Oper. Res. Lett.* 10, 2 (1991), 75–81.

[23] Igor Klep, Victor Magron, and Janez Povh. 2021. Sparse noncommutative polynomial optimization. *Math. Program.* 193, 2 (2022), 789–829.

[24] Jean B. Lasserre. 2001. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.* 11, 3 (2001), 796–817.

[25] Jean B. Lasserre. 2006. Convergent SDP-relaxations in polynomial optimization with sparsity. *SIAM J. Optim.* 17, 3 (2006), 822–843.

[26] Jean-Bernard Lasserre. 2010. *Moments, Positive Polynomials and Their Applications.* Vol. 1. World Scientific.

[27] J. B. Lasserre. 2016. A MAX-CUT formulation of 0/1 programs. *Oper. Res. Letters* 44 (2016), 158–164.

[28] Jongwon Lee, Venkataramanan Balakrishnan, Cheng-Kok Koh, and Dan Jiao. 2009. From $O(k^2N)$ to $O(N)$: A fast complex-valued eigenvalue solver for large-scale on-chip interconnect analysis. In *IEEE MTT-S International Microwave Symposium Digest.* IEEE, 181–184.

[29] Richard B. Lehoucq, Danny C. Sorensen, and Chao Yang. 1998. *ARPACK Users’ Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* SIAM.

[30] N. H. A. Mai, J.-B. Lasserre, and V. Magron. 2020. A hierarchy of spectral relaxations for polynomial optimization. *arXiv preprint arXiv:2007.09027* (2020).
[31] M. Marshall. 2009. Representations of non-negative polynomials, degree bounds and applications to optimization. *Canad. J. Math.* 61, 1 (2009), 205–221.

[32] Brendan O’Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. 2016. Conic optimization via operator splitting and homogeneous self-dual embedding. *J. Optim. Theor. Appl.* 169, 3 (2016), 1042–1068.

[33] Dávid Papp and Sercan Yildiz. 2019. Sum-of-squares optimization without semidefinite programming. *SIAM J. Optim.* 29, 1 (2019), 822–851.

[34] Dávid Papp and Sercan Yildiz. 2021. Alfonso: Matlab package for nonsymmetric conic optimization. *Accepted in INFORMS Journal on Computing* 34, 1 (2022). 11–19.

[35] Kim-Chuan Toh, Michael J. Todd, and Reha H. Tütüncü. 1999. SDPT3—a MATLAB software package for semidefinite programming, version 1.3. *Optim. Meth. Softw.* 11, 1–4 (1999), 545–581.

[36] Lieven Vandenberghe, V. Ragu Balakrishnan, Ragnar Wallin, Anders Hansson, and Tae Roh. 2005. Interior-point algorithms for semidefinite programming problems derived from the KYP lemma. In *Positive Polynomials in Control*. Springer, 195–238.

[37] Lieven Vandenberghe and Stephen Boyd. 1996. Semidefinite programming. *SIAM Rev.* 38, 1 (1996), 49–95.

[38] Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. 2006. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM J. Optim.* 17, 1 (2006), 218–242.

[39] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. 2006. Sums of squares and semidefinite programming relaxations for polynomial optimization problems with structured sparsity. *SIAM J. Optim.* 17, 1 (2006), 218–242.

[40] Hayato Waki, Sunyoung Kim, Masakazu Kojima, Masakazu Muramatsu, and Hiroshi Sugimoto. 2008. Algorithm 883: SparsePOP—A sparse semidefinite programming relaxation of polynomial optimization problems. *ACM Trans. Math. Softw.* 35, 2 (2008), 15.

[41] Irene Waldspurger and Alden Waters. 2020. Rank optimality for the Burer-Monteiro factorization. *SIAM J. Optim.* 30, 3 (2020), 2577–2602.

[42] Jie Wang and Victor Magron. 2020. A second order cone characterization for sums of nonnegative circuits. In *45th International Symposium on Symbolic and Algebraic Computation*. 450–457.

[43] Jie Wang and Victor Magron. 2021. Exploiting term sparsity in noncommutative polynomial optimization. *Comput. Optim. Appl.* 80, 2 (2021), 483–521.

[44] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. 2021. Chordal-TSSOS: A moment-SOS hierarchy that exploits term sparsity with chordal extension. *SIAM J. Optim.* 31, 1 (2021), 114–141.

[45] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. 2021. TSSOS: A moment-SOS hierarchy that exploits term sparsity. *SIAM J. Optim.* 31, 1 (2021), 30–58.

[46] Jie Wang, Victor Magron, Jean B. Lasserre, and Ngoc Hoang Anh Mai. 2020. CS-TSSOS: Correlative and term sparsity for large-scale polynomial optimization. arXiv preprint arXiv:2005.02828 (2020).

[47] Tillmann Weisser, Benoit Legat, Chris Coey, Lea Kapelevich, and Juan Pablo Vielma. 2019. Polynomial and moment optimization in Julia and JuMP. In *JuliaCon*. Retrieved from https://pretalx.com/juliacon2019/talk/QZBKAU/.

[48] Alp Yurtsever, Olivier Fercoq, and Volkan Cevher. 2019. A conditional-gradient-based augmented Lagrangian framework. In *International Conference on Machine Learning*. PMLR, 7272–7281.

[49] Alp Yurtsever, Joel A. Tropp, Olivier Fercoq, Madeleine Udell, and Volkan Cevher. 2021. Scalable semidefinite programming. *SIAM J. Math. Data Sci.* 3, 1 (2021), 171–200.

Received 16 December 2020; revised 2 March 2022; accepted 21 July 2022