A MORE EFFICIENT REFORMULATION OF COMPLEX SDP AS REAL SDP[*]

JIE WANG[†]

Abstract. This note proposes a new reformulation of complex semidefinite programs (SDPs) as real SDPs. As an application, we present an economical reformulation of complex SDP relaxations of complex polynomial optimization problems as real SDPs and derive some further reductions by exploiting inner structure of the complex SDP relaxations. Various numerical examples demonstrate that our new reformulation runs significantly faster than the usual popular reformulation.

Key words. complex semidefinite programming, complex polynomial optimization, semidefinite programming, the complex moment-HSOS hierarchy, quantum information

MSC codes. 90C22, 90C23

1. Introduction.

Complex semidefinite programs (SDPs) arise from a diverse set of areas, such as combinatorial optimization [9], optimal power flow [10, 12], quantum information theory [2, 4, 17], signal processing [11, 14]. In particular, they appear as convex relaxations of complex polynomial optimization problems (CPOPs), giving rise to the complex moment-Hermitian-sum-of-squares (moment-HSOS) hierarchy [10, 15, 16]. However, most modern SDP solvers deal with only real SDPs[1]. In order to handle complex SDPs via real SDP solvers, it is mandatory to reformulate complex SDPs as equivalent real SDPs. A popular way[2] to do so is to use the equivalent condition

$$H \succeq 0 \quad \Longleftrightarrow \quad Y = \begin{bmatrix} H_R & -H_I \\ H_I & H_R \end{bmatrix} \succeq 0 \tag{1.1}$$

for an Hermitian matrix variable $H = H_R + H_I\mathbf{i} \in \mathbb{C}^{n\times n}$ with H_R and H_I being its real and imaginary parts respectively. Note that the right-hand-side constraint in (1.1) entails certain structure and to feed it to an SDP solver, we need to impose extra affine constraints to the positive semidefinite (PSD) constraint $Y \succeq 0$:

$$Y_{i,j} = Y_{i+n,j+n}, Y_{i,j+n} + Y_{j,i+n} = 0, \quad i = 1, \ldots, n, j = i, \ldots, n. \tag{1.2}$$

This conversion is quite simple but could be inefficient when n is large. In this note, inspired by Lagrange duality, we propose a new reformulation of complex SDPs as real SDPs. The benefit of this new reformulation is that there is no need to add extra affine constraints and hence it owns lower complexity. In the same manner, we can obtain a new reformulation of complex SDP relaxations of CPOPs as real SDPs. Moreover, by exploiting inner structure of the complex SDP relaxations, we are able to remove a bunch of redundant affine constraints, which leads to an even more economical real reformulation of the complex SDP relaxations. Various numerical experiments (including randomly generated CPOPs and the alternating current optimal power

[*]Submitted to the editors DATE.

Funding: This work was funded by NSFC-12201618 and NSFC-12171324.

[†]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China (wangjie212@amss.ac.cn, https://wangjie212.github.io/jiewang/)

[1]As far as the author knows, `SeDuMi` [13], `Sdolab` [7], and `Hypatia` [6] are the only solvers that can handle complex SDPs directly.

[2]See for instance the online modeling cookbook of the commercial SDP solver `MOSEK`: https://docs.mosek.com/modeling-cookbook/sdo.html.

flow (AC-OPF) problem) confirm our theoretical expectation and demonstrate that the new reformulation is more efficient than the usual popular one. Actually, our implementation of the new reformulation with `MOSEK` [1] also runs much faster than the implementation of the original complex formulation with `Hypatia` [8], probably because the SDP solvers based on real numbers are more mature and robust.

Notation. The symbol $\mathbb{N}$ denotes the set of nonnegative integers. For $n \in \mathbb{N} \setminus \{0\}$, let $[n] \coloneqq \{1, 2, \ldots, n\}$. We use $|A|$ to stand for the cardinality of a set A. Let $\mathbf{i}$ be the imaginary unit, satisfying $\mathbf{i}^2 = -1$. For $d \in \mathbb{N}$, let $\mathbb{N}^n_d \coloneqq \{(\alpha_i)_i \in \mathbb{N}^n \mid \sum_{i=1}^n \alpha_i \le d\}$ and let $\omega_{n,d} \coloneqq \binom{n+d}{d}$ be the cardinality of $\mathbb{N}^n_d$. For $\boldsymbol{\alpha} = (\alpha_i)_i \in \mathbb{N}^n_d$ and an n-tuple of variables $\mathbf{z} = \{z_1, \ldots, z_n\}$, let $\mathbf{z}^{\boldsymbol{\alpha}} \coloneqq z_1^{\alpha_1} \cdots z_n^{\alpha_n}$. For a complex number a, $\overline{a}$ (resp. $\mathcal{R}(a), \mathcal{I}(a)$) denotes the conjugate (resp. real part, imaginary part) of a, and for a complex vector v, v^{H} denotes the conjugate transpose of v. For a positive integer n, the set of $n \times n$ symmetric (resp. Hermitian) matrices is denoted by $\mathbf{S}^n$ (resp. H^n). We use $A \succeq 0$ to indicate that the matrix A is PSD. For $A, B \in \mathbb{C}^{n\times n}$, we denote by $\langle A, B\rangle$ the trace inner-product, defined by $\langle A, B\rangle = \mathrm{Tr}(A^{\mathrm{H}} B)$, where A^{H} stands for the conjugate transpose of A. For $A \in \mathbb{R}^{n\times n}$, $A^{\intercal}$ stands for the transpose of A.

We endow $\mathbb{C}^m$ (viewed as a $\mathbb{R}$-vector space) with the scalar product $\langle \cdot, \cdot \rangle_{\mathbb{R}}$ defined by

$$\langle u, v\rangle_{\mathbb{R}} = \mathcal{R}(u^{\mathrm{H}} v) = \mathcal{R}(u)^{\intercal}\mathcal{R}(v) + \mathcal{I}(u)^{\intercal}\mathcal{I}(v), \quad u, v \in \mathbb{C}^m.$$

For $u, v \in \mathbb{R}^m$, $\langle u, v\rangle \coloneqq u^{\intercal} v$, where $u^{\intercal}$ stands for the transpose of u. For each $A \in \mathbb{C}^{n\times n}$, we associate it with an Hermitian matrix $\mathcal{H}(A) \coloneqq \frac{1}{2}(A + A^{\mathrm{H}})$. One can check that $\mathcal{R}(\langle A, H\rangle) = \langle \mathcal{H}(A), H\rangle$ for any $H \in \mathrm{H}^n$.

2. The real reformulations of complex SDPs. Given a tuple of complex matrices $A_1, \ldots, A_m \in \mathbb{C}^{n\times n}$, we define a $\mathbb{R}$-linear operator $\mathscr{A} \colon \mathrm{H}^n \to \mathbb{C}^m$ by

$$\mathscr{A}(H) \coloneqq (\langle A_i, H\rangle)_{i=1}^m \in \mathbb{C}^m, \quad \forall H \in \mathrm{H}^n. \tag{2.1}$$

Let us consider the following complex SDP:

$$\text{(PSDP-}\mathbb{C}\text{)} \qquad \begin{cases} \sup\limits_{H\in \mathrm{H}^n} & \langle C, H\rangle \\ \text{s.t.} & \mathscr{A}(H) = b, \\ & H \succeq 0, \end{cases}$$

where $C \in \mathrm{H}^n, b \in \mathbb{C}^m$. In order to convert (PSDP-$\mathbb{C}$) to a real SDP, we define two real linear operators $\mathscr{A}_R, \mathscr{A}_I \colon \mathbb{R}^{n\times n} \to \mathbb{R}^m$ associated to $\mathscr{A}$ by

$$\mathscr{A}_R(S) \coloneqq (\langle \mathcal{R}(A_i), S\rangle)_{i=1}^m \in \mathbb{R}^m, \quad \forall S \in \mathbb{R}^{n\times n} \tag{2.2}$$

and

$$\mathscr{A}_I(S) \coloneqq (\langle \mathcal{I}(A_i), S\rangle)_{i=1}^m \in \mathbb{R}^m, \quad \forall S \in \mathbb{R}^{n\times n}, \tag{2.3}$$

respectively. Moreover, assume $H = H_R + H_I\mathbf{i}, C = C_R + C_I\mathbf{i}, b = b_R + b_I\mathbf{i}$ with

$H_R, H_I, C_R, C_I \in \mathbb{R}^{n\times n}$, $b_R, b_I \in \mathbb{R}^m$. We can now convert (PSDP-$\mathbb{C}$) to a real SDP:

$$\text{(PSDP-}\mathbb{R}\text{)} \qquad \begin{cases} \sup\limits_{Y\in\mathbf{S}^{2n}} & \langle C_R, H_R\rangle + \langle C_I, H_I\rangle \\ \text{s.t.} & \mathscr{A}_R(H_R) + \mathscr{A}_I(H_I) = b_R, \\ & \mathscr{A}_R(H_I) - \mathscr{A}_I(H_R) = b_I, \\ & Y = \begin{bmatrix} H_R & -H_I \\ H_I & H_R \end{bmatrix} \succeq 0. \end{cases}$$

As mentioned in the introduction, to feed the PSD constraint in (PSDP-$\mathbb{R}$) to an SDP solver, we need to include also the extra $n(n+1)$ affine constraints listed in (1.2), which could be inefficient in practice. Below we show that by taking a dual point of view, we can actually get rid of this issue.

Before formulating the dual problem of (PSDP-$\mathbb{C}$), we explicitly give the adjoint operator of $\mathscr{A}$.

LEMMA 2.1. *The adjoint operator $\mathscr{A}^*$ of $\mathscr{A}$ satisfies $\mathscr{A}^*(y) = \mathcal{H}\left(\sum_{i=1}^m y_i A_i\right)$ for $y \in \mathbb{C}^m$.*

Proof. For any $H \in \mathrm{H}^n$, we have

$$\begin{aligned} \langle \mathscr{A}^*(y), H\rangle = \langle y, \mathscr{A}(H)\rangle_{\mathbb{R}} &= \mathcal{R}\left(\sum_{i=1}^m \overline{y}_i \langle A_i, H\rangle\right) \\ &= \mathcal{R}\left(\left\langle \sum_{i=1}^m y_i A_i, H\right\rangle\right) = \left\langle \mathcal{H}\left(\sum_{i=1}^m y_i A_i\right), H\right\rangle, \end{aligned}$$

which yields $\mathscr{A}^*(y) = \mathcal{H}\left(\sum_{i=1}^m y_i A_i\right)$. □

Then by the convex duality theory, the dual problem of (PSDP-$\mathbb{C}$) reads as

$$\text{(DSDP-}\mathbb{C}\text{)} \qquad \begin{cases} \inf\limits_{y\in\mathbb{C}^m} & \langle b, y\rangle_{\mathbb{R}} \\ \text{s.t.} & \mathscr{A}^*(y) \succeq C. \end{cases}$$

Assume $y = y_R + y_I\mathbf{i}$ with $y_R, y_I \in \mathbb{R}^m$. Using Lemma 2.1, we deduce that $\mathscr{A}^*(y) = U + V\mathbf{i}$ with

$$U := \frac{1}{2}\sum_{i=1}^m \mathcal{R}(y_i)\mathcal{R}(A_i + A_i^\intercal) - \mathcal{I}(y_i)\mathcal{I}(A_i + A_i^\intercal)$$

and

$$V := \frac{1}{2}\sum_{i=1}^m \mathcal{I}(y_i)\mathcal{R}(A_i - A_i^\intercal) + \mathcal{R}(y_i)\mathcal{I}(A_i - A_i^\intercal).$$

Thus, we can convert (DSDP-$\mathbb{C}$) to a real SDP by using the equivalent condition (1.1):

$$\text{(DSDP-}\mathbb{R}\text{)} \qquad \begin{cases} \inf\limits_{y_R, y_I\in\mathbb{R}^m} & b_R^\intercal y_R + b_I^\intercal y_I \\ \text{s.t.} & \begin{bmatrix} U - C_R & -V + C_I \\ V - C_I & U - C_R \end{bmatrix} \succeq 0. \end{cases}$$

Let $X = \left[\begin{smallmatrix} X_1 & X_3^\intercal \\ X_3 & X_2 \end{smallmatrix}\right] \in \mathbf{S}^{2n}$ be the dual PSD variable of (DSDP-$\mathbb{R}$) with $X_1, X_2, X_3 \in \mathbb{R}^{n\times n}$. Then the Lagrangian associated with (DSDP-$\mathbb{R}$) given by

$$\begin{aligned}
&L(X, y_R, y_I)\\
&= b_R^\intercal y_R + b_I^\intercal y_I - \left\langle \begin{bmatrix} X_1 & X_3^\intercal \\ X_3 & X_2 \end{bmatrix}, \begin{bmatrix} U - C_R & -V + C_I \\ V - C_I & U - C_R \end{bmatrix} \right\rangle\\
&= b_R^\intercal y_R + b_I^\intercal y_I - \langle X_1 + X_2, U - C_R\rangle - \langle X_3 - X_3^\intercal, V - C_I\rangle\\
&= b_R^\intercal y_R + b_I^\intercal y_I + \langle C_R, X_1 + X_2\rangle + \langle C_I, X_3 - X_3^\intercal\rangle\\
&\quad - \sum_{i=1}^m \langle \mathcal{R}(A_i), X_1 + X_2\rangle \mathcal{R}(y_i) + \sum_{i=1}^m \langle \mathcal{I}(A_i), X_1 + X_2\rangle \mathcal{I}(y_i)\\
&\quad - \sum_{i=1}^m \langle \mathcal{R}(A_i), X_3 - X_3^\intercal\rangle \mathcal{I}(y_i) - \sum_{i=1}^m \langle \mathcal{I}(A_i), X_3 - X_3^\intercal\rangle \mathcal{R}(y_i)\\
&= \langle C_R, X_1 + X_2\rangle + \langle C_I, X_3 - X_3^\intercal\rangle + \langle b_R - \mathscr{A}_R(X_1 + X_2) - \mathscr{A}_I(X_3 - X_3^\intercal), y_R\rangle\\
&\quad + \langle b_I - \mathscr{A}_R(X_3 - X_3^\intercal) + \mathscr{A}_I(X_1 + X_2), y_I\rangle.
\end{aligned}$$

Therefore, the dual problem of (DSDP-$\mathbb{R}$) can be written down as

$$\text{(PSDP-}\mathbb{R}\text{')} \qquad \begin{cases} \sup\limits_{X\in\mathbf{S}^{2n}} & \langle C_R, X_1 + X_2\rangle + \langle C_I, X_3 - X_3^\intercal\rangle \\ \text{s.t.} & \mathscr{A}_R(X_1 + X_2) + \mathscr{A}_I(X_3 - X_3^\intercal) = b_R, \\ & \mathscr{A}_R(X_3 - X_3^\intercal) - \mathscr{A}_I(X_1 + X_2) = b_I, \\ & X = \begin{bmatrix} X_1 & X_3^\intercal \\ X_3 & X_2 \end{bmatrix} \succeq 0. \end{cases}$$

Theorem 2.2. (PSDP-$\mathbb{R}$') *is equivalent to* (PSDP-$\mathbb{R}$) *(in the sense that they share the same optimum). As a result,* (PSDP-$\mathbb{R}$') *is equivalent to* (PSDP-$\mathbb{C}$). *In addition, if* $X^\star = \left[\begin{smallmatrix} X_1^\star & (X_3^\star)^\intercal \\ X_3^\star & X_2^\star \end{smallmatrix}\right]$ *is an optimal solution to* (PSDP-$\mathbb{R}$')*, then* $H^\star = (X_1^\star + X_2^\star) + (X_3^\star - (X_3^\star)^\intercal)\mathbf{i}$ *is an optimal solution to* (PSDP-$\mathbb{C}$).

Proof. Let us denote the optima of (PSDP-$\mathbb{R}$) and (PSDP-$\mathbb{R}$') by v and v', respectively. Suppose that $Y = \left[\begin{smallmatrix} H_R & -H_I \\ H_I & H_R \end{smallmatrix}\right]$ is a feasible solution to (PSDP-$\mathbb{R}$). Then one can easily check that $X \coloneqq \frac{1}{2}Y$ is a feasible solution to (PSDP-$\mathbb{R}$'). Moreover, we have $\langle C_R, X_1 + X_2\rangle + \langle C_I, X_3 - X_3^\intercal\rangle = \langle C_R, H_R\rangle + \langle C_I, H_I\rangle$ and it follows $v \le v'$. On the other hand, suppose $X = \left[\begin{smallmatrix} X_1 & X_3^\intercal \\ X_3 & X_2 \end{smallmatrix}\right]$ is a feasible solution to (PSDP-$\mathbb{R}$'). We then have

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}^{-1} X \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} X_2 & -X_3 \\ -X_3^\intercal & X_1 \end{bmatrix} \succeq 0,$$

and thus

$$Y = \begin{bmatrix} H_R & -H_I \\ H_I & H_R \end{bmatrix} \coloneqq \begin{bmatrix} X_1 + X_2 & X_3^\intercal - X_3 \\ X_3 - X_3^\intercal & X_1 + X_2 \end{bmatrix} \succeq 0.$$

One can easily see that Y is a feasible solution to (PSDP-$\mathbb{R}$) and in addition, it holds $\langle C_R, H_R\rangle + \langle C_I, H_I\rangle = \langle C_R, X_1 + X_2\rangle + \langle C_I, X_3 - X_3^\intercal\rangle$. Thus $v \ge v'$, which proves the equivalence. The latter statement of the theorem is clear from the above arguments. ∎

In contrast to (PSDP-$\mathbb{R}$), the PSD constraint in (PSDP-$\mathbb{R}$') is straightforward, and thus no extra affine constraint is required. This is why the conversion (PSDP-$\mathbb{R}$') is more appealing than (PSDP-$\mathbb{R}$) from the computational perspective.

Remark 2.3. A similar reformulation to (PSDP-$\mathbb{R}$') but for a restricted class of complex SDP relaxations of multiple-input multiple-output detection has appeared in [11].

3. Application to complex SDP relaxations for CPOPs. In this section, we apply the reformulation (PSDP-$\mathbb{R}$') to complex SDP relaxations arising from the complex moment-HSOS hierarchy for CPOPs. A CPOP is given by

$$\text{(CPOP)} \qquad \begin{cases} \inf\limits_{\mathbf{z}\in\mathbb{C}^s} & f(\mathbf{z},\overline{\mathbf{z}}) = \sum_{\boldsymbol{\beta},\boldsymbol{\gamma}} b_{\boldsymbol{\beta},\boldsymbol{\gamma}}\mathbf{z}^{\boldsymbol{\beta}}\overline{\mathbf{z}}^{\boldsymbol{\gamma}} \\ \text{s.t.} & g_i(\mathbf{z},\overline{\mathbf{z}}) = \sum_{\boldsymbol{\beta},\boldsymbol{\gamma}} g^i_{\boldsymbol{\beta},\boldsymbol{\gamma}}\mathbf{z}^{\boldsymbol{\beta}}\overline{\mathbf{z}}^{\boldsymbol{\gamma}} \ge 0, \quad i\in[t], \end{cases}$$

where $\overline{\mathbf{z}} \coloneqq (\overline{z}_1,\dots,\overline{z}_s)$ stands for the conjugate of complex variables $\mathbf{z} \coloneqq (z_1,\dots,z_s)$. The functions $f, g_1, \dots, g_t$ are real-valued polynomials and their coefficients satisfy $b_{\boldsymbol{\beta},\boldsymbol{\gamma}} = \overline{b_{\boldsymbol{\gamma},\boldsymbol{\beta}}}$, $g^i_{\boldsymbol{\beta},\boldsymbol{\gamma}} = \overline{g^i_{\boldsymbol{\gamma},\boldsymbol{\beta}}}$. The *support* of f is defined by $\text{supp}(f) \coloneqq \{(\boldsymbol{\beta},\boldsymbol{\gamma}) \mid b_{\boldsymbol{\beta},\boldsymbol{\gamma}} \neq 0\}$. For $i\in[t]$, $\text{supp}(g_i)$ is defined in the same way.

Fix a $d\in\mathbb{N}$. Let $y = (y_{\boldsymbol{\beta},\boldsymbol{\gamma}})_{(\boldsymbol{\beta},\boldsymbol{\gamma})\in\mathbb{N}^s_d\times\mathbb{N}^s_d} \subseteq \mathbb{C}$ be a sequence indexed by $(\boldsymbol{\beta},\boldsymbol{\gamma}) \in \mathbb{N}^s_d\times\mathbb{N}^s_d$ and satisfying $y_{\boldsymbol{\beta},\boldsymbol{\gamma}} = \overline{y_{\boldsymbol{\gamma},\boldsymbol{\beta}}}$. Let L_y be the linear functional defined by

$$f = \sum_{(\boldsymbol{\beta},\boldsymbol{\gamma})} b_{\boldsymbol{\beta},\boldsymbol{\gamma}}\mathbf{z}^{\boldsymbol{\beta}}\overline{\mathbf{z}}^{\boldsymbol{\gamma}} \mapsto L_y(f) = \sum_{(\boldsymbol{\beta},\boldsymbol{\gamma})} b_{\boldsymbol{\beta},\boldsymbol{\gamma}} y_{\boldsymbol{\beta},\boldsymbol{\gamma}}.$$

The *complex moment* matrix $\mathbf{M}_d(y)$ associated with y is the Hermitian matrix indexed by $\mathbb{N}^s_d$ such that

$$[\mathbf{M}_d(y)]_{\boldsymbol{\beta},\boldsymbol{\gamma}} \coloneqq L_y(\mathbf{z}^{\boldsymbol{\beta}}\overline{\mathbf{z}}^{\boldsymbol{\gamma}}) = y_{\boldsymbol{\beta},\boldsymbol{\gamma}}, \quad \forall \boldsymbol{\beta},\boldsymbol{\gamma}\in\mathbb{N}^s_d.$$

Suppose that $g = \sum_{(\boldsymbol{\beta}',\boldsymbol{\gamma}')} g_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}\mathbf{z}^{\boldsymbol{\beta}'}\overline{\mathbf{z}}^{\boldsymbol{\gamma}'}$ is a complex polynomial. The *complex localizing* matrix $\mathbf{M}_d(gy)$ associated with g and y is the Hermitian matrix indexed by $\mathbb{N}^s_d$ such that

$$[\mathbf{M}_d(gy)]_{\boldsymbol{\beta},\boldsymbol{\gamma}} \coloneqq L_y(g\mathbf{z}^{\boldsymbol{\beta}}\overline{\mathbf{z}}^{\boldsymbol{\gamma}}) = \sum_{(\boldsymbol{\beta}',\boldsymbol{\gamma}')} g_{\boldsymbol{\beta}',\boldsymbol{\gamma}'} y_{\boldsymbol{\beta}+\boldsymbol{\beta}',\boldsymbol{\gamma}+\boldsymbol{\gamma}'}, \quad \forall \boldsymbol{\beta},\boldsymbol{\gamma}\in\mathbb{N}^s_d.$$

Let $d_0 \coloneqq \max\{|\boldsymbol{\beta}|,|\boldsymbol{\gamma}| : b_{\boldsymbol{\beta},\boldsymbol{\gamma}} \neq 0\}$ and $d_i \coloneqq \max\{|\boldsymbol{\beta}|,|\boldsymbol{\gamma}| : g^i_{\boldsymbol{\beta},\boldsymbol{\gamma}} \neq 0\}$ for $i\in[t]$, where $|\cdot|$ denotes the sum of entries. Let $d_{\min} \coloneqq \max\{d_0,d_1,\dots,d_m\}$. For any $d \ge d_{\min}$, the d-th (d is called the *relaxation order*) complex moment relaxation for (CPOP) is given by

$$\text{(Mom-}\mathbb{C}\text{)} \qquad \begin{cases} \inf\limits_{y} & L_y(f) = \langle b, y\rangle_{\mathbb{R}} \\ \text{s.t.} & \mathbf{M}_d(y) \succeq 0, \\ & \mathbf{M}_{d-d_i}(g_i y) \succeq 0, \quad i\in[t], \\ & y_{\mathbf{0},\mathbf{0}} = 1. \end{cases}$$

(Mom-$\mathbb{C}$) and its dual problem form the complex moment-HSOS hierarchy of (CPOP). For more details on this hierarchy, we refer the reader to [10, 15].

For any $(\boldsymbol{\beta},\boldsymbol{\gamma})\in\mathbb{N}^s_d\times\mathbb{N}^s_d$, we associate it with a matrix $A^0_{\boldsymbol{\beta},\boldsymbol{\gamma}}\in\mathbb{R}^{\omega_{s,d}\times\omega_{s,d}}$ defined by

$$[A^0_{\boldsymbol{\beta},\boldsymbol{\gamma}}]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}=\begin{cases}1, & \text{if } (\boldsymbol{\beta}',\boldsymbol{\gamma}')=(\boldsymbol{\beta},\boldsymbol{\gamma}),\\ 0, & \text{otherwise.}\end{cases}\tag{3.1}$$

Moreover, for each $i\in[t]$, we associate any $(\boldsymbol{\beta},\boldsymbol{\gamma})\in\mathbb{N}^s_{d-d_i}\times\mathbb{N}^s_{d-d_i}$ with a matrix $A^i_{\boldsymbol{\beta},\boldsymbol{\gamma}}\in\mathbb{C}^{\omega_{s,d-d_i}\times\omega_{s,d-d_i}}$ defined by

$$[A^i_{\boldsymbol{\beta},\boldsymbol{\gamma}}]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}=\begin{cases}g^i_{\boldsymbol{\beta}'',\boldsymbol{\gamma}''}, & \text{if } (\boldsymbol{\beta}'+\boldsymbol{\beta}'',\boldsymbol{\gamma}'+\boldsymbol{\gamma}'')=(\boldsymbol{\beta},\boldsymbol{\gamma}),\\ 0, & \text{otherwise.}\end{cases}\tag{3.2}$$

Now for each $i=0,1,\ldots,t$, we define the $\mathbb{R}$-linear operator $\mathscr{A}^i$ by

$$\mathscr{A}^i(H):=\left(\langle A^i_{\boldsymbol{\beta},\boldsymbol{\gamma}},H\rangle\right)_{(\boldsymbol{\beta},\boldsymbol{\gamma})\in\mathbb{N}^s_{d-d_i}\times\mathbb{N}^s_{d-d_i}},\quad H\in\mathrm{H}^{\omega_{s,d-d_i}}.$$

For convenience let us set $g_0:=1$. By construction, it holds

$$\mathbf{M}_{d-d_i}(g_iy)=\sum_{(\boldsymbol{\beta},\boldsymbol{\gamma})\in\mathbb{N}^s_{d-d_i}\times\mathbb{N}^s_{d-d_i}}A^i_{\boldsymbol{\beta},\boldsymbol{\gamma}}y_{\boldsymbol{\beta},\boldsymbol{\gamma}}=(\mathscr{A}^i)^*(y),\quad i=0,1,\ldots,t.$$

Therefore, one can rewrite (Mom-$\mathbb{C}$) as follows:

$$\text{(Mom-}\mathbb{C}\text{')}\qquad\begin{cases}\inf\limits_{y} & \langle b,y\rangle_{\mathbb{R}}\\ \text{s.t.} & (\mathscr{A}^i)^*(y)\succeq 0,\quad i=0,1,\ldots,t,\\ & y_{\mathbf{0},\mathbf{0}}=1,\end{cases}$$

whose dual reads as

$$\text{(HSOS-}\mathbb{C}\text{)}\qquad\begin{cases}\sup\limits_{\lambda,H^i} & \lambda\\ \text{s.t.} & \sum_{i=0}^t[\mathscr{A}^i(H^i)]_{\boldsymbol{\beta},\boldsymbol{\gamma}}+\delta_{(\boldsymbol{\beta},\boldsymbol{\gamma}),(\mathbf{0},\mathbf{0})}\lambda=b_{\boldsymbol{\beta},\boldsymbol{\gamma}},\quad(\boldsymbol{\beta},\boldsymbol{\gamma})\in\mathbb{N}^s_d\times\mathbb{N}^s_d,\\ & H^i\succeq 0,\quad i=0,1,\ldots,t.\end{cases}$$

Note that we have used the Kronecker delta $\delta_{(\boldsymbol{\beta},\boldsymbol{\gamma}),(\mathbf{0},\mathbf{0})}$ in (HSOS-$\mathbb{C}$).

Let us from now on fix any order "$<$" on $\mathbb{N}^s$.

Proposition 3.1. (HSOS-$\mathbb{C}$) *is equivalent to the following complex SDP:*

$$\text{(HSOS-}\mathbb{C}\text{')}\qquad\begin{cases}\sup\limits_{\lambda,H^i} & \lambda\\ \text{s.t.} & \sum_{i=0}^t[\mathscr{A}^i(H^i)]_{\boldsymbol{\beta},\boldsymbol{\gamma}}+\delta_{(\boldsymbol{\beta},\boldsymbol{\gamma}),(\mathbf{0},\mathbf{0})}\lambda=b_{\boldsymbol{\beta},\boldsymbol{\gamma}},\\ & \boldsymbol{\beta}\le\boldsymbol{\gamma},\quad(\boldsymbol{\beta},\boldsymbol{\gamma})\in\mathbb{N}^s_d\times\mathbb{N}^s_d,\\ & H^i\succeq 0,\quad i=0,1,\ldots,t.\end{cases}$$

Proof. It suffices to show that for $\boldsymbol{\beta}<\boldsymbol{\gamma}$, $\sum_{i=0}^t[\mathscr{A}^i(H^i)]_{\boldsymbol{\gamma},\boldsymbol{\beta}}=b_{\boldsymbol{\gamma},\boldsymbol{\beta}}$ is equivalent to $\sum_{i=0}^{t}[\mathscr{A}^i(H^i)]_{\boldsymbol{\beta},\boldsymbol{\gamma}} = b_{\boldsymbol{\beta},\boldsymbol{\gamma}}$. Indeed, this equivalence follows from $b_{\boldsymbol{\gamma},\boldsymbol{\beta}} = \overline{b_{\boldsymbol{\beta},\boldsymbol{\gamma}}}$ and

$$\begin{aligned}\sum_{i=0}^{t}[\mathscr{A}^i(H^i)]_{\boldsymbol{\gamma},\boldsymbol{\beta}} &= \sum_{i=0}^{t}\langle A^i_{\boldsymbol{\gamma},\boldsymbol{\beta}}, H^i\rangle \\ &= [H^0]_{\boldsymbol{\gamma},\boldsymbol{\beta}} + \sum_{i=1}^{t}\sum_{\substack{(\boldsymbol{\gamma}',\boldsymbol{\beta}')\in\mathbb{N}^s_{d-d_i}\times\mathbb{N}^s_{d-d_i}\\ (\boldsymbol{\gamma}'',\boldsymbol{\beta}'')\in\text{supp}(g)\\ (\boldsymbol{\gamma}'+\boldsymbol{\gamma}'',\boldsymbol{\beta}'+\boldsymbol{\beta}'')=(\boldsymbol{\gamma},\boldsymbol{\beta})}} g^i_{\boldsymbol{\gamma}'',\boldsymbol{\beta}''}[H^i]_{\boldsymbol{\gamma}',\boldsymbol{\beta}'} \\ &= \sum_{i=0}^{t}\langle\overline{A^i_{\boldsymbol{\beta},\boldsymbol{\gamma}}}, H^i\rangle = \sum_{i=0}^{t}\overline{[\mathscr{A}^i(H^i)]_{\boldsymbol{\beta},\boldsymbol{\gamma}}}.\end{aligned}$$ ∎

Let $\mathscr{A}^i_R, \mathscr{A}^i_I$ be the real linear operator associated with $\mathscr{A}^i$ which are defined in a similar way as (2.2) and (2.3). With $H^i = H^i_R + H^i_I\mathbf{i}, b = b_R + b_I\mathbf{i}$, (HSOS-$\mathbb{C}$') is equivalent to the following real SDP by using the equivalent condition (1.1):

(HSOS-$\mathbb{R}$)

$$\begin{cases}\sup\limits_{\lambda,Y^i} & \lambda\\ \text{s.t.} & \sum_{i=0}^{t}\left([\mathscr{A}^i_R(H^i_R)]_{\boldsymbol{\beta},\boldsymbol{\gamma}} + [\mathscr{A}^i_I(H^i_I)]_{\boldsymbol{\beta},\boldsymbol{\gamma}}\right) + \delta_{(\boldsymbol{\beta},\boldsymbol{\gamma}),(\mathbf{0},\mathbf{0})}\lambda = [b_R]_{\boldsymbol{\beta},\boldsymbol{\gamma}},\\ & \sum_{i=0}^{t}\left([\mathscr{A}^i_R(H^i_I)]_{\boldsymbol{\beta},\boldsymbol{\gamma}} - [\mathscr{A}^i_I(H^i_R)]_{\boldsymbol{\beta},\boldsymbol{\gamma}}\right) = [b_I]_{\boldsymbol{\beta},\boldsymbol{\gamma}},\\ & \boldsymbol{\beta}\le\boldsymbol{\gamma}, \quad (\boldsymbol{\beta},\boldsymbol{\gamma})\in\mathbb{N}^s_d\times\mathbb{N}^s_d,\\ & Y^i = \begin{bmatrix} H^i_R & -H^i_I\\ H^i_I & H^i_R\end{bmatrix}\succeq 0, \quad i = 0,1,\dots,t.\end{cases}$$

On the other hand, by invoking Theorem 2.2, we obtain another equivalent real SDP conversion of (HSOS-$\mathbb{C}$'):

(3.3)

$$\begin{cases}\sup\limits_{\lambda,X^i} & \lambda\\ \text{s.t.} & \sum_{i=0}^{t}\left([\mathscr{A}^i_R(X^i_1+X^i_2)]_{\boldsymbol{\beta},\boldsymbol{\gamma}} + [\mathscr{A}^i_I(X^i_3-(X^i_3)^\intercal)]_{\boldsymbol{\beta},\boldsymbol{\gamma}}\right) + \delta_{(\boldsymbol{\beta},\boldsymbol{\gamma}),(\mathbf{0},\mathbf{0})}\lambda = [b_R]_{\boldsymbol{\beta},\boldsymbol{\gamma}},\\ & \sum_{i=0}^{t}\left([\mathscr{A}^i_R(X^i_3-(X^i_3)^\intercal)]_{\boldsymbol{\beta},\boldsymbol{\gamma}} - [\mathscr{A}^i_I(X^i_1+X^i_2)]_{\boldsymbol{\beta},\boldsymbol{\gamma}}\right) = [b_I]_{\boldsymbol{\beta},\boldsymbol{\gamma}},\\ & \boldsymbol{\beta}\le\boldsymbol{\gamma}, \quad (\boldsymbol{\beta},\boldsymbol{\gamma})\in\mathbb{N}^s_d\times\mathbb{N}^s_d,\\ & X^i = \begin{bmatrix} X^i_1 & (X^i_3)^\intercal\\ X^i_3 & X^i_2\end{bmatrix}\succeq 0, \quad i = 0,1,\dots,t.\end{cases}$$

PROPOSITION 3.2. (3.3) *is equivalent to the following real SDP:*

(HSOS-$\mathbb{R}$')

$$\begin{cases}\sup\limits_{\lambda,X^i} & \lambda\\ \text{s.t.} & \sum_{i=0}^{t}\left([\mathscr{A}^i_R(X^i_1+X^i_2)]_{\boldsymbol{\beta},\boldsymbol{\gamma}} + [\mathscr{A}^i_I(X^i_3-(X^i_3)^\intercal)]_{\boldsymbol{\beta},\boldsymbol{\gamma}}\right) + \delta_{(\boldsymbol{\beta},\boldsymbol{\gamma}),(\mathbf{0},\mathbf{0})}\lambda = [b_R]_{\boldsymbol{\beta},\boldsymbol{\gamma}},\\ & \sum_{i=0}^{t}\left([\mathscr{A}^i_R(X^i_3-(X^i_3)^\intercal)]_{\boldsymbol{\beta},\boldsymbol{\gamma}} - [\mathscr{A}^i_I(X^i_1+X^i_2)]_{\boldsymbol{\beta},\boldsymbol{\gamma}}\right) = [b_I]_{\boldsymbol{\beta},\boldsymbol{\gamma}}, \quad \boldsymbol{\beta}\ne\boldsymbol{\gamma},\\ & \boldsymbol{\beta}\le\boldsymbol{\gamma}, \quad (\boldsymbol{\beta},\boldsymbol{\gamma})\in\mathbb{N}^s_d\times\mathbb{N}^s_d,\\ & X^i = \begin{bmatrix} X^i_1 & (X^i_3)^\intercal\\ X^i_3 & X^i_2\end{bmatrix}\succeq 0, \quad i = 0,1,\dots,t.\end{cases}$$

Proof. We need to show that the following constraints

$$\sum_{i=0}^{t}\left([\mathscr{A}^i_R(X^i_3-(X^i_3)^\intercal)]_{\boldsymbol{\beta},\boldsymbol{\beta}} - [\mathscr{A}^i_I(X^i_1+X^i_2)]_{\boldsymbol{\beta},\boldsymbol{\beta}}\right) = [b_I]_{\boldsymbol{\beta},\boldsymbol{\beta}} = 0, \quad \boldsymbol{\beta}\in\mathbb{N}^s_d \tag{3.4}$$

in (3.3) are redundant. For each $i=0,1,\ldots,t$ and $\boldsymbol{\beta}\in\mathbb{N}^s_d$, we have

$$\begin{aligned}
&\left\langle (A^i_{\boldsymbol{\beta},\boldsymbol{\beta}})_R, X^i_3-(X^i_3)^\intercal\right\rangle\\
=&\sum_{\substack{(\boldsymbol{\beta}',\boldsymbol{\gamma}')\in\mathbb{N}^s_{d-d_i}\times\mathbb{N}^s_{d-d_i}\\(\boldsymbol{\beta}'',\boldsymbol{\gamma}'')\in\operatorname{supp}(g)\\(\boldsymbol{\beta}'+\boldsymbol{\beta}'',\boldsymbol{\gamma}'+\boldsymbol{\gamma}'')=(\boldsymbol{\beta},\boldsymbol{\beta})}}\mathcal{R}(g^i_{\boldsymbol{\beta}'',\boldsymbol{\gamma}''})\left([X^i_3]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}-[(X^i_3)^\intercal]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}\right)\\
=&\frac{1}{2}\sum_{\substack{(\boldsymbol{\beta}',\boldsymbol{\gamma}')\in\mathbb{N}^s_{d-d_i}\times\mathbb{N}^s_{d-d_i}\\(\boldsymbol{\beta}'',\boldsymbol{\gamma}'')\in\operatorname{supp}(g)\\(\boldsymbol{\beta}'+\boldsymbol{\beta}'',\boldsymbol{\gamma}'+\boldsymbol{\gamma}'')=(\boldsymbol{\beta},\boldsymbol{\beta})}}\mathcal{R}(g^i_{\boldsymbol{\beta}'',\boldsymbol{\gamma}''})\left([X^i_3]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}+[X^i_3]_{\boldsymbol{\gamma}',\boldsymbol{\beta}'}-[(X^i_3)^\intercal]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}-[(X^i_3)^\intercal]_{\boldsymbol{\gamma}',\boldsymbol{\beta}'}\right)\\
=&\,0,
\end{aligned}$$

where we have used fact that $[(X^i_3)^\intercal]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}=[X^i_3]_{\boldsymbol{\gamma}',\boldsymbol{\beta}'}$ and $[(X^i_3)^\intercal]_{\boldsymbol{\gamma}',\boldsymbol{\beta}'}=[X^i_3]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}$. It follows that $[\mathscr{A}^i_R(X^i_3-(X^i_3)^\intercal)]_{\boldsymbol{\beta},\boldsymbol{\beta}}=\langle (A^i_{\boldsymbol{\beta},\boldsymbol{\beta}})_R, X^i_3-(X^i_3)^\intercal\rangle=0$.

In addition, for each $i=0,1,\ldots,t$ and $\boldsymbol{\beta}\in\mathbb{N}^s_d$, we have

$$\begin{aligned}
&\left\langle (A^i_{\boldsymbol{\beta},\boldsymbol{\beta}})_I, X^i_1+X^i_2\right\rangle\\
=&\sum_{\substack{(\boldsymbol{\beta}',\boldsymbol{\gamma}')\in\mathbb{N}^s_{d-d_i}\times\mathbb{N}^s_{d-d_i}\\(\boldsymbol{\beta}'',\boldsymbol{\gamma}'')\in\operatorname{supp}(g)\\(\boldsymbol{\beta}'+\boldsymbol{\beta}'',\boldsymbol{\gamma}'+\boldsymbol{\gamma}'')=(\boldsymbol{\beta},\boldsymbol{\beta})}}\mathcal{I}(g^i_{\boldsymbol{\beta}'',\boldsymbol{\gamma}''})\left([X^i_1]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}+[X^i_2]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}\right)\\
=&\frac{1}{2}\sum_{\substack{(\boldsymbol{\beta}',\boldsymbol{\gamma}')\in\mathbb{N}^s_{d-d_i}\times\mathbb{N}^s_{d-d_i}\\(\boldsymbol{\beta}'',\boldsymbol{\gamma}'')\in\operatorname{supp}(g)\\(\boldsymbol{\beta}'+\boldsymbol{\beta}'',\boldsymbol{\gamma}'+\boldsymbol{\gamma}'')=(\boldsymbol{\beta},\boldsymbol{\beta})}}\mathcal{I}\left(g^i_{\boldsymbol{\beta}'',\boldsymbol{\gamma}''}+g^i_{\boldsymbol{\gamma}'',\boldsymbol{\beta}''}\right)\left([X^i_1]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}+[X^i_2]_{\boldsymbol{\beta}',\boldsymbol{\gamma}'}\right)\\
=&\,0,
\end{aligned}$$

where we have used fact that $\mathcal{I}(g^i_{\boldsymbol{\beta}'',\boldsymbol{\gamma}''}+g^i_{\boldsymbol{\gamma}'',\boldsymbol{\beta}''})=\mathcal{I}(g^i_{\boldsymbol{\beta}'',\boldsymbol{\gamma}''}+\overline{g}^i_{\boldsymbol{\beta}'',\boldsymbol{\gamma}''})=0$ and X^i_1,X^i_2 are symmetric. It follows that $[\mathscr{A}^i_I(X^i_1+X^i_2)]_{\boldsymbol{\beta},\boldsymbol{\beta}}=\langle (A^i_{\boldsymbol{\beta},\boldsymbol{\beta}})_I, X^i_1+X^i_2\rangle=0$.

Putting all above together yields (3.4). ∎

We have proved the following theorem.

THEOREM 3.3. (HSOS-ℝ') *is equivalent to* (HSOS-ℂ).

Before closing the section, we compare complexity of different real SDP reformulations for complex SDP relaxations of (CPOP) in Table 1.

TABLE 1
Comparison of complexity of different real SDP reformulations for complex SDP relaxations of (CPOP). $n_{\rm sdp}$*: the maximal size of SDP matrix,* $m_{\rm sdp}$*: the number of affine constraints.*

	(HSOS-ℝ)	(HSOS-ℝ')
$n_{\rm sdp}$	$2\omega_{s,d}$	$2\omega_{s,d}$
$m_{\rm sdp}$	$2\omega_{s,d}^2+2\omega_{s,d}+\sum_{i=1}^t\omega_{s,d-d_i}$	$\omega_{s,d}^2$

4. Numerical experiments. In this section, we benchmark the performance of the two real reformulations for complex SDPs using the software `TSSOS` 1.2.1[3] in which

[3]`TSSOS` is freely available at https://github.com/wangjie212/TSSOS.

MOSEK 10.0 [1] is employed as an SDP solver with default settings. For comparison, we also include the results of directly solving complex SDPs obtained with Hypatia 0.8.1 [6] in Sections 4.1–4.3. All numerical experiments were performed on a desktop computer with Intel(R) Core(TM) i9-10900 CPU@2.80GHz and 64G RAM. When presenting the results, the column labelled by 'opt' records the optimum and the column labelled by 'time' records running time in seconds. Moreover, the symbol '-' means the SDP solver runs out of memory, and the symbol '$*$' means running time exceeds 10000 seconds.

4.1. Minimizing a random complex quartic polynomial over the unit sphere. Our first example is to minimize a complex quartic polynomial over the unit sphere:

$$\begin{cases} \inf_{\mathbf{z}\in\mathbb{C}^s} & [\mathbf{z}]_2^{\mathrm{H}} Q[\mathbf{z}]_2 \\ \text{s.t.} & |z_1|^2 + \cdots + |z_s|^2 = 1, \end{cases} \tag{4.1}$$

where $[\mathbf{z}]_2$ is the column vector of monomials in $\mathbf{z}$ up to degree two and $Q \in \mathrm{H}^{|[\mathbf{z}]_2|}$ is a random Hermitian matrix whose entries are selected with respect to the standard normal distribution.

We approach (4.1) for $s = 5, 7, \ldots, 15$ with the second and third HSOS relaxations. The related results are shown in Table 2. From the table, we see that the reformulation (HSOS-$\mathbb{R}$') is several ($2 \sim 7$) times as fast as the reformulation (HSOS-$\mathbb{R}$), and the speedup becomes more significant as the SDP size grows. On the other hand, solving the original complex SDP with Hypatia is much slower than solving the real SDP reformulation (HSOS-$\mathbb{R}$') with MOSEK.

TABLE 2
Minimizing a random complex quartic polynomial over the unit sphere.

s	d	(HSOS-$\mathbb{R}$)			(HSOS-$\mathbb{R}$')			(HSOS-$\mathbb{C}$')	
		m_{sdp}	opt	time	m_{sdp}	opt	time	opt	time
5	2	966	-11.2409	0.11	441	-11.2409	0.05	-11.2409	0.12
	3	6846	-9.47725	8.13	3136	-9.47725	2.00	-9.47725	6.54
7	2	2736	-14.2314	0.97	1296	-14.2314	0.28	-14.2314	0.59
	3	30372	-11.0407	389	14400	-11.0407	57.0	-11.0407	474
9	2	6270	-19.0019	5.73	3025	-19.0019	1.62	-19.0019	4.61
	3	100320	-	-	48400	-15.5614	1944	-	-
11	2	12480	-22.8630	31.7	6084	-22.8630	6.67	-22.8630	32.3
	3	271882	-	-	132496	-	-	-	-
13	2	22470	-25.6352	145	11025	-25.6352	23.5	-25.6352	174
	3	639450	-	-	313600	-	-	-	-
15	2	37536	-29.1672	585	18496	-29.1672	86.1	-29.1672	802
	3	1351976	-	-	665856	-	-	-	-

4.2. Minimizing a random complex quartic polynomial with unit-norm variables. The second example is to minimize a random complex quartic polynomial with unit-norm variables:

$$\begin{cases} \inf_{\mathbf{z}\in\mathbb{C}^s} & [\mathbf{z}]_2^{\mathrm{H}} Q[\mathbf{z}]_2 \\ \text{s.t.} & |z_i|^2 = 1, \quad i = 1, \ldots, s, \end{cases} \tag{4.2}$$

where $Q \in \mathrm{H}^{|[\mathbf{z}]_2|}$ is a random Hermitian matrix whose entries are selected with respect to the uniform probability distribution on $[0, 1]$.

We approach (4.2) for $s = 5, 7, \ldots, 15$ with the second and third HSOS relaxations. The related results are shown in Table 3. From the table, we see that the reformulation (HSOS-ℝ') is about one magnitude faster than the reformulation (HSOS-ℝ), and the speedup becomes more significant as the SDP size grows. Again, solving the original complex SDP with `Hypatia` is much slower than solving the real SDP reformulation (HSOS-ℝ') with `MOSEK`.

TABLE 3
Minimizing a random complex quartic polynomial with unit-norm variables.

s	d	(HSOS-ℝ)			(HSOS-ℝ')			(HSOS-ℂ')	
		m_{sdp}	opt	time	m_{sdp}	opt	time	opt	time
5	2	734	-24.4919	0.10	271	-24.4919	0.03	-24.4919	0.21
	3	4474	-24.4919	2.34	1281	-24.4919	0.26	-24.4919	10.9
7	2	2202	-56.5289	0.65	869	-56.5289	0.16	-56.5289	1.15
	3	21158	-46.7128	132	6637	-46.7128	7.44	-46.7128	520
9	2	5242	-114.342	4.62	2161	-114.342	0.73	-114.342	5.29
	3	73312	-	-	24691	-81.2676	184	-	-
11	2	10718	-202.436	32.1	4555	-202.436	3.86	-202.436	30.0
	3	206188	-	-	73327	-	-	-	-
13	2	19686	-338.041	126	8555	-338.041	12.7	-338.041	162
	3	499438	-	-	185277	-	-	-	-
15	2	33394	-514.226	678	14761	-514.226	55.1	-514.226	705
	3	1081514	-	-	414841	-	-	-	-

4.3. Minimizing a randomly generated sparse complex quartic polynomial over multi-spheres. Given $l \in \mathbb{N}\backslash\{0\}$, we randomly generate a sparse complex quartic polynomial as follows: Let $f = \sum_{i=1}^{l} f_i \in \mathbb{C}[z_1, \ldots, z_{5(l+1)}, \overline{z}_1, \ldots, \overline{z}_{5(l+1)}]$,[4] where for all $i \in [l]$, $f_i = \overline{f}_i \in \mathbb{C}[z_{5(i-1)+1}, \ldots, z_{5(i-1)+10}, \overline{z}_{5(i-1)+1}, \ldots, \overline{z}_{5(i-1)+10}]$ is a sparse complex quartic polynomial whose coefficients (real/imaginary parts) are selected with respect to the uniform probability distribution on $[-1, 1]$. Then we consider the following CPOP:

$$\begin{cases} \inf_{\mathbf{z}\in\mathbb{C}^{5(l+1)}} & f(\mathbf{z}, \overline{\mathbf{z}}) \\ \text{s.t.} & \sum_{j=1}^{10} |z_{5(i-1)+j}|^2 = 1, \quad i = 1, \ldots, l. \end{cases} \tag{4.3}$$

The sparsity in (4.3) can be exploited to derive a sparsity-adapted complex moment-HSOS hierarchy [15]. We solve the second sparse HSOS relaxation of (4.3) for

[4] $\mathbb{C}[\mathbf{z}, \overline{\mathbf{z}}]$ denotes the ring of complex polynomials in variables $\mathbf{z}, \overline{\mathbf{z}}$.

$l = 40, 80, \ldots, 400$. The results are displayed in Table 4. From the table we see that the reformulation (HSOS-$\mathbb{R}$') is $1.5 \sim 2$ times as fast as the reformulation (HSOS-$\mathbb{R}$). Moreover, for this problem, solving the original complex SDP with `Hypatia` is extremely slow probably due to the fact that the SDP contains many PSD blocks.

TABLE 4
Minimizing a randomly generated sparse complex quartic polynomial over multi-spheres.

l	(HSOS-$\mathbb{R}$)			(HSOS-$\mathbb{R}$')			(HSOS-$\mathbb{C}$')	
	m_{sdp}	opt	time	m_{sdp}	opt	time	opt	time
40	23090	-98.9240	3.12	12529	-98.9240	2.06	-98.9240	886
80	46768	-197.577	12.6	25549	-197.577	8.07	-197.577	5433
120	70958	-292.024	30.1	38871	-292.024	19.0	$*$	$*$
160	94278	-389.652	45.9	51563	-389.652	30.7	$*$	$*$
200	117526	-482.684	84.5	64185	-482.684	37.7	$*$	$*$
240	140298	-578.896	130	76389	-578.896	59.5	$*$	$*$
280	162504	-671.047	173	89241	-671.047	65.4	$*$	$*$
320	187528	-766.403	206	102171	-766.403	88.5	$*$	$*$
360	210370	-866.771	291	114589	-866.771	147	$*$	$*$
400	233396	-963.137	297	127173	-963.137	138	$*$	$*$

4.4. Application to the AC-OPF problem. The AC-OPF is a central problem in power systems, which aims to minimize the generation cost of an alternating current transmission network under physical and operational constraints. Mathematically, it can be formulated as the following CPOP:

$$
\tag{4.4}
\begin{cases}
\inf\limits_{V_i, S_k^g} & \sum_{k\in G} \left(\mathbf{c}_{2k}(\mathcal{R}(S_k^g))^2 + \mathbf{c}_{1k}\mathcal{R}(S_k^g) + \mathbf{c}_{0k}\right) \\
\text{s.t.} & \angle V_r = 0, \\
& \mathbf{S}_k^{gl} \le S_k^g \le \mathbf{S}_k^{gu}, \quad \forall k \in G, \\
& \boldsymbol{v}_i^l \le |V_i| \le \boldsymbol{v}_i^u, \quad \forall i \in N, \\
& \sum_{k\in G_i} S_k^g - \mathbf{S}_i^d - \mathbf{Y}_i^{sh}|V_i|^2 = \sum_{(i,j)\in E_i\cup E_i^R} S_{ij}, \quad \forall i \in N, \\
& S_{ij} = (\overline{\mathbf{Y}}_{ij} - \mathbf{i}\frac{\mathbf{b}_{ij}^c}{2})\frac{|V_i|^2}{|\mathbf{T}_{ij}|^2} - \overline{\mathbf{Y}}_{ij}\frac{V_i\overline{V}_j}{\mathbf{T}_{ij}}, \quad \forall (i,j) \in E, \\
& S_{ji} = (\overline{\mathbf{Y}}_{ij} - \mathbf{i}\frac{\mathbf{b}_{ij}^c}{2})|V_j|^2 - \overline{\mathbf{Y}}_{ij}\frac{\overline{V}_iV_j}{\overline{\mathbf{T}}_{ij}}, \quad \forall (i,j) \in E, \\
& |S_{ij}| \le \mathbf{s}_{ij}^u, \quad \forall (i,j) \in E \cup E^R, \\
& \boldsymbol{\theta}_{ij}^{\Delta l} \le \angle(V_i\overline{V}_j) \le \boldsymbol{\theta}_{ij}^{\Delta u}, \quad \forall (i,j) \in E,
\end{cases}
$$

where V_i is the voltage, S_k^g is the power generation, S_{ij} is the power flow (all are complex variables; $\angle\cdot$ stands for the angle of a complex number) and all symbols in boldface are constants. Notice that G is the collection of generators and N is the collection of buses. For a full description on the AC-OPF problem, we refer the reader to [3] as well as [5].

We select test cases from the AC-OPF library PGLiB-OPF [3]. For each case, we solve the minimal relaxation step of the sparse HSOS hierarchy [15]. The results are displayed in Table 5. From the table, we again see that the reformulation (HSOS-$\mathbb{R}$') is several ($1.4 \sim 5$) times as fast as the reformulation (HSOS-$\mathbb{R}$).

TABLE 5
The results for the AC-OPF problem. s: the number of CPOP variables; t: the number of CPOP constraints.

Case	s	t	(HSOS-$\mathbb{R}$)			(HSOS-$\mathbb{R}$')		
			m_{sdp}	opt	time	m_{sdp}	opt	time
14_ieee	19	147	2346	1.9940e3	0.19	422	1.9940e3	0.10
30_ieee	36	297	4828	8.1959e3	0.73	836	8.1960e3	0.37
30_as	36	297	4828	5.0371e2	0.55	836	5.0371e2	0.24
39_epri	49	361	5270	1.3568e5	0.74	966	1.3579e5	0.54
89_pegase	101	1221	57888	9.4098e4	63.6	10262	9.4101e4	15.1
57_ieee	64	563	11102	3.6644e4	2.36	2008	3.6644e4	1.06
118_ieee	172	1325	25374	9.3216e4	8.27	4471	9.3216e4	2.68
162_ieee_dtc	174	1809	64874	1.0492e5	43.4	11327	1.0495e5	13.8
179_goc	208	1827	25712	6.0859e5	10.3	4368	6.0860e5	3.57
240_pserc	383	3039	52172	2.8153e6	31.9	9243	2.8170e6	10.7
300_ieee	369	2983	53946	5.3037e5	40.6	9647	5.3037e5	10.6
500_goc	671	5255	90502	3.9697e5	89.8	15918	3.9697e5	25.4
588_sdet	683	5287	79362	1.9799e5	91.7	13933	1.9749e5	21.3
793_goc	890	7019	104978	1.1194e5	105	18536	1.1222e5	31.5
1888_rte	2178	18257	280580	1.2537e6	939	47205	1.2545e6	180
2000_goc	2238	23009	455530	9.1876e5	2087	77974	9.1881e5	439

Acknowledgments. The authors would like to thank Jurij Volčič for helpful comments on an earlier preprint of this note.

REFERENCES

[1] E. D. Andersen and K. D. Andersen, *The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm*, in High Performance Optimization, vol. 33 of Applied Optimization, Springer US, 2000, pp. 197–232, https://doi.org/10.1007/978-1-4757-3216-0_8.

[2] M. Araújo, M. Huber, M. Navascués, M. Pivoluska, and A. Tavakoli, *Quantum key distribution rates from semidefinite programming*, Quantum, 7 (2023), p. 1019.

[3] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, et al., *The power grid library for benchmarking AC optimal power flow algorithms*, arXiv preprint arXiv:1908.02788, (2019).

[4] J. Bae and L.-C. Kwek, *Quantum state discrimination and its applications*, Journal of Physics A: Mathematical and Theoretical, 48 (2015), p. 083001.

[5] D. Bienstock, M. Escobar, C. Gentile, and L. Liberti, *Mathematical programming formulations for the alternating current optimal power flow problem*, 4OR, 18 (2020), pp. 249–292.

[6] C. Coey, L. Kapelevich, and J. P. Vielma, *Solving natural conic formulations with Hypatia.jl*, INFORMS Journal on Computing, 34 (2022), pp. 2686–2699, https://doi.org/https://doi.org/10.1287/ijoc.2022.1202.

[7] J. C. Gilbert, *SDOlab-A solver of real or complex number semidefinite optimization problems*, PhD thesis, INRIA Paris, 2017.

[8] J. C. Gilbert and C. Josz, *Plea for a semidefinite optimization solver in complex numbers*, PhD thesis, Inria Paris, 2017.

[9] M. X. Goemans and D. Williamson, *Approximation algorithms for max-3-cut and other*

problems via complex semidefinite programming, in Proceedings of the thirty-third annual ACM symposium on Theory of computing, 2001, pp. 443–452.

[10] C. Josz and D. K. Molzahn, *Lasserre hierarchy for large scale polynomial optimization in real and complex variables*, SIAM Journal on Optimization, 28 (2018), pp. 1017–1048.

[11] C. Lu, Y.-F. Liu, W.-Q. Zhang, and S. Zhang, *Tightness of a new and enhanced semidefinite relaxation for mimo detection*, SIAM Journal on Optimization, 29 (2019), pp. 719–742.

[12] A. Oustry, C. D'Ambrosio, L. Liberti, and M. Ruiz, *Certified and accurate sdp bounds for the acopf problem*, Electric Power Systems Research, 212 (2022), p. 108278.

[13] J. F. Sturm, *Using sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11 (1999), pp. 625–653, https://doi.org/10.1080/10556789908805766, https://doi.org/10.1080/10556789908805766.

[14] I. Waldspurger, A. d'Aspremont, and S. Mallat, *Phase recovery, maxcut and complex semidefinite programming*, Mathematical Programming, 149 (2015), pp. 47–81.

[15] J. Wang and V. Magron, *Exploiting sparsity in complex polynomial optimization*, Journal of Optimization Theory and Applications, 192 (2022), pp. 335–359.

[16] J. Wang and V. Magron, *A real moment-hsos hierarchy for complex polynomial optimization with real coefficients*, arXiv preprint arXiv:2308.14631, (2023).

[17] J. Watrous, *Semidefinite programs for completely bounded norms*, Theory Comput., 5 (2009), pp. 217–238, https://doi.org/10.4086/toc.2009.v005a011.