Block Factor-Width-Two Matrices in Semidefinite Programming

Aivar Sootla, Yang Zheng, and Antonis Papachristodoulou

Abstract—In this paper, we introduce a set of block factor-width-two matrices, which is a generalisation of factor-width-two matrices and is a subset of positive semidefinite matrices. The set of block factor-width-two matrices is a proper cone and we compute a closed-form expression for its dual cone. We use these cones to build hierarchies of inner and outer approximations of the cone of positive semidefinite matrices. The main feature of these cones is that they enable a decomposition of a large semidefinite constraint into a number of smaller semidefinite constraints. As the main application of these classes of matrices, we envision large-scale semidefinite feasibility optimisation programs including sum-of-squares (SOS) programs. We present numerical examples from SOS optimisation showcasing the properties of this decomposition.

I. INTRODUCTION

Optimisation programs with positive semidefinite (PSD) constraints (or semidefinite programs — SDPs) are one of the major computational tools in linear systems theory [1], [2]. The introduction of sum-of-squares polynomial optimisation (or SOS programming) [3], [4] (and the dual moment approach [5]) extended the use of SDPs to polynomial optimisation and thus allowed addressing many nonlinear control problems in polynomial time.

Modern SDPs (and especially SOS programs) are often large-scale, that is, the PSD constraints have large dimensions. Consequently, developing fast SDP solvers has received considerable attention in the literature. Solvers for sparse problems were developed in [6], [7], [8] (ADMM-based) and in [9], [10] (interior-point solver) and a general purpose ADMM-based solver was developed in [11]. The sparsity of the PSD constraint was also exploited in the context of SOS programming [12], [13], [14]. The key idea in these sparsity-exploiting approaches is to decompose large PSD constraints into a number of smaller PSD constraints, while the optimal objective of the program remains the same for a special class of sparsity patterns [15]. Since the PSD constraint typically induces the largest computational burden, the computational time can be significantly reduced by using these techniques. These sparsity exploiting techniques can also be used for linear control applications [16].

A related approach to speed-up SOS programming was taken in [17], where the authors replaced the PSD cone with the cone of factor-width-two matrices (which we denote $F_{2}^{N}$ where $N$ stands for the dimension of the matrix). A matrix has a factor width two if it can be represented as a sum of rank two PSD matrices [18] and hence it is also PSD. A certificate for $F_{2}^{N}$ matrices can be written as a number of second-order cone constraints, which can reduce the computational and memory burden as demonstrated in [17]. We note that $F_{2}^{N}$ matrices are also scaled diagonally dominant (SDD) as discussed in [18]. The reader unfamiliar with SDD matrices is referred to [19] for details. We only highlight that the individual entries of SDD matrices satisfy a particular set of constraints.

As discussed in [17] the size of the cone $F_{2}^{N}$ is significantly smaller than the size of the PSD cone, therefore, the restricted problem may be infeasible or the optimal solution of the $F_{2}^{N}$ program may be significantly different from the optimal solution of the original SDP. There are several approaches to bridge this restriction gap, cf. [20]. For example, one can employ factor-width-$k$ matrices, which can be decomposed into a sum of PSD matrices of rank $k$. Enforcing this constraint, however, is problematic due to a large number of $k \times k$ PSD constraints, which is $N$ choose $k$, i.e., $N \choose k$. Therefore, the computational burden can actually increase in comparison to the original SDP.

In this paper, we take a different route in order to enrich the cone of factor-width-two matrices: We draw inspiration from SDD matrices and consider their block extension. The key idea of this extension is to partition a matrix into a set of non-intersecting blocks of entries and enforce the SDD constraints on these blocks instead of the individual entries [21]. We introduce the class of block factor-width-two matrices based on the block SDD definitions from [22], [25]. A block factor-width-two matrix is also PSD and the constraint “the matrix is block factor-width-two” can be enforced using a number of PSD constraints whose size is determined by the size of the blocks. We proceed by deriving a hierarchy of inner and outer approximations of the PSD cone based on the block partition. We propose to use this approximation in SDPs by replacing the PSD cone constraint with a block “factor-width-two” constraint. The optimal objective value of the SDPs typically cannot be achieved using this technique, however, the computational cost is reduced. Striking the balance between the accuracy of the solution and the speed can be delicate in general, therefore, we envision the feasibility of SDPs without a specific sparsity structure as the main application. For example, finding a Lyapunov function certifying stability of a nonlinear system often results in a feasibility SDP without a particular sparsity structure. Therefore, in this paper, we
mainly focus on SOS programs as an application.

In Section III we cover preliminaries. In Section IV we introduce block factor-width-two matrices, a hierarchy of inner and outer approximations of the PSD cone and their SDP and SOS applications. We present numerical examples in Section V and conclude the paper in Section VI.

Notation. The matrix $A^T$ denotes the transpose of $A \in \mathbb{R}^{n \times n}$. We denote the sets of $n$ by $n$ symmetric, positive definite, positive semidefinite matrices as $\mathbb{S}_n^+$, $\mathbb{S}_n$, $\mathbb{S}_n^-$, respectively. We use $I_k$ to denote an identity matrix of dimension $k \times k$.

II. Preliminaries

A. Partitioned Matrices

We say that a matrix $A \in \mathbb{R}^{N \times N}$ has $\alpha = \{k_1, \ldots, k_p\}$-partition with $N = \sum_{i=1}^p k_i$, if $A$ can be written as

$$A = \begin{bmatrix}
A_{11} & A_{12} & \cdots & A_{1p} \\
A_{21} & A_{22} & \cdots & A_{2p} \\
\vdots & \vdots & \ddots & \vdots \\
A_{p1} & A_{p2} & \cdots & A_{pp}
\end{bmatrix},$$

where $A_{ij} \in \mathbb{R}^{k_i \times k_j}$. For a partition $\alpha = \{k_1, \ldots, k_p\}$ we define block basis matrices

$$E_{ij} = [\begin{smallmatrix}
E_i^T & E_j^T
\end{smallmatrix}] \in \mathbb{R}^{(k_i + k_j) \times N}, i \neq j,$$

where

$$I = \begin{bmatrix}
I_{k_1} & & \\
& I_{k_2} & \\
& & \ddots \\
& & & I_{k_p}
\end{bmatrix} = \begin{bmatrix}
E_1 \\
E_2 \\
\vdots \\
E_p
\end{bmatrix},$$

$$E_i = [0 \ldots I_{k_i} \ldots 0] \in \mathbb{R}^{k_i \times N}.$$

We also define a relation between a partition $\beta$ of the matrix $A$ and a coarser partition $\alpha$ of the same matrix.

Definition 1: Let $\alpha = \{k_1, \ldots, k_p\}$ and $\beta = \{l_1, \ldots, l_p\}$, where $p_1 < p_2$ and $\sum_{i=1}^{p_1} l_i = \sum_{i=1}^{p_2} k_i$. We say that $\beta$ is a sub-partition of $\alpha$ and write $\alpha \succeq \beta$, if there exist integers $\{m_i\}_{i=1}^{p_1}$ such that $k_i = \sum_{j=m_i+1}^{m_{i+1}} l_j$ and $m_1 = 1$, $m_{p_1} = p_2$, $m_i < m_{i+1}$ for all $i$.

For example, given $\alpha = \{4, 2\}$, $\beta = \{2, 2, 2\}$ and $\gamma = \{1, 1, 1, 1, 1\}$, we have $\alpha \supseteq \beta \supseteq \gamma$.

B. Semidefinite and sum-of-squares programming

The standard primal-form semidefinite program (SDP) is an optimisation problem of the form:

$$\begin{align*}
\min_x & \quad \langle C, X \rangle, \\
\text{subject to} & \quad \langle A_i, X \rangle = b_i, i = 1, \ldots, m, \\
& \quad X \in \mathbb{S}_n^+,
\end{align*}$$

where $C, A_i \in \mathbb{S}_n^+$, $i = 1, \ldots, m$ and $b \in \mathbb{R}^m$ are given problem data.

SDPs have many applications in linear systems theory, such as stabilization and $\mathcal{H}_2/\mathcal{H}_\infty$ control [2]. Also, nonlinear control problems in a polynomial field can often be written as polynomial optimisation programs: Given a set of polynomials $f_0(x), f_1(x), \ldots, f_m(x)$ (their coefficients are given) with $x \in \mathbb{R}^n$ and the vector $b \in \mathbb{R}^m$ we aim to solve

$$\begin{align*}
\min_y & \quad \langle b, y \rangle, \\
\text{subject to} & \quad f_0(x) + \sum_{i=1}^m y_i f_i(x) \geq 0, \forall x \in \mathbb{R}^n.
\end{align*}$$

Even though the nonnegativity constraint in (3) is convex, the program is infinite dimensional due to the dependence on $x$. Therefore, a tractable sum-of-squares relaxation of the nonnegative constraint is typically used. Given $x \in \mathbb{R}^n$, a polynomial $p(x)$ of degree $2d$ is called a sum-of-squares (SOS) polynomial if it can be written into a sum of squares of other polynomials of degree no greater than $d$. It is known ([24], [25]) that $p(x)$ admits an SOS decomposition if and only if there exists $Q \in \mathbb{S}_n^+$ with $N = (n+d)$ such that

$$p(x) = v_d(x)^T Q v_d(x),$$

where $v_d(x)$ is a vector of monomials of degree no greater than $d$. Replacing the nonnegative constraint with an SOS constraint yields the following optimisation programme:

$$\begin{align*}
\min_y & \quad \langle b, y \rangle, \\
\text{subject to} & \quad v_d(x)^T Q v_d(x) = \\
& \quad f_0(x) + \sum_{i=1}^m y_i f_i(x), \forall x \in \mathbb{R}^n,
\end{align*}$$

where the constraints imply that $f_0(x) + \sum_{i=1}^m y_i f_i(x)$ is an SOS polynomial. Matching the coefficients on both sides polynomial equality leads to a set of linear equality constraints on $Q$ and $y$, and we obtain an SDP of the form [2] with additional free variables.

C. Factor Width of Positive Semidefinite Matrices

It is well-known that small and medium-sized SDPs can be solved up to an arbitrary accuracy in polynomial time via interior point methods [1]. However, as the size of the PSD cone $N$ in [2] increases, the current state-of-the-art interior point algorithms become impractical in terms of memory requirements, computational burden or numerical accuracy. In the recent work [17] it was proposed to speed up semidefinite and SOS optimisation by replacing the PSD cone by a cone of factor-width-two matrices. This work is based on the following definitions from [13].

Definition 2: A matrix $X \in \mathbb{S}_n^+$ belongs to the class of factor-width-$k$ matrices (denoted as $\mathcal{FW}_n^k$) if and only if

$$X = \sum_{i=1}^s e_i^T X e_i, \quad \text{with} \quad X_i \in \mathbb{S}_k^+, \quad e_i \in \mathcal{T}_k,$$

where $\mathcal{T}_k$ is a collection of matrices $e_i \in \mathbb{R}^{k \times N}$ with every row having only one non-zero element equal to one, the columns being orthonormal, and $s = \binom{n}{k}$. 
The matrices $e_i$ can be seen as a decomposition basis for the matrix $X$. It can be shown that a dual (with respect to the trace inner product) set to $\mathcal{FW}_k^N$ can be characterised as follows

$$(\mathcal{FW}_k^N)^* = \{Z \in \mathcal{S}_N^{} | e_i Z e_i^T \in \mathcal{S}_+^k, \forall e_i \in T_k\}.$$ 

One can also show that the following hierarchy of inner and outer approximations of $\mathcal{S}_+^N$ holds:

$$\mathcal{FW}_1^N \subset \mathcal{FW}_2^N \subset \ldots \subset \mathcal{FW}_N^N = \mathcal{S}_+^N = (\mathcal{FW}_N^N)^* \subset \ldots \subset (\mathcal{FW}_2^N)^* \subset (\mathcal{FW}_1^N)^*.$$ 

Replacing $\mathcal{S}_+^N$ in (2) with $\mathcal{FW}_k^N$ leads to a restriction with multiple $k \times k$ PSD matrices. In particular, the factor-width-two matrices can be written as

$$X = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} E_{ij}^T X_{ij} E_{ij}, \text{ with } X_{ij} \in \mathcal{S}_+^2,$$

where the matrices $E_{ij}$ are defined as in Section II-A with $\alpha = \{1, \ldots, 1\}$ and $p = N$. The constraints $X_{ij} \in \mathcal{S}_+^2$ can be equivalently written as second-order cone constraints, which can be solved much faster compared to solving SDPs. This fact has been used to solve large-scale SOS programs in [17]. However, the solution from $\mathcal{FW}_N^N$ might be very conservative.

As was pointed out in [17], increasing the factor width may reduce the degree of conservatism, but this requires working with a combinatorial number $\binom{N}{k}$ of PSD cones of size $k \times k$, which is not practical.

III. BLOCK FACTOR-WIDTH-TWO MATRICES

A. Block factor-width-two matrices

In this section, we introduce the class of block factor-width-two matrices, which is less conservative than $\mathcal{FW}_N^N$ and more scalable than $\mathcal{FW}_k^N$ ($k \geq 3$).

**Definition 3:** We say that $\alpha = \{k_1, \ldots, k_p\}$-partitioned matrix $X \in \mathcal{S}_N^{}$ belongs to the class $\mathcal{FW}_{\alpha,2}^N$ if and only if

$$X = \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} E_{ij}^T X_{ij} E_{ij},$$

where $X_{ij} \in \mathcal{S}_{k_i+k_j}^+$ and $E_{ij}$ are defined in (1).

It is straightforward to show that the set $\mathcal{FW}_{\alpha,2}^N$ is a cone with a non-empty interior, which also is:

- **convex:** for any $X, Y \in \mathcal{FW}_{\alpha,2}^N$, $0 \leq \theta \leq 1$, we have that $\theta X + (1-\theta) Y \in \mathcal{FW}_{\alpha,2}^N$;
- **salient:** for any nonzero $X \in \mathcal{FW}_{\alpha,2}^N$, $-X \not\in \mathcal{FW}_{\alpha,2}^N$;
- **pointed:** the zero matrix is in $\mathcal{FW}_{\alpha,2}^N$.

We will show in what follows that this cone is additionally closed, which makes it a proper cone (closed, convex, pointed, salient cone with non-empty interior).

The main difference with the definition of factor-width-two matrices comes in the partition $\alpha$, which dictates the sizes of $X_{ij}$’s and $E_{ij}$’s, as well as their number. The number of basis matrices $E_{ij}$ is the same as in the case when we treat every block $X_{ij}$ as a scalar and apply the factor width decomposition to it. In our definition, we have a fixed partition $\alpha$ and a fixed “block factor-width”, which is equal to two. In order to create a hierarchy of approximations of $\mathcal{S}_+^N$ we can increase the “block factor-width”, which means increasing the number of basis matrices $E_{ij}$. However, we can also build a hierarchy based on the partition coarsening, which reduces the number of basis matrices $E_{ij}$.

**Theorem 1:** Given $\alpha = \{k_1, \ldots, k_p\}$, $\beta = \{\tilde{k}_1, \ldots, \tilde{k}_q\}$ and $\alpha \supseteq \beta$, we have the following inclusion:

$$\mathcal{FW}_2^N = \mathcal{FW}_{\beta,2}^N \subset \mathcal{FW}_{\alpha,2} \subset \mathcal{FW}_{\alpha,2}^N$$

where $\beta, \alpha \subseteq \{1, 1, \ldots, 1\}$, $K_1, K_2$ are positive integers and $K_1 + K_2 = N$.

**Proof.** First, $\mathcal{FW}_2^N = \mathcal{FW}_{\beta,2}^N$, $\mathcal{FW}_{\alpha,2} \subset \mathcal{FW}_{\alpha,2}^N$ hold by definition. Furthermore, $\mathcal{FW}_{\alpha,2} \subset \mathcal{FW}_{\beta,2}^N$ is true since in the decomposition for $\mathcal{FW}_{\alpha,2}$ we use PSD matrices of dimension at most $\max_{\alpha,\beta} k_i + k_j$.

In order to prove $\mathcal{FW}_2^N \subset \mathcal{FW}_{\alpha,2}$ it suffices to consider the case $\beta = \{k_1, \ldots, k_{p-1}, k_p, k_{p+1}\}$, $\alpha = \{k_1, \ldots, k_{p-1}, k_p, k_{p+1}\}$. Let $E_{\beta ij}$ for $i, j = 1, \ldots, p + 1$ be the decomposition basis for the $\beta$-partitions and $E_{\alpha ij}$ for $i, j = 1, \ldots, p$ be the decomposition basis the $\alpha$-partitions. By the premise, there exist $X_{ij} \in \mathcal{S}_{k_i+k_j}^+$ such that

$$X = \sum_{i=1}^{p} \sum_{j=i+1}^{p+1} E_{\beta ij}^T X_{ij} E_{\beta ij} = \sum_{i=1}^{p} \sum_{j=i+1}^{p+1} E_{\beta ij}^T X_{ij} E_{\beta ij} +$$

$$\sum_{i=1}^{p} E_{\beta ip}^T X_{ip} E_{\beta ip} + \sum_{i=1}^{p} E_{\beta(i+1)p}^T X_{i(p+1)} E_{\beta(i+1)p}.$$ 

We need to construct $\tilde{X}_{ij}$ so that $X$ is decomposed as:

$$X = \sum_{i=1}^{p} \sum_{j=i+1}^{p} E_{\alpha ij}^T \tilde{X}_{ij} E_{\alpha ij}. \tag{6}$$

Since the first $p - 1$ blocks in both partitions are the same, we have that $E_{\alpha ij} = E_{\beta ij}$ and $\tilde{X}_{ij} = X_{ij}$ for all $i, j < p$. Therefore, in order to obtain the decomposition (6), it remains to construct $\tilde{X}_{ip}$ for $i = 1, \ldots, p - 1$ such that

$$\sum_{i=1}^{p} E_{\alpha ip}^T \tilde{X}_{ip} E_{\alpha ip} = \sum_{i=1}^{p} E_{\beta ip}^T X_{ip} E_{\beta ip} +$$

$$\sum_{i=1}^{p} E_{\beta(i+1)p}^T X_{i(p+1)} E_{\beta(i+1)p}. \tag{7}$$

Consider the matrices $X_{ij}$ for $i < p$ and $j = p, p+1$ and split them according to the partition

$$X_{ij} = \begin{pmatrix} X_{ij}^{11} & X_{ij}^{12} \\ X_{ij}^{21} & X_{ij}^{22} \end{pmatrix}$$

for $i < p$ and $j = p, p+1$.
where $X_{ij}^{11} \in S_{+}^{k_j}$, $X_{ij}^{12} \in \mathbb{R}^{k_i \times k_j}$, $X_{ij}^{22} \in S_{+}^{k_i}$. It can be verified by direct computation that the identity \(7\) holds if $\bar{X}_{ip}$ for $i < p$ are chosen as follows:

\[
\bar{X}_{ip} = \begin{pmatrix}
0 & 0 & 0 \\
0 & X_{ip}^{11} & X_{ip}^{12} \\
0 & X_{ip}^{12} & X_{ip}^{22}
\end{pmatrix} + \frac{1}{p - 1} 
\]

Thus we complete the proof.

**Example 1:** Consider the following PSD matrix

\[
X = \begin{pmatrix}
6 & 8 & -2 & -2 \\
8 & 16 & 1 & 1 \\
-2 & 1 & 10 & -1 \\
-2 & 1 & -1 & 24
\end{pmatrix}
\]

It can be verified that $X \in \mathcal{F}\mathcal{W}_{\beta,2}$ for the partition $\beta = \{1, 1, 1\}$ and the matrices in the decomposition can be chosen as follows:

\[
X_{12} = \begin{pmatrix}
4.5 & 8 & 14.5 \\
8 & 12 & 14.5 \\
14.5 & 12 & 24
\end{pmatrix},
X_{13} = \begin{pmatrix}
1 & -2 & 6 \\
-2 & 1 & 6 \\
6 & 1 & 2
\end{pmatrix},
X_{23} = \begin{pmatrix}
1 & 1 \\
1 & 2
\end{pmatrix}
\]

If we collapse the last two entries into a block and obtain the partition $\alpha = \{1, 1, 2\}$, then we can use the constructions in Theorem 1 in order to obtain the matrices $\bar{X}_{12} = X_{12}$.

\[
\bar{X}_{13} = \begin{pmatrix}
1.5 & -2 & -2 \\
-2 & 7 & -0.5 \\
-2 & -0.5 & 9
\end{pmatrix},
\bar{X}_{23} = \begin{pmatrix}
1.5 & 1 & 1 \\
1 & 3 & -0.5 \\
1 & -0.5 & 9
\end{pmatrix}
\]

The matrices $\bar{X}_{12}, \bar{X}_{13}, \bar{X}_{23}$ are PSD, which shows that $X \in \mathcal{F}\mathcal{W}_{\alpha,2}$.

We can also describe a dual set of matrices to $\mathcal{F}\mathcal{W}_{\alpha,2}$ matrices (with respect to the trace inner product), which creates an outer approximation hierarchy for the cone $S_{+}^{N}$.

**Corollary 1:** The dual to $\mathcal{F}\mathcal{W}_{\alpha,2}$ with respect to the trace inner product is defined as:

\[
(\mathcal{F}\mathcal{W}_{\alpha,2})^* = \{ Z \in S^{N} | E_{ij} Z E_{ij}^T \in S_{+}^{k_i+k_j}, \forall 1 \leq i < j \leq p \}.
\]

Furthermore, let $\alpha = \{k_1, \ldots, k_p\}$ and $\beta = \{\bar{k}_1, \ldots, \bar{k}_q\}$, $\alpha \supseteq \beta$, then we have the following inclusions:

\[
(\mathcal{F}\mathcal{W}_{\alpha,2})^* \supset (\mathcal{F}\mathcal{W}_{\beta,2})^* \supset (\mathcal{F}\mathcal{W}_{N}^N)^* \supset (\mathcal{F}\mathcal{W}_{N,1}^{N,1_{\max(\alpha)}}(k_1+k_2))^* \supset S_{+}^{N},
\]

where $K_1, K_2$ are positive integer and $K_1 + K_2 = N$.

**Proof:** The proof of the first part follows after noticing that for any matrix $Z \in S^{N}$ such that $E_{ij} Z E_{ij}^T \in S_{+}^{k_i}$ for all $1 \leq i < j \leq p$, and for any matrix $X \in \mathcal{F}\mathcal{W}_{\alpha,2}$, we have trace($X Z$) $\geq$ 0. The proof of the second part is straightforward.

**Remark 1:** Using the terminology in [26] the cone $(\mathcal{F}\mathcal{W}_{\alpha,2})^*$ is a partially separable cone, which ensures that its dual is $\mathcal{F}\mathcal{W}_{\alpha,2}$ and $(\mathcal{F}\mathcal{W}_{\alpha,2})^*$ is a proper cone.

The major difference between our hierarchy of $\mathcal{F}\mathcal{W}_{\alpha,2}$ and the hierarchy of $\mathcal{F}\mathcal{W}_{k}$ is the number of basis matrices, which in our case is substantially lower due to two reasons: We use factor-width-two generalisations and we coarsen the partitions. Therefore the number of basis matrices is equal to $p(p - 1)/2$ for $\alpha = \{k_1, \ldots, k_p\}$, and as we make a partition coarser the number $p$ and hence the number of basis matrices decreases. We note however that the set $\mathcal{F}\mathcal{W}_{\alpha,2}$ is not contained in $\mathcal{F}\mathcal{W}_{\alpha,2}$ for $p > 2$. This is because $\mathcal{F}\mathcal{W}_{\alpha,2}$ contains all possible combinations of $e_i$'s as the basis vectors (read all possible partitions). In contrast, $\mathcal{F}\mathcal{W}_{\alpha,2}$ will not consider certain choices of partitions. Therefore, our approach has a particular advantage in applications where partitions come as a natural property of the problem.

**B. Applications to SDPs in the Standard Primal Form**

The main idea is to replace the cone $S_{+}^{N}$ with $\mathcal{F}\mathcal{W}_{\alpha,2}$ or its dual in order to obtain a restriction of the original program. Consider a restriction of (2), where we assume the matrix $X$ is partitioned according to $\alpha = \{k_1, \ldots, k_p\}$,

\[
\begin{align*}
\min_X & \langle C, X \rangle, \\
\text{subject to} & \langle A_i, X \rangle = b_i, i = 1, \ldots, m \quad (8)
\end{align*}
\]

This program can be cast in the SDP form as follows:

\[
\begin{align*}
\min_{X_{ij}} & \sum_{j=1}^{p} \sum_{i<j}^{p-1} \langle E_{ij} C E_{ij}^T, X_{ij} \rangle, \\
\text{subject to} & \sum_{j=1}^{p} \sum_{i=1}^{p} \langle E_{ij} A_i E_{ij}^T, X_{ij} \rangle = b_i, i = 1, \ldots, m \\
& X_{ij} \in S_{+}^{k_i+k_j}, 1 \leq j < l \leq p.
\end{align*}
\]

which is amenable for a straightforward implementation in standard SDP solvers such as SeDuMi [27], MOSEK [28], or SCS [11]. This program has the same number of equality constraints as (4), but the number and the dimensions of PSD constraints are different. We can also perform a relaxation of (2) by replacing $X \in S_{+}^{N}$ by $X \in (\mathcal{F}\mathcal{W}_{\alpha,2})^*$.

**IV. NUMERICAL EXAMPLES**

In our numerical examples, we used YALMIP [29] in order to reformulate the polynomial optimisation program into a standard SDP and we solve the SDPs using MOSEK [28].

\footnote{Code is available via https://github.com/zhengy09/SDPfw}
The number of SDP constraints, as discussed above is equal to the maximum number of blocks of size \(k\), which grows combinatorially when we vary \(n\) and \(d\). Therefore, another technique was proposed in [17], which replaces the constraint (12) with

\[
P(x) = (I_n \otimes v_d(x))^T Q(I_n \otimes v_d(x)), \quad \forall x \in \mathbb{R}^n
\]

where the constraint (11) is actually a linear constraint linking the coefficients of \(P(x)\) with the matrix \(Q\). The SOS programs are known to suffer from the curse of dimensionality, in particular, the size of \(Q\) grows combinatorially as we vary both \(m\) and \(d\). Therefore, we use a well-known linear technique [30] which is the SOS-relaxation, which results in the following reformulation [30]

\[
P(x) = (I_n \otimes v_d(x))^T Q(I_n \otimes v_d(x)), \quad \forall x \in \mathbb{R}^n
\]

where the constraint (11) is actually a linear constraint linking the coefficients of \(P(x)\) with the matrix \(Q\). The SOS programs are known to suffer from the curse of dimensionality, in particular, the size of \(Q\) grows combinatorially when we vary both \(m\) and \(d\). Therefore, another technique was proposed in [17], which replaces the constraint (12) with

\[
P(x) = (I_n \otimes v_d(x))^T Q(I_n \otimes v_d(x)), \quad \forall x \in \mathbb{R}^n
\]

where the constraint (11) is actually a linear constraint linking the coefficients of \(P(x)\) with the matrix \(Q\). The SOS programs are known to suffer from the curse of dimensionality, in particular, the size of \(Q\) grows combinatorially when we vary both \(m\) and \(d\). Therefore, another technique was proposed in [17], which replaces the constraint (12) with

\[
P(x) = (I_n \otimes v_d(x))^T Q(I_n \otimes v_d(x)), \quad \forall x \in \mathbb{R}^n
\]

where the constraint (11) is actually a linear constraint linking the coefficients of \(P(x)\) with the matrix \(Q\). The SOS programs are known to suffer from the curse of dimensionality, in particular, the size of \(Q\) grows combinatorially when we vary both \(m\) and \(d\). Therefore, another technique was proposed in [17], which replaces the constraint (12) with

\[
P(x) = (I_n \otimes v_d(x))^T Q(I_n \otimes v_d(x)), \quad \forall x \in \mathbb{R}^n
\]

where the constraint (11) is actually a linear constraint linking the coefficients of \(P(x)\) with the matrix \(Q\). The SOS programs are known to suffer from the curse of dimensionality, in particular, the size of \(Q\) grows combinatorially when we vary both \(m\) and \(d\). Therefore, another technique was proposed in [17], which replaces the constraint (12) with

\[
P(x) = (I_n \otimes v_d(x))^T Q(I_n \otimes v_d(x)), \quad \forall x \in \mathbb{R}^n
\]

where the constraint (11) is actually a linear constraint linking the coefficients of \(P(x)\) with the matrix \(Q\). The SOS programs are known to suffer from the curse of dimensionality, in particular, the size of \(Q\) grows combinatorially when we vary both \(m\) and \(d\). Therefore, another technique was proposed in [17], which replaces the constraint (12) with

\[
P(x) = (I_n \otimes v_d(x))^T Q(I_n \otimes v_d(x)), \quad \forall x \in \mathbb{R}^n
\]

where the constraint (11) is actually a linear constraint linking the coefficients of \(P(x)\) with the matrix \(Q\). The SOS programs are known to suffer from the curse of dimensionality, in particular, the size of \(Q\) grows combinatorially when we vary both \(m\) and \(d\). Therefore, another technique was proposed in [17], which replaces the constraint (12) with

\[
P(x) = (I_n \otimes v_d(x))^T Q(I_n \otimes v_d(x)), \quad \forall x \in \mathbb{R}^n
\]

where the constraint (11) is actually a linear constraint linking the coefficients of \(P(x)\) with the matrix \(Q\). The SOS programs are known to suffer from the curse of dimensionality, in particular, the size of \(Q\) grows combinatorially when we vary both \(m\) and \(d\). Therefore, another technique was proposed in [17], which replaces the constraint (12) with

\[
P(x) = (I_n \otimes v_d(x))^T Q(I_n \otimes v_d(x)), \quad \forall x \in \mathbb{R}^n
\]

where the constraint (11) is actually a linear constraint linking the coefficients of \(P(x)\) with the matrix \(Q\). The SOS programs are known to suffer from the curse of dimensionality, in particular, the size of \(Q\) grows combinatorially when we vary both \(m\) and \(d\). Therefore, another technique was proposed in [17], which replaces the constraint (12) with

\[
P(x) = (I_n \otimes v_d(x))^T Q(I_n \otimes v_d(x)), \quad \forall x \in \mathbb{R}^n
\]

where the constraint (11) is actually a linear constraint linking the coefficients of \(P(x)\) with the matrix \(Q\). The SOS programs are known to suffer from the curse of dimensionality, in particular, the size of \(Q\) grows combinatorially when we vary both \(m\) and \(d\). Therefore, another technique was proposed in [17], which replaces the constraint (12) with

\[
P(x) = (I_n \otimes v_d(x))^T Q(I_n \otimes v_d(x)), \quad \forall x \in \mathbb{R}^n
\]
Using the $\mathcal{FW}_{1,2}^N$ restriction provides with a larger set of solutions than $\mathcal{FW}_{1,2}$. For example, the polynomial matrix:

$$P(x) = \begin{bmatrix} 4x^2 + 9y^2 + 0.315 & x + y & x + y \\ x + y & 9x^2 + 4y^2 + 0.315 & x + y \\ x + y & x + y & x^2 + 25y^2 + 0.315 \end{bmatrix}$$

satisfies the constraints \[15, 16\], but does not satisfy the constraints \[13, 14\].

We further test our approach on the following program:

$$\max_{P(x), \gamma} \quad \gamma$$

$$P(x) - \gamma I \in \mathbb{S}^n_+, \quad \forall x \in \mathbb{R}^3,$$  \hspace{1cm} (17)

where every entry of $P(x)$ is a random polynomial of degree two in three variables, and we vary the dimension of $P(x)$. The computational results are depicted in Table II. Our restriction offers faster computational solutions with almost the same optimal objectives compared to the standard SOS technique, while the technique \[17\] provides even faster solutions, but their quality is worse.

| Table II | Computational results for Section IV-B |
|-----------|---------------------------------------|
| $n$      | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
| -----------|---------------------------------------|
| SOS       | 5.28 | 14.3 | 35.9 | 87.2 | 175.0 | 316.0 | 487.8 |
| $\mathcal{FW}_{1,2}$ | 7.90 | 10.8 | 16.6 | 25.3 | 36.0 | 57.4 | 71.4 |
| $\mathcal{FW}_2$ | 1.04 | 1.1 | 1.3 | 1.6 | 2.1 | 2.6 | 3.3 |
| Objective value | 149.0 | 266.5 | 316.2 | 460.8 | 562.0 | 746.9 | 919.8 |
| $\mathcal{FW}_{1,2}$ | 149.0 | 266.5 | 316.2 | 460.8 | 562.0 | 746.9 | 919.8 |
| $\mathcal{FW}_2$ | 154.4 | 270.3 | 324.8 | 477.7 | 570.9 | 762.2 | 961.7 |

V. CONCLUSION AND DISCUSSION

We introduced a novel class of matrices and presented a hierarchy of inner and outer approximations of the cone of positive semidefinite (PSD) matrices. Both inner and outer approximations are proper cones and enjoy useful duality relations, furthermore, the inclusion certificates for these cones is a set of PSD constraints smaller than the size of the matrix. This allows deriving a hierarchy of scalable relaxations and restrictions of semidefinite programs (SDPs). The inner approximations (cones $\mathcal{FW}_{1,2}^N$) are built by partitioning the matrix into a non-intersecting set of entries. It is not entirely clear at the moment how to build “the best” partition in terms of the solution of the particular SDP. However, in some problems, the partition comes naturally from the problem formulation, e.g., the matrix-version of SOS programs discussed in Section IV-B. Our numerical experiments suggest that these hierarchies can be used for dense large-scale SDPs, which arise in SOS programming.

Our future work will investigate the consequences of block factor-width-two matrices in relevant control applications that involve SDPs. Also, it would be interesting to incorporate the properties of block factor-width-two matrices in the development of first-order algorithms (e.g., the solvers \[11, 6\]) for solving general SDPs.

REFERENCES

[1] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2004.
[2] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, Linear matrix inequalities in system and control theory. SIAM, 1994, vol. 15.
[3] V. Powers and T. Wörmann, “An algorithm for sums of squares of real polynomials,” J. Pure Applied Algebra, vol. 127, no. 1, pp. 99–104, 1998.
[4] P. A. Parrilo, “Semidefinite programming relaxations for semialgebraic problems,” Math. Prog. vol. 96, no. 2, pp. 293–320, 2003.
[5] J.-B. Lasserre, Moments, positive polynomials and their applications. World Scientific, 2010, vol. 1.
[6] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Gouart, and A. Wynn, “Fast ADMM for semidefinite programs with chordal sparsity,” in Proc Am Control Conf, 2017, pp. 3335–3340.
[7] ——, “Fast ADMM for homogeneous self-dual embeddings of sparse SDPs,” Proc IFAC-PapersOnLine, vol. 50, no. 1, pp. 8411–8416, 2017.
[8] ——, “Chordal decomposition in operator-splitting methods for sparse semidefinite programs,” Math Prog A, 2019.
[9] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota, “Exploiting sparsity in semidefinite programming via matrix completion II: Implementation and numerical results,” Mathematical Programming, vol. 95, no. 2, pp. 303–327, 2003.
[10] S. Kim, M. Kojima, M. Mevissen, and M. Yamashita, “Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion,” Math Prog, vol. 129, no. 1, pp. 33–68, 2011.
[11] B. O’Donoghue, E. Chu, N. Parikh, and S. Boyd, “Conic optimization via operator splitting and homogeneous self-dual embedding,” J Optimization Theory Appl, vol. 169, no. 3, pp. 1042–1068, June 2016.
[12] H. Waki, S. Kim, M. Kojima, and M. Muramatsu, “Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity,” SIAM Journal on Optimization, vol. 17, no. 1, pp. 218–242, 2006.
[13] Y. Zheng, G. Fantuzzi, and A. Papachristodoulou, “Sparse sum-of-squares (SOS) optimization: A bridge between DSOS/SDSOS and SOS optimization for sparse polynomials,” in To appear in Proc Am Control Conf, 2019.
[14] ——, “Decomposition and completion of sum-of-squares matrices,” in Proc IEEE Conf Decision Control, 2018, pp. 4026–4031.
[15] L. Vandenberghe, M. S. Andersen, et al., “Chordal graphs and semidefinite optimization,” Foundations and Trends® in Optimization, vol. 1, no. 4, pp. 241–433, 2015.
[16] Y. Zheng, M. Kamgarpour, A. Sootla, and A. Papachristodoulou, “Scalable analysis of linear networked systems via chordal decomposition,” in Proc European Control Conf, 2018, pp. 2260–2265.
[17] A. A. Ahmadi and A. Majumdar, “DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization,” arXiv preprint arXiv:1706.02586, 2017.
[18] E. G. Boman, D. Chen, O. Parekh, and S. Toledo, “On factor width and symmetric H-matrices,” Linear algebra and its applications, vol. 405, pp. 239–248, 2005.
[19] A. Berman and R. J. Plemmons, Nonnegative Matrices in the Mathematical Sciences. SIAM, 1994, vol. 9.
[20] A. A. Ahmadi and G. Hall, “Sum of squares basis pursuit with linear and second order cone programming,” Contemporary Mathematics, vol. 685, pp. 25–54, 2017.
[21] D. Feingold and R. Varga, “Block diagonally dominant matrices and generalizations of the Gerschgorin circle theorem,” Pacific J. Math, vol. 12, no. 4, pp. 1241–1250, 1962.
[22] A. Sootla and J. Anderson, “On existence of solutions to structured Lyapunov inequalities,” in Proc of Am Control Conf. IEEE, 2016, pp. 7013–7018.
[23] A. Sootla, Y. Zheng, and A. Papachristodoulou, “Block-diagonal solutions to Lyapunov inequalities and generalizations of diagonal dominance,” in Proc IEEE Conf Decision Control, Dec 2017.
[24] A. Papachristodoulou and S. Prajna, “A tutorial on sum of squares techniques for systems analysis,” in Proc Am Control Conf. IEEE, 2005, pp. 2686–2700.
[25] P. A. Parrilo, “Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization,” Ph.D. dissertation, California Institute of Technology, 2004.
[26] Y. Sun, M. S. Andersen, and L. Vandenberghe, “Decomposition in conic optimization with partially separable structure,” SIAM Journal on Optimization, vol. 24, no. 2, pp. 873–897, 2014.

[27] J. F. Sturm, “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,” Optim. Method. Softw., vol. 11–12, pp. 625–653, 1999.

[28] 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. Springer, 2000, pp. 197–232.

[29] J. Löfberg, “YALMIP: A toolbox for modeling and optimization in MATLAB,” in Proc. IEEE Int. Symposium Comput.-Aided Control Syst. Des., Taipei, Taiwan, 2004, pp. 284–289.

[30] K. Gatermann and P. A. Parrilo, “Symmetry groups, semidefinite programs, and sums of squares,” J Pure Appl Algebra, vol. 192, no. 1-3, pp. 95–128, 2004.