ISSN 1342-2804

Research Reports on Mathematical and Computing Sciences

Exploiting Sparsity in Linear and Nonlinear Matrix Inequalities via Positive Semidefinite Matrix Completion

Sunyoung Kim, Masakazu Kojima, Martin Mevissen and Makoto Yamashita

January 2009, B–452
Revised in February 2010

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

SERIES B: Operations Research

Exploiting Sparsity in Linear and Nonlinear Matrix Inequalities via Positive Semidefinite Matrix Completion

Sunyoung Kim[⋆], Masakazu Kojima[†], Martin Mevissen[‡] and Makoto Yamashita[♯]

January 2009, Revised in February 2010.

Abstract.

A basic framework for exploiting sparsity via positive semidefinite matrix completion is presented for an optimization problem with linear and nonlinear matrix inequalities. The sparsity, characterized with a chordal graph structure, can be detected in the variable matrix or in a linear or nonlinear matrix-inequality constraint of the problem. We classify the sparsity in two types, the domain-space sparsity (d-space sparsity) for the symmetric matrix variable in the objective and/or constraint functions of the problem, which is required to be positive semidefinite, and the range-space sparsity (r-space sparsity) for a linear or nonlinear matrix-inequality constraint of the problem. Four conversion methods are proposed in this framework: two for exploiting the d-space sparsity and the other two for exploiting the r-space sparsity. When applied to a polynomial semidefinite program (SDP), these conversion methods enhance the structured sparsity of the problem called the correlative sparsity. As a result, the resulting polynomial SDP can be solved more effectively by applying the sparse SDP relaxation. Preliminary numerical results on the conversion methods indicate their potential for improving the efficiency of solving various problems.

Key words.

Semidefinite Program, Matrix Inequalities, Polynomial Optimization, Positive Semidefinite Matrix Completion, Sparsity, Chordal Graph

[⋆] Department of Mathematics, Ewha W. University, 11-1 Dahyun-dong, Sudaemoon-gu, Seoul 120-750 Korea. S. Kim's research was supported by KRF 2008-531-C00013. *skim@ewha.ac.kr*

[†] Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan. M. Kojima's research was supported by Grant-in-Aid for Scientific Research (B) 19310096. *kojima@is.titech.ac.jp*

[‡] Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan. M. Mevissen's research was supported by the Doctoral Scholarship of the German Academic Exchange Service (DAAD). *mevissen.m.aa@m.titech.ac.jp*

[♯] Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan. M. Yamashita's research was supported by Grant-in-Aid for Young Scientists (B) 18710141. *Makoto.Yamashita@is.titech.ac.jp*

1 Introduction

Optimization problems with nonlinear matrix inequalities, including quadratic and polynomial matrix inequalities, are known as hard problems. They frequently belong to large-scale optimization problems. Exploiting sparsity thus has been one of the essential tools for solving such large-scale optimization problems. We present a basic framework for exploiting the sparsity characterized in terms of a chordal graph structure via positive semidefinite matrix completion [7]. Depending on where the sparsity is observed, two types of sparsities are studied: *the domain-space sparsity (d-space sparsity)* for a symmetric matrix $\boldsymbol{X}$ that appears as a variable in objective and/or constraint functions of a given optimization problem and is required to be positive semidefinite, and *the range-space sparsity (r-space sparsity)* for a matrix inequality involved in the constraint of the problem.

The d-space sparsity is basically equivalent to the sparsity studied by Fukuda et. al [6] for an equality standard form SDP[1]. See also [20]. Two methods, the completion method and the conversion method, were proposed to exploit the aggregated sparsity pattern over all coefficient matrices of the linear objective and constraint functions via the positive semidefinite matrix completion. Their conversion method transforms an equality standard form of SDP with a single (large) matrix variable $\boldsymbol{X}$ in the space $\mathbb{S}^n$ of $n \times n$ real symmetric matrices to an SDP with multiple smaller matrix variables $\boldsymbol{X}^1, \boldsymbol{X}^2, \ldots, \boldsymbol{X}^p$ and some additional linear equality constraints. We can interpret their aggregated sparsity pattern as the set of row and column index pairs (i, j) such that the value of X_{ij} is relevant (or necessary) in the evaluation of the linear objective and/or constraint functions. Thus we call their aggregated sparsity as the d-space sparsity and their method as a d-space conversion method in this paper. With this interpretation, their conversion method can be directly extended to a d-space conversion method for more general optimization problems. One of the two d-space conversion methods proposed in this paper corresponds to this extension, and the other d-space conversion method is an extension of the method used for the sparse SDP relaxation of polynomial optimization problems in [23, 24] and for the sparse SDP relaxation of a sensor network localization problem in [11].

The r-space sparsity concerns with a matrix inequality

$$\boldsymbol{M}(\boldsymbol{y}) \succeq \boldsymbol{O}, \tag{1}$$

involved in a general nonlinear optimization problem. Here $\boldsymbol{M}$ denotes a mapping from the s-dimensional Euclidean space $\mathbb{R}^s$ into $\mathbb{S}^n$, and $\boldsymbol{A} \succeq \boldsymbol{O}$ implies that $\boldsymbol{A} \in \mathbb{S}^n$ is positive semidefinite. If $\boldsymbol{M}$ is linear, (1) is known as a linear matrix inequality (LMI), which appears in the constraint of a dual standard form of SDP. If each element of $\boldsymbol{M}(\boldsymbol{y})$ is a multivariate polynomial function in $\boldsymbol{y} \in \mathbb{R}^s$, (1) is called a polynomial matrix inequality and the SDP relaxation [8, 9, 14, 15, 16, 18], which is an extension of the SDP relaxation [17] for polynomial optimization problems, can be applied to (1). We assume a similar chordal graph structured sparsity as the d-space sparsity (or the aggregated sparsity by Fukuda et al. [6]) on the set of row and column index pairs (i, j) of the mapping $\boldsymbol{M}$ such that M_{ij} is not identically zero, *i.e.*, $M_{ij}(\boldsymbol{y}) \neq 0$ for some $\boldsymbol{y} \in \mathbb{R}^s$. A representative example satisfying the r-space sparsity can be found with tridiagonal $\boldsymbol{M}$. We do not impose any additional

[1] This paper is concerned with linear, nonlinear, polynomial and quadratic SDPs. We simply say an SDP for a linear SDP.

assumption on (1) to derive a r-space conversion method. When $\boldsymbol{M}$ is polynomial in $\boldsymbol{y} \in \mathbb{R}^s$, we can effectively combine it with the sparse SDP relaxation method [14, 16] for polynomial optimization problems over symmetric cones to solve (1).

We propose two methods to exploit the r-space sparsity. One may be regarded as a dual of the d-space conversion method by Fukuda et. al [6]. More precisely, it exploits the sparsity of the mapping $\boldsymbol{M}$ in the range space via a dual of the positive semidefinite matrix completion to transform the matrix inequality (1) to a system of multiple matrix inequalities with smaller sizes and an auxiliary vector variable $\boldsymbol{z}$ of some dimension q. The resulting matrix inequality system is of the form

$$\widetilde{\boldsymbol{M}}^k(\boldsymbol{y}) - \widetilde{\boldsymbol{L}}^k(\boldsymbol{z}) \succeq \boldsymbol{O} \ (k = 1, 2, \ldots, p), \tag{2}$$

and $\boldsymbol{y} \in \mathbb{R}^s$ is a solution of (1) if and only if it satisfies (2) for some $\boldsymbol{z}$. Here $\widetilde{\boldsymbol{M}}^k$ denotes a mapping from $\mathbb{R}^s$ into the space of symmetric matrices with some size and $\widetilde{\boldsymbol{L}}^k$ a linear mapping from $\mathbb{R}^q$ into the space of symmetric matrices with the same size. The sizes of symmetric matrix valued mappings $\widetilde{\boldsymbol{M}}^k$ $(k = 1, 2, \ldots, p)$ and the dimension q of the auxiliary variable vector $\boldsymbol{z}$ are determined by the r-space sparsity pattern of $\boldsymbol{M}$. For example, if $\boldsymbol{M}$ is tridiagonal, the sizes of $\widetilde{\boldsymbol{M}}^k$ are all 2×2 and $q = n-2$. The other r-space conversion method corresponds to a dual of the second d-space conversion method mentioned previously.

Another type of sparsity discussed in this paper is *the correlative sparsity* [13], which has been used under two different circumstances. First, in the primal-dual interior-point method for solving a linear optimization problem over symmetric cones that includes an SDP as a special case, the correlative sparsity of the problem characterizes the sparsity of the Schur complement matrix. We note that the Schur complement matrix is the coefficient matrix of a system of linear equations that is solved at each iteration of the primal-dual interior-point method by the Cholesky factorization to compute a search direction. As the Cholesky factor of the Schur complement matrix becomes sparse, each iteration is executed more efficiently. Second, in the sparse SDP relaxation [14, 16, 23, 24] of a polynomial optimization problem and a polynomial SDP, the correlative sparsity is used for its application to the problem. In addition, the correlative sparsity of the original problem is inherited to its SDP relaxation. We discuss how the d-space and r-space conversion methods enhance the correlative sparsity.

The main contribution of this paper is to extend the d-space conversion method, which was originally proposed by Fukuda et al. [6] for linear SDPs, to general nonlinear optimization problems with positive semidefinite symmetric matrix variables, and introduce the r-space sparsity in linear and nonlinear matrix inequalities. In addition, this paper includes a survey of the conversion techniques [6, 11, 20, 24] developed for sparse SDPs via positive semidefinite matrix completion.

The organization of the paper is as follows: In Section 2, to illustrate the d-sparsity, the r-sparsity, and the correlative sparsity, a very sparse SDP is shown as an example. It is followed by the introduction of the positive semidefinite matrix completion, a chordal graph, and their basic properties. In Section 3, we describe two d-space conversion methods using positive semidefinite matrix completion. Section 4 includes the discussion on duality in positive semidefinite matrix completion, and Section 5 is devoted to two r-space conversion methods based on the duality. In Section 6, we show how the d-space and r-space conversion methods enhance the correlative sparsity. In Section 7, the r-space conversion methods combined with the d-space conversion methods are applied to sparse quadratic SDPs, and preliminary numerical results are provided. The numerical results are only to assess the effectiveness of the proposed d- and r-space conversion methods for solving sparse optimization problems with linear and nonlinear matrix inequalities. There remain important issues on implementation of the methods and how to apply the methods to practical problems. More comprehensive numerical experiments for various problems are necessary to evaluate the numerical efficiency of the methods. These issues are discussed briefly in Section 8.

2 Preliminaries

2.1 An SDP example

A simple SDP example is shown to illustrate the three types of sparsities considered in this paper, the d-space sparsity, the r-space sparsity, and the correlative sparsity that characterizes the sparsity of the Schur complement matrix. These sparsities are discussed in Sections 3, 5 and 6, respectively.

Let $\boldsymbol{A}^0$ be a tridiagonal matrix in $\mathbb{S}^n$ such that $A^0_{ij} = 0$ if $|i-j| > 1$, and define a mapping $\boldsymbol{M}$ from $\mathbb{S}^n$ into $\mathbb{S}^n$ by

$$\boldsymbol{M}(\boldsymbol{X}) \;=\; \begin{pmatrix} 1-X_{11} & 0 & 0 & \dots & 0 & X_{12} \\ 0 & 1-X_{22} & 0 & \dots & 0 & X_{23} \\ 0 & 0 & \ddots & & 0 & X_{34} \\ \dots & \dots & \dots & \ddots & \dots & \dots \\ 0 & 0 & 0 & & 1-X_{n-1,n-1} & X_{n-1,n} \\ X_{21} & X_{32} & X_{43} & \dots & X_{n,n-1} & 1-X_{nn} \end{pmatrix}$$

for every $\boldsymbol{X} \in \mathbb{S}^n$. Consider an SDP

$$\text{minimize } \boldsymbol{A}^0 \bullet \boldsymbol{X} \text{ subject to } \boldsymbol{M}(\boldsymbol{X}) \succeq \boldsymbol{O},\ \boldsymbol{X} \succeq \boldsymbol{O}. \tag{3}$$

Among the elements X_{ij} $(i = 1, 2, \dots, n,\ j = 1, 2, \dots, n)$ of the matrix variable $\boldsymbol{X} \in \mathbb{S}^n$, the elements X_{ij} with $|i-j| \leq 1$ are relevant and all other elements X_{ij} with $|i-j| > 1$ are unnecessary in evaluating the objective function $\boldsymbol{A}^0 \bullet \boldsymbol{X}$ and the matrix inequality $\boldsymbol{M}(\boldsymbol{X}) \succeq \boldsymbol{O}$. Hence, we can describe the d-sparsity pattern as a symbolic tridiagonal matrix with the nonzero symbol $\star$

$$\begin{pmatrix} \star & \star & 0 & \dots & 0 & 0 \\ \star & \star & \star & \dots & 0 & 0 \\ 0 & \star & \star & \ddots & 0 & 0 \\ \dots & \dots & \ddots & \ddots & \ddots & \dots \\ 0 & 0 & \dots & \ddots & \star & \star \\ 0 & 0 & \dots & \dots & \star & \star \end{pmatrix}.$$

On the other hand, the r-space sparsity pattern is described as

$$\begin{pmatrix} \star & 0 & \dots & 0 & \star \\ 0 & \star & \dots & 0 & \star \\ \dots & \dots & \ddots & \dots & \dots \\ 0 & 0 & \dots & \star & \star \\ \star & \star & \dots & \star & \star \end{pmatrix}.$$

Applying the d-space conversion method using basis representation described in Section 3.2, and the r-space conversion method using clique trees presented in Section 5.1, we can reduce the SDP (3) to

$$\left.\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{n-1} \left(A^0_{ii} X_{ii} + 2A^0_{i,i+1} X_{i,i+1} \right) + A^0_{nn} X_{nn} \\ \text{subject to} & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} X_{11} & -X_{12} \\ -X_{21} & -z_1 \end{pmatrix} \succeq \boldsymbol{O}, \\ & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} X_{ii} & -X_{i,i+1} \\ -X_{i+1,i} & z_{i-1} - z_i \end{pmatrix} \succeq \boldsymbol{O} \ (i = 2, 3, \dots, n-2), \\ & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} X_{n-1,n-1} & -X_{n-1,n} \\ -X_{n,n-1} & X_{n,n} + z_{n-2} \end{pmatrix} \succeq \boldsymbol{O}, \\ & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} -X_{ii} & -X_{i,i+1} \\ -X_{i+1,i} & -X_{i+1,i+1} \end{pmatrix} \succeq \boldsymbol{O} \ (i = 1, 2, \dots, n-1). \end{array}\right\} \tag{4}$$

This problem has $(3n-3)$ real variables X_{ii} $(i = 1, 2, \dots, n)$, $X_{i,i+1}$ $(i = 1, 2, \dots, n-1)$ and z_i $(i = 1, 2, \dots, n-2)$, and $(2n-1)$ linear matrix inequalities with size 2×2. Since the original SDP (3) involves an $n \times n$ matrix variable $\boldsymbol{X}$ and an $n \times n$ matrix inequality $\boldsymbol{M}(\boldsymbol{X}) \succeq \boldsymbol{O}$, we can expect to solve the SDP (4) much more efficiently than the SDP (3) as n becomes larger.

We can formulate both SDPs in terms of a dual standard form for SeDuMi [22]:

$$\text{maximize } \boldsymbol{b}^T \boldsymbol{y} \text{ subject to } \boldsymbol{c} - \boldsymbol{A}^T \boldsymbol{y} \succeq \boldsymbol{0},$$

where $\boldsymbol{b}$ denotes an ℓ-dimensional column vector, $\boldsymbol{A}$ an $\ell \times m$ matrix and $\boldsymbol{c}$ an m-dimensional column vector for some positive integers ℓ and m. See (36) in Section 6 for the case of SDP (3). Table 1 shows numerical results on the SDPs (3) and (4) solved by SeDuMi. We observe that the SDP (4) greatly reduces the size of the coefficient matrix $\boldsymbol{A}$, the number of nonzeros in $\boldsymbol{A}$ and the maximum SDP block compared to the original SDP (3). In addition, it should be emphasized that the $\ell \times \ell$ Schur complement matrix is sparse in the SDP (4) while it is fully dense in the the original SDP (3). As shown in Figure 1, the Schur complement matrix in the SDP (4) allows a very sparse Cholesky factorization. The sparsity of the Schur complement matrix is characterized by the correlative sparsity whose definition is given in Section 6. Notice "a hidden correlative sparsity" in the SDP (3), that is, each element X_{ij} of the matrix variable $\boldsymbol{X}$ appears at most once in the elements of $\boldsymbol{M}(\boldsymbol{X})$. This leads to the correlative sparsity when the SDP (3) is decomposed into the SDP (4). The sparsity of the Schur complement matrix and the reduction in the size of matrix variable from 10000 to 2 are the main reasons that SeDuMi can solve the largest SDP in Table 1 with a 29997×79992 coefficient matrix $\boldsymbol{A}$ in less than 100 seconds. In Section 6, we discuss in detail how the exploitation of the d- and r-space sparsities contributes to increasing the sparsity of the Schur complement matrix.

	SeDuMi CPU time in seconds (sizeA, nnzA, maxBl, nnzSchur)	
n	the SDP (3)	the SDP (4)
10	0.2 (55×200,128,10,3025)	0.1 (27×72, 80,2,161)
100	1091.4 (5050×20000,10298,100,25502500)	0.6 (297×792,890,2,1871)
1000	-	6.3 (2997×7992,8990,2,18971)
10000	-	99.2 (29997×79992,89990,2,189971)

Table 1: Numerical results on the SDPs (3) and (4). Here sizeA denotes the size of the coefficient matrix $\boldsymbol{A}$, nnzA the number of nonzero elements in $\boldsymbol{A}$, maxBl the maximum SDP block size, and nnzSchur the number of nonzeros in the Schur complement matrix. "-" means out of memory error.

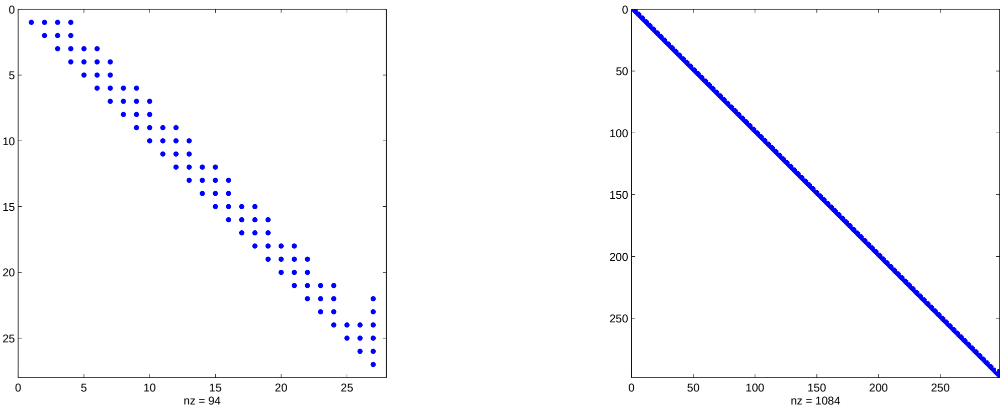

Figure 1: The sparsity pattern of the Cholesky factor of the Schur complement matrix for the SDP (4) with $n = 10$ and $n = 100$.

2.2 Notation and symbols

Let $N = \{1, 2, \ldots, n\}$ denote the set of row and column indices of $n \times n$ symmetric matrices. A problem of positive semidefinite matrix completion is: Given an $n \times n$ partial symmetric matrix $\boldsymbol{X}$ with entries specified in a proper subset F of $N \times N$, find an $n \times n$ positive semidefinite symmetric matrix $\overline{\boldsymbol{X}}$ satisfying $\overline{X}_{ij} = X_{ij}$ $((i,j) \in F)$ if it exists. If $\overline{\boldsymbol{X}}$ is a solution of this problem, we say that $\boldsymbol{X}$ is completed to a positive semidefinite symmetric matrix $\overline{\boldsymbol{X}}$. For example, the following 3×3 partial symmetric matrix

$$\boldsymbol{X} = \begin{pmatrix} 3 & 3 & \\ 3 & 3 & 2 \\ & 2 & 2 \end{pmatrix}$$

is completed to a 3×3 positive semidefinite symmetric matrix

$$\overline{\boldsymbol{X}} = \begin{pmatrix} 3 & 3 & 2 \\ 3 & 3 & 2 \\ 2 & 2 & 2 \end{pmatrix}.$$

For a class of problems of positive semidefinite matrix completion, we discuss the existence of a solution and its characterization in this section. This provides a theoretical basis for both d- and r-space conversion methods.

Let us use a graph $G(N, E)$ with the node set $N = \{1, 2, \ldots, n\}$ and an edge set $E \subseteq N \times N$ to describe a class of $n \times n$ partial symmetric matrices. We assume that $(i,i) \notin E$, *i.e.*, the graph $G(N, E)$ has no loop. We also assume that if $(i,j) \in E$, then $(j,i) \in E$, and (i,j) and (j,i) are interchangeably identified. Define

$$\begin{aligned} E^{\bullet} &= E \cup \{(i,i) : i \in N\}, \\ \mathbb{S}^n(E,?) &= \text{the set of } n \times n \text{ partial symmetric matrices with entries} \\ &\ \text{specified in } E^{\bullet}, \\ \mathbb{S}^n_+(E,?) &= \{\boldsymbol{X} \in \mathbb{S}^n(E,?) : \exists \overline{\boldsymbol{X}} \in \mathbb{S}^n_+; \overline{X}_{ij} = X_{ij} \text{ if } (i,j) \in E^{\bullet}\} \\ &\ \text{(the set of } n \times n \text{ partial symmetric matrices with entries} \\ &\ \text{specified in } E^{\bullet} \text{ that can be completed to positive} \\ &\ \text{semidefinite symmetric matrices).} \end{aligned}$$

For a graph $G(N, E)$ shown in Figure 2 as an illustrative example, we have

$$\mathbb{S}^6(E,?) = \left\{ \begin{pmatrix} X_{11} & & & & & X_{16} \\ & X_{22} & & & & X_{26} \\ & & X_{33} & X_{34} & & X_{36} \\ & & X_{43} & X_{44} & X_{45} & \\ & & & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & & X_{65} & X_{66} \end{pmatrix} : X_{ij} \in \mathbb{R} \ \ (i,j) \in E^{\bullet} \right\}. \qquad (5)$$

Let

$$\#C = \text{the number of elements in } C \text{ for every } C \subseteq N,$$

$$\begin{array}{rcl} \mathbb{S}^C & = & \{\boldsymbol{X} \in \mathbb{S}^n : X_{ij} = 0 \text{ if } (i,j) \notin C \times C\} \text{ for every } C \subseteq N, \\ \mathbb{S}^C_+ & = & \{\boldsymbol{X} \in \mathbb{S}^C : \boldsymbol{X} \succeq \boldsymbol{O}\} \text{ for every } C \subseteq N, \\ \boldsymbol{X}(C) & = & \widetilde{\boldsymbol{X}} \in \mathbb{S}^C \text{ such that } \widetilde{X}_{ij} = X_{ij} \ ((i,j) \in C \times C) \\ & & \text{for every } \boldsymbol{X} \in \mathbb{S}^n \text{ and every } C \subseteq N, \\ J(C) & = & \{(i,j) \in C \times C : 1 \leq i \leq j \leq n\} \text{ for every } C \subseteq N. \end{array}$$

Note that $\boldsymbol{X} \in \mathbb{S}^C$ is an $n \times n$ matrix although $X_{ij} = 0$ for every $(i,j) \notin C \times C$. Thus, $\boldsymbol{X} \in \mathbb{S}^C$ and $\boldsymbol{X'} \in \mathbb{S}^{C'}$ can be added even when C and C' are distinct subsets of N. When all matrices involved in an equality or a matrix inequality belong to $\mathbb{S}^C$, matrices in $\mathbb{S}^C$ are frequently identified with the $\#C \times \#C$ matrix whose elements are indexed with $(i,j) \in C \times C$. If $N = \{1,2,3\}$ and $C = \{1,3\}$, then a matrix variable $\boldsymbol{X} \in \mathbb{S}^C \subset \mathbb{S}^n$ has full and compact representations as follows:

$$\boldsymbol{X} = \begin{pmatrix} X_{11} & 0 & X_{13} \\ 0 & 0 & 0 \\ X_{31} & 0 & X_{33} \end{pmatrix} \text{ and } \boldsymbol{X} = \begin{pmatrix} X_{11} & X_{13} \\ X_{31} & X_{33} \end{pmatrix}.$$

It should be noted that $\boldsymbol{X} \in \mathbb{S}^C \subset \mathbb{S}^n$ has elements X_{ij} with $(i,j) \in C \times C$ in the 2×2 compact representation on the right. Let

$$\begin{array}{rcl} \boldsymbol{E}_{ij} & = & \text{the } n \times n \text{ symmetric matrix with 1 in } (i,j)\text{th and } (j,i)\text{th} \\ & & \text{elements and 0 elsewhere} \end{array}$$

for every $(i,j) \in N \times N$. Then $\boldsymbol{E}_{ij}$ $(1 \leq i \leq j \leq n)$ form a basis of $\mathbb{S}^n$. Obviously, if $i,\ j \in C \subseteq N$, then $\boldsymbol{E}_{ij} \in \mathbb{S}^C$. We also observe the identity

$$\boldsymbol{X}(C) = \sum_{(i,j) \in J(C)} \boldsymbol{E}_{ij} X_{ij} \text{ for every } C \subseteq N. \tag{6}$$

This identity is utilized in Section 3.

2.3 Positive semidefinite matrix completion

Let $G(N,E)$ be a graph and C_k $(k = 1,2,\ldots,p)$ be the maximal cliques of $G(N,E)$. We assume that $\boldsymbol{X} \in \mathbb{S}^n(E,?)$. The condition $\boldsymbol{X}(C_k) \in \mathbb{S}^{C_k}_+$ $(k = 1,2,\ldots,p)$ is necessary for $\boldsymbol{X} \in \mathbb{S}^n_+(E,?)$. For the graph $G(N,E)$ shown in Figure 2, the maximal cliques are $C_1 = \{1,6\}$, $C_2 = \{2,6\}$, $C_3 = \{3,4\}$, $C_4 = \{3,6\}$, $C_5 = \{4,5\}$ and $C_6 = \{5,6\}$. Hence, the necessary condition for $\boldsymbol{X} \in \mathbb{S}^6(E,?)$ to be completed to a positive semidefinite matrix is that its 6 principal submatrices $\boldsymbol{X}(C_k)$ $(k = 1,2,\ldots,6)$ are positive semidefinite. Although this condition is not sufficient in general, it is a sufficient condition for $\boldsymbol{X} \in \mathbb{S}^n_+(E,?)$ when $G(N,E)$ is chordal. In this case, the number of the maximal cliques is bounded by the number of nodes of $G(N,E)$, *i.e.,* $p \leq n$. A graph is said chordal if every (simple) cycle of the graph with more than three edges has a chord. See [3] for basic properties on choral graphs.

Lemma 2.1. (Theorem 7 of Grone et al. [7]) *Let C_k $(k = 1,2,\ldots,p)$ be the maximal cliques of a chordal graph $G(N,E)$. Suppose that $\boldsymbol{X} \in \mathbb{S}^n(E,?)$. Then $\boldsymbol{X} \in \mathbb{S}^n_+(E,?)$ if and only if $\boldsymbol{X}(C_k) \in \mathbb{S}^{C_k}_+$ $(k = 1,2,\ldots,p)$.*

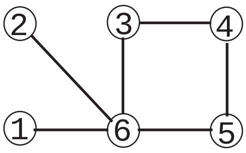

Figure 2: A graph $G(N, E)$ with $N = \{1, 2, 3, 4, 5, 6\}$

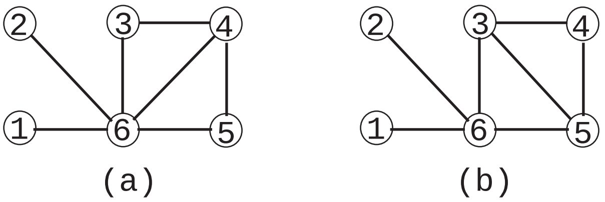

Figure 3: Chordal extensions of the graph $G(N, E)$ given in Figure 2. (a) The maximal cliques are $C_1 = \{3, 4, 6\}$, $C_2 = \{4, 5, 6\}$, $C_3 = \{1, 6\}$ and $C_4 = \{2, 6\}$. (b) The maximal cliques are $C_1 = \{3, 4, 5\}$, $C_2 = \{3, 5, 6\}$, $C_3 = \{1, 6\}$ and $C_4 = \{2, 6\}$.

Since the graph $G(N, E)$ in Figure 2 is not a chordal graph, we can not apply Lemma 2.1 to determine whether $\boldsymbol{X} \in \mathbb{S}^6(E, ?)$ of the form (5) belongs to $\mathbb{S}^6_+(E, ?)$. In such a case, we need to introduce a chordal extension of the graph $G(N, E)$ to use the lemma effectively. A graph $G(N, \overline{E})$ is a chordal extension of $G(N, E)$ if it is a chordal graph and $E \subseteq \overline{E}$. From the definition, Figure 3 shows two chordal extensions. If we choose the left graph as a chordal extension $G(N, \overline{E})$ of $G(N, E)$, the maximal cliques are $C_1 = \{3, 4, 6\}$, $C_2 = \{4, 5, 6\}$, $C_3 = \{1, 6\}$ and $C_4 = \{2, 6\}$, consequently, $\boldsymbol{X} \in \mathbb{S}^6_+(\overline{E}, ?)$ is characterized by $\boldsymbol{X}(C_k) \in \mathbb{S}^{C_k}_+$ $(k = 1, 2, 3, 4)$.

Remark 2.2. To compute the positive definite matrix completion of a matrix, we can recursively apply Lemma 2.6 of [6]. A numerical example is shown on page 657 of [6].

3 Exploiting the domain-space sparsity

In this section, we consider a general nonlinear optimization problem involving a matrix variable $\boldsymbol{X} \in \mathbb{S}^n$:

$$\text{minimize } f_0(\boldsymbol{x}, \boldsymbol{X}) \text{ subject to } \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{X}) \in \Omega \text{ and } \boldsymbol{X} \in \mathbb{S}^n_+, \tag{7}$$

where $f_0 : \mathbb{R}^s \times \mathbb{S}^n \to \mathbb{R}$, $\boldsymbol{f} : \mathbb{R}^s \times \mathbb{S}^n \to \mathbb{R}^m$ and $\Omega \subset \mathbb{R}^m$. Let E denote the set of distinct row and column index pairs (i, j) such that a value of X_{ij} is necessary to evaluate $f_0(\boldsymbol{x}, \boldsymbol{X})$ and/or $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{X})$. More precisely, E is a collection of index (i, j) such that even if $X^1_{k\ell} = X^2_{k\ell}$ for $\forall$ $(k, \ell) \neq (i, j)$, we have $f_0(\boldsymbol{x}, \boldsymbol{X}^1) \neq f_0(\boldsymbol{x}, \boldsymbol{X}^2)$ and/or $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{X}^1) \neq \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{X}^2)$ for some $\boldsymbol{x} \in \mathbb{R}^s$, $\boldsymbol{X}^1 \in \mathbb{S}^n$ and $\boldsymbol{X}^2 \in \mathbb{S}^n$. Consider a graph $G(N, E)$. We call E *the d-space sparsity pattern* and $G(N, E)$ *the d-space sparsity pattern graph.* If $G(N, \overline{E})$ is an extension of $G(N, E)$, then we may replace the condition $\boldsymbol{X} \in \mathbb{S}^n_+$ by $\boldsymbol{X} \in \mathbb{S}^n_+(\overline{E}, ?)$. To apply Lemma 2.1, we choose a chordal extension $G(N, \overline{E})$ of $G(N, E)$. Let $C_1, C_2, \ldots, C_p$ be its maximal cliques. Then we may regard f_0 and $\boldsymbol{f}$ as functions in $\boldsymbol{x} \in \mathbb{R}^s$ and $\boldsymbol{X}(C_k)$ $(k = 1, 2, \ldots, p)$, *i.e.,* there are functions $\tilde{f}_0$ and $\tilde{\boldsymbol{f}}$ in the variables $\boldsymbol{x}$ and $\boldsymbol{X}(C_k)$ $(k = 1, 2, \ldots, p)$ such that

$$\left.\begin{array}{l} f_0(\boldsymbol{x}, \boldsymbol{X}) = \tilde{f}_0(\boldsymbol{x}, \boldsymbol{X}(C_1), \boldsymbol{X}(C_2), \ldots, \boldsymbol{X}(C_p)) \text{ for every } (\boldsymbol{x}, \boldsymbol{X}) \in \mathbb{R}^s \times \mathbb{S}^n, \\ \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{X}) = \tilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{X}(C_1), \boldsymbol{X}(C_2), \ldots, \boldsymbol{X}(C_p)) \text{ for every } (\boldsymbol{x}, \boldsymbol{X}) \in \mathbb{R}^s \times \mathbb{S}^n. \end{array}\right\} \tag{8}$$

Therefore, the problem (7) is equivalent to

$$\begin{array}{ll} \text{minimize} & \tilde{f}_0(\boldsymbol{x}, \boldsymbol{X}(C_1), \boldsymbol{X}(C_2), \ldots, \boldsymbol{X}(C_p)) \\ \text{subject to} & \tilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{X}(C_1), \boldsymbol{X}(C_2), \ldots, \boldsymbol{X}(C_p)) \in \Omega \text{ and} \\ & \boldsymbol{X}(C_k) \in \mathbb{S}^{C_k}_+ \ (k = 1, 2, \ldots, p). \end{array} \tag{9}$$

As an illustrative example, we consider the problem whose d-space sparsity pattern graph $G(N, E)$ is shown in Figure 2:

$$\left.\begin{array}{ll} \text{minimize} & -\displaystyle\sum_{(i,j)\in E,\ i<j} X_{ij} \\ \text{subject to} & \displaystyle\sum_{i=1}^{6}(X_{ii} - \alpha_i)^2 \leq 6, \ \boldsymbol{X} \in \mathcal{S}^6_+, \end{array}\right\} \tag{10}$$

where $\alpha_i > 0$ $(i = 1, 2, \ldots, 6)$. As a chordal extension, we choose the graph $G(N, \overline{E})$ in (a) of Figure 3. Then, the problem (9) becomes

$$\left.\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{4} \tilde{f}_{0k}(\boldsymbol{X}(C_k)) \\ \text{subject to} & \displaystyle\sum_{k=1}^{4} \tilde{f}_k(\boldsymbol{X}(C_k)) \leq 6, \ \boldsymbol{X}(C_k) \in \mathbb{S}^{C_k}_+ \ (k = 1, 2, 3, 4), \end{array}\right\} \tag{11}$$

where

$$\left.\begin{array}{rcl} \tilde{f}_{01}(\boldsymbol{X}(C_1)) & = & -X_{34} - X_{36}, \ \tilde{f}_{02}(\boldsymbol{X}(C_2)) = -X_{45} - X_{56}, \\ \tilde{f}_{03}(\boldsymbol{X}(C_3)) & = & -X_{16}, \ \tilde{f}_{04}(\boldsymbol{X}(C_4)) = -X_{26}, \\ \tilde{f}_1(\boldsymbol{X}(C_1)) & = & (X_{33} - \alpha_3)^2 + (X_{44} - \alpha_4)^2 + (X_{66} - \alpha_6)^2, \\ \tilde{f}_2(\boldsymbol{X}(C_2)) & = & (X_{55} - \alpha_5)^2, \ \tilde{f}_3(\boldsymbol{X}(C_3)) = (X_{11} - \alpha_1)^2, \\ \tilde{f}_4(\boldsymbol{X}(C_4)) & = & (X_{22} - \alpha_2)^2. \end{array}\right\} \tag{12}$$

The positive semidefinite condition $\boldsymbol{X}(C_k) \in \mathbb{S}^{C_k}_+$ $(k = 1, 2, \ldots, p)$ in the problem (9) is not an ordinary positive semidefinite condition in the sense that overlapping variables X_{ij} $((i, j) \in C_k \cap C_\ell)$ exist in two distinct positive semidefinite constraints $\boldsymbol{X}(C_k) \in \mathbb{S}^{C_k}_+$ and $\boldsymbol{X}(C_\ell) \in \mathbb{S}^{C_\ell}_+$ if $C_k \cap C_\ell \neq \emptyset$. We describe two methods to transform the condition into an ordinary positive semidefinite condition. The first one was given in the papers [6, 20] where a d-space conversion method was proposed, and the second one was originally used for the sparse SDP relaxation of polynomial optimization problems [23, 24] and also in the paper [11] where a d-space conversion method was applied to an SDP relaxation of a sensor network localization problem. We call the first one *the d-space conversion method using clique trees* and the second one *the d-space conversion method using basis representation.*

3.1 The d-space conversion method using clique trees

We can replace $\boldsymbol{X}(C_k)$ $(k = 1, 2, \ldots, p)$ by p independent matrix variables $\boldsymbol{X}^k$ $(k = 1, 2, \ldots, p)$ if we add all equality constraints $X^k_{ij} = X^\ell_{ij}$ for every $(i, j) \in C_k \cap C_\ell$ with $i \leq j$ and every pair of C_k and C_ℓ such that $C_k \cap C_\ell \neq \emptyset$. For the chordal graph $G(N, \overline{E})$ given in (a) of Figure 3, those equalities turn out to be the 8 equalities

$$X^k_{66} - X^\ell_{66} = 0 \ (1 \leq k < \ell \leq 4), \ X^1_{44} = X^2_{44}, \ X^1_{46} = X^2_{46}$$

These equalities are linearly dependent, and we can choose a maximal number of linearly independent equalities that are equivalent to the original equalities. For example, either of a set of 5 equalities

$$X^1_{44} - X^2_{44} = 0, \ X^1_{46} - X^2_{46} = 0, \ X^1_{66} - X^2_{66} = 0, \ X^1_{66} - X^3_{66} = 0, \ X^1_{66} - X^4_{66} = 0 \tag{13}$$

or a set of 5 equalities

$$X^1_{44} - X^2_{44} = 0, \ X^1_{46} - X^2_{46} = 0, \ X^1_{66} - X^2_{66} = 0, \ X^2_{66} - X^3_{66} = 0, \ X^3_{66} - X^4_{66} = 0 \tag{14}$$

is equivalent to the set of 8 equalities above.

In general, we use a clique tree $\mathcal{T}(\mathcal{K}, \mathcal{E})$ with $\mathcal{K} = \{C_1, C_2, \ldots, C_p\}$ and $\mathcal{E} \subseteq \mathcal{K} \times \mathcal{K}$ to consistently choose a set of maximal number of linearly independent equalities. Here $\mathcal{T}(\mathcal{K}, \mathcal{E})$ is called a clique tree if it satisfies the *clique-intersection property*, that is, for each pair of nodes $C_k \in \mathcal{K}$ and $C_\ell \in \mathcal{K}$, the set $C_k \cap C_\ell$ is contained in every node on the (unique) path connecting C_k and C_ℓ. See [3] for basic properties on clique trees. We fix one clique for a root node of the tree $\mathcal{T}(\mathcal{K}, \mathcal{E})$, say C_1. For simplicity, we assume that the nodes $C_2, \ldots, C_p$ are indexed so that if a sequence of nodes $C_1, C_{\ell_2}, \ldots, C_{\ell_k}$ forms a path from the root node C_1 to a leaf node C_{ℓ_k}, then $1 < \ell_2 < \cdots < \ell_k$, and each edge is directed from the node with a smaller index to the other node with a larger index. Thus, the clique tree $\mathcal{T}(\mathcal{K}, \mathcal{E})$ is directed from the root node C_1 to its leaf nodes. Each edge (C_k, C_ℓ) of the clique tree $\mathcal{T}(\mathcal{K}, \mathcal{E})$ induces a set of equalities

$$X^k_{ij} - X^\ell_{ij} = 0 \ ((i, j) \in J(C_k \cap C_\ell)),$$

or equivalently,

$$\boldsymbol{E}_{ij} \bullet \boldsymbol{X}^k - \boldsymbol{E}_{ij} \bullet \boldsymbol{X}^\ell = 0 \ ((i, j) \in J(C_k \cap C_\ell)),$$

where $J(C) = \{(i, j) \in C \times C : i \leq j\}$ for every $C \subseteq N$. We add equalities of the form above for all $(C_k, C_\ell) \in \mathcal{E}$ when we replace $\boldsymbol{X}(C_k)$ $(k = 1, 2, \ldots, p)$ by p independent matrix variables $\boldsymbol{X}^k$ $(k = 1, 2, \ldots, p)$. We thus obtain a problem

$$\left.\begin{array}{ll} \text{minimize} & \tilde{f}_0(\boldsymbol{x}, \boldsymbol{X}^1, \boldsymbol{X}^2, \ldots, \boldsymbol{X}^p) \\ \text{subject to} & \tilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{X}^1, \boldsymbol{X}^2, \ldots, \boldsymbol{X}^p) \in \Omega, \\ & \boldsymbol{E}_{ij} \bullet \boldsymbol{X}^k - \boldsymbol{E}_{ij} \bullet \boldsymbol{X}^\ell = 0 \ ((i, j, k, \ell) \in \Lambda), \\ & \boldsymbol{X}^k \in \mathbb{S}^{C_k}_+ \ (k = 1, 2, \ldots, p), \end{array}\right\} \tag{15}$$

where

$$\Lambda = \{(g, h, k, \ell) : (g, h) \in J(C_k \cap C_\ell), \ (C_k, C_\ell) \in \mathcal{E}\}. \tag{16}$$

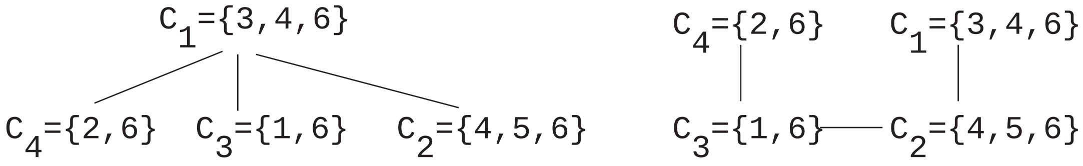

Figure 4: Two clique trees with $\mathcal{K} = \{C_1 = \{3,4,6\},\ C_2 = \{4,5,6\},\ C_3 = \{1,6\},\ C_4 = \{2,6\}\}$

This is equivalent to the problem (9). See Section 4 of [20] for more details.

Now we illustrate the conversion process above by the simple example (10). Figure 4 shows two clique trees for the graph given in (a) of Figure 3. The left clique tree in Figure 4 leads to the 5 equalities in (13), while the right clique tree in Figure 4 induces the 5 equalities in (14). In both cases, the problem (15) has the following form

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{4} \hat{f}_{0k}(\boldsymbol{X}^k) \\ \text{subject to} & \displaystyle\sum_{k=1}^{4} \hat{f}_k(\boldsymbol{X}^k) \leq 6, \\ & \text{the 5 equalities in (13) or (14),} \\ & \boldsymbol{X}^k \in \mathbb{S}_+^{C_k} \ (k = 1,2,3,4), \end{array}$$

where

$$\begin{array}{rcl} \hat{f}_{01}(\boldsymbol{X}^1) & = & -X^1_{34} - X^1_{36},\ \hat{f}_{02}(\boldsymbol{X}^2) = -X^2_{45} - X^2_{56}, \\ \hat{f}_{03}(\boldsymbol{X}^3) & = & -X^3_{16},\ \hat{f}_{04}(\boldsymbol{X}^4) = -X^4_{26}, \\ \hat{f}_1(\boldsymbol{X}^1) & = & (X^1_{33} - \alpha_3)^2 + (X^1_{44} - \alpha_4) + (X^1_{66} - \alpha_6)^2, \\ \hat{f}_2(\boldsymbol{X}^2) & = & (X^2_{55} - \alpha_5)^2,\ \hat{f}_3(\boldsymbol{X}^3) = (X^3_{11} - \alpha_1)^2,\ \hat{f}_4(\boldsymbol{X}^4) = (X^4_{22} - \alpha_2)^2. \end{array}$$

Remark 3.1. The d-space conversion method using clique trees can be implemented in many different ways. The fact that the chordal extension $G(N, \overline{E})$ of $G(N, E)$ is not unique offers flexibility in constructing an optimization problem of the form (15). More precisely, a choice of chordal extension $G(N, \overline{E})$ of $G(N, E)$ decides how “small” and “sparse” an optimization problem of the form (15) is, which is an important issue for solving the problem more efficiently. For the size of the problem (15), we need to consider the sizes of the matrix variables $\boldsymbol{X}^k$ $(k = 1, 2, \ldots, p)$ and the number of equalities in (15). Note that the sizes of the matrix variables $\boldsymbol{X}^k$ $(k = 1, 2, \ldots, p)$ are determined by the sizes of the maximal cliques C_k $(k = 1, 2, \ldots, p)$. This indicates that a chordal extension $G(N, \overline{E})$ with smaller maximal cliques C_k $(k = 1, 2, \ldots, p)$ may be better theoretically. (In computation, however, this is not necessarily true because of overhead of processing too many small positive semidefinite matrix variables.) The number of equalities in (15) or the cardinality of Λ is also determined by the chordal extension $G(N, \overline{E})$ of $G(N, E)$. Choosing a chordal extension $G(N, \overline{E})$ with smaller maximal cliques increases the number of equalities. Balancing these two contradicting targets, decreasing the sizes of the matrix variables and decreasing the number of equalities was studied in the paper [20] by combining some adjacent cliques along the clique tree $\mathcal{T}(\mathcal{K},\mathcal{E})$. See Section 4 of [20] for more details. In addition to the choice of a chordal extension $G(N,\overline{E})$ of $G(N,E)$, the representation of the functions and the choice of a clique tree add flexibilities in the construction of the problem (15). That is, the representation of the functions $f_0 : \mathbb{R}^s \times \mathbb{S}^n \to \mathbb{R}$ and $\boldsymbol{f} : \mathbb{R}^s \times \mathbb{S}^n \to \mathbb{R}^m$ in the vector variable $\boldsymbol{x}$ and the matrix variables $\boldsymbol{X}(C_k)$ $(k = 1, 2, \ldots, p)$ as in (8); for example, we could move the term $(X_{66} - \alpha_6)^2$ from $\tilde{f}_1(\boldsymbol{x}, \boldsymbol{X}(C_1))$ to either of $\tilde{f}_k(\boldsymbol{x}, \boldsymbol{X}(C_k))$ $(k = 2, 3, 4)$. These choices of the functions f_0, $\boldsymbol{f}$ and a clique tree affect the sparse structure of the resulting problem (15), which is also important for efficient computation.

Remark 3.2. Software packages [2, 10] are available to generate a chordal extension $G(N,\overline{E})$ of a graph $G(N,E)$ and the maximal cliques of $G(N,\overline{E})$. Alternatively, a chordal extension and its maximal cliques can be obtained by the Matlab functions symamd.m (a symmetric approximate minimum degree ordering) and chol.m (the Cholesky factorization). We used the Matlab functions in [11, 24] and the numerical experiments in Section 7.

3.2 The domain-space conversion method using basis representation

Define

$$\begin{aligned}
\bar{J} &= \bigcup_{k=1}^{p} J(C_k), \\
(X_{ij} : (i,j) \in \bar{J}) &= \text{the vector variable consisting of } X_{ij} \ ((i,j) \in \bar{J}), \\
\bar{f}_0(\boldsymbol{x}, (X_{ij} : (i,j) \in \bar{J})) &= f_0(\boldsymbol{x}, \boldsymbol{X}) \text{ for every } (\boldsymbol{x}, \boldsymbol{X}) \in \mathbb{R}^s \times \mathbb{S}^n, \\
\bar{\boldsymbol{f}}(\boldsymbol{x}, (X_{ij} : (i,j) \in \bar{J})) &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{X}) \text{ for every } (\boldsymbol{x}, \boldsymbol{X}) \in \mathbb{R}^s \times \mathbb{S}^n.
\end{aligned}$$

We represent each $\boldsymbol{X}(C_k)$ in terms of a linear combination of the basis $\boldsymbol{E}_{ij}$ $((i,j) \in J(C_k))$ of the space $\mathbb{S}^{C_k}$ as in (6) with $C = C_k$ $(k = 1, 2, \ldots, p)$. Substituting this basis representation into the problem (9), we obtain

$$\left.\begin{array}{ll}
\text{minimize} & \bar{f}_0(\boldsymbol{x}, (X_{ij} : (i,j) \in \bar{J})) \\
\text{subject to} & \bar{\boldsymbol{f}}(\boldsymbol{x}, (X_{ij} : (i,j) \in \bar{J}) \in \Omega, \\
& \displaystyle\sum_{(i,j) \in J(C_k)} \boldsymbol{E}_{ij} X_{ij} \in \mathbb{S}^{C_k}_{+} \ (k = 1, 2, \ldots, p).
\end{array}\right\} \tag{17}$$

We observe that the illustrative example (10) is converted into the problem

$$\left.\begin{array}{ll}
\text{minimize} & \displaystyle -\sum_{(i,j) \in E, \ i<j} X_{ij} \\
\text{subject to} & \displaystyle\sum_{i=1}^{6} (X_{ii} - \alpha_i)^2 \leq 6, \\
& \displaystyle\sum_{(i,j) \in J(C_k)} \boldsymbol{E}_{ij} X_{ij} \in \mathbb{S}^{C_k}_{+} \ (k = 1, 2, 3, 4).
\end{array}\right\} \tag{18}$$

Remark 3.3. Compared to the d-space conversion method using clique trees, the d-space conversion method using basis representation described above provides limited flexibilities.

To make the size of the problem (17) smaller, we need to select a chordal extension $G(N, \overline{E})$ of $G(N, E)$ with smaller maximal cliques C_k $(k = 1, 2, \ldots, p)$. As a result, the sizes of semidefinite constraints become smaller. As we mentioned in Remark 3.1, however, too many smaller positive semidefinite matrix variables may yield heavy overhead in computation.

4 Duality in positive semidefinite matrix completion

Throughout this section, we assume that $G(N, E)$ denotes a chordal graph. In Lemma 2.1, we have described a necessary and sufficient condition for a partial symmetric matrix $\boldsymbol{X} \in \mathbb{S}^n(E, ?)$ to be completed to a positive semidefinite symmetric matrix. Let

$$\begin{aligned} \mathbb{S}^n(E,0) &= \{\boldsymbol{A} \in \mathbb{S}^n : A_{ij} = 0 \text{ if } (i,j) \notin E^{\bullet}\}, \\ \mathbb{S}^n_+(E,0) &= \{\boldsymbol{A} \in \mathbb{S}^n(E,0) : \boldsymbol{A} \succeq \boldsymbol{O}\}. \end{aligned}$$

In this section, we derive a necessary and sufficient condition for a symmetric matrix $\boldsymbol{A} \in \mathbb{S}^n(E, 0)$ to be positive semidefinite, *i.e.*, $\boldsymbol{A} \in \mathbb{S}^n_+(E, 0)$. This condition is used for the range-space conversion methods in Section 5. We note that for $\boldsymbol{A} \in \mathbb{S}^n(E, 0)$, these two issues have primal-dual relationship:

$$\boldsymbol{A} \in \mathbb{S}^n_+(E,0) \text{ if and only if } \sum_{(i,j)\in E^{\bullet}} A_{ij}X_{ij} \geq 0 \text{ for every } \boldsymbol{X} \in \mathbb{S}^n_+(E,?). \tag{19}$$

Suppose $\boldsymbol{A} \in \mathbb{S}^n(E, 0)$. Let $C_1, C_2, \ldots, C_p$ be the maximal cliques of $G(N, E)$. Then, we can consistently decompose $\boldsymbol{A} \in \mathbb{S}^n(E, 0)$ into $\widetilde{\boldsymbol{A}}^k \in \mathbb{S}^{C_k}$ $(k = 1, 2, \ldots, p)$ such that $\boldsymbol{A} = \sum_{k=1}^{p} \widetilde{\boldsymbol{A}}^k$. We know that $\boldsymbol{A}$ is positive semidefinite if and only if $\boldsymbol{A} \bullet \boldsymbol{X} \geq 0$ for every $\boldsymbol{X} \in \mathbb{S}^n_+$. This relation and Lemma 2.1 are used in the following.

Since $\boldsymbol{A} \in \mathbb{S}^n(E, 0)$, the condition $\boldsymbol{A} \succeq \boldsymbol{O}$ can be relaxed to the condition on the right-hand side of (19). Therefore, $\boldsymbol{A}$ is positive semidefinite if and only if the following SDP has the optimal value 0.

$$\text{minimize} \quad \sum_{(i,j)\in E^{\bullet}} \left[\sum_{k=1}^{p} \widetilde{\boldsymbol{A}}^k\right]_{ij} X_{ij} \quad \text{subject to } \boldsymbol{X} \in \mathbb{S}^n_+(E,?). \tag{20}$$

We can rewrite the objective function as

$$\begin{aligned} \sum_{(i,j)\in E^{\bullet}} \left[\sum_{k=1}^{p} \widetilde{\boldsymbol{A}}^k\right]_{ij} X_{ij} &= \sum_{k=1}^{p} \left[\sum_{(i,j)\in E^{\bullet}} \widetilde{A}^k_{ij} X_{ij}\right] \\ &= \sum_{k=1}^{p} \left(\widetilde{\boldsymbol{A}}^k \bullet \boldsymbol{X}(C_k)\right) \text{ for every } \boldsymbol{X} \in \mathbb{S}^n_+(E,?). \end{aligned}$$

Note that the second equality follows from $\widetilde{\boldsymbol{A}}^k \in \mathbb{S}^{C_k}$ $(k = 1, 2, \ldots, p)$. Applying Lemma 2.1 to the constraint $\boldsymbol{X} \in \mathbb{S}^n_+(E, ?)$ of the SDP (20), we obtain an SDP

$$\text{minimize} \quad \sum_{k=1}^{p} \left(\widetilde{\boldsymbol{A}}^k \bullet \boldsymbol{X}(C_k)\right) \quad \text{subject to } \boldsymbol{X}(C_k) \in \mathbb{S}^{C_k}_+ \ (k = 1, 2, \ldots, p), \tag{21}$$

which is equivalent to the SDP (20).

The SDP (21) involves multiple positive semidefinite matrix variables with overlapping elements. We have described two methods to convert such multiple matrix variables into independent ones with no overlapping elements in Sections 3.1 and 3.2, respectively. We apply the method given in Section 3.1 to the SDP (21). Let $\mathcal{T}(\mathcal{K},\mathcal{E})$ be a clique tree with $\mathcal{K} = \{C_1, C_2, \ldots, C_p\}$ and $\mathcal{E} \subseteq \mathcal{K} \times \mathcal{K}$. Then, we obtain an SDP

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{p} \left(\widetilde{\boldsymbol{A}}^k \bullet \boldsymbol{X}^k \right) \\ \text{subject to} & \boldsymbol{E}_{ij} \bullet \boldsymbol{X}^k - \boldsymbol{E}_{ij} \bullet \boldsymbol{X}^\ell = 0 \ ((i,j,k,\ell) \in \Lambda), \\ & \boldsymbol{X}^k \in \mathbb{S}^{C_k}_+ \ (k = 1,2,\ldots,p), \end{array} \tag{22}$$

which is equivalent to the SDP (21). Here Λ is given in (16).

Theorem 4.1. $\boldsymbol{A} \in \mathbb{S}^n(E,0)$ *is positive semidefinite if and only if the system of LMIs*

$$\widetilde{\boldsymbol{A}}^k - \widetilde{\boldsymbol{L}}^k(\boldsymbol{z}) \succeq \boldsymbol{O} \quad (k = 1,2,\ldots,p). \tag{23}$$

has a solution $\boldsymbol{v} = (v_{ghk\ell} : (g,h,k,\ell) \in \Lambda)$*. Here* $\boldsymbol{z} = (z_{ghk\ell} : (g,h,k,\ell) \in \Lambda)$ *denotes a vector variable consisting of* $z_{ghk\ell}$ $((g,h,k,\ell) \in \Lambda)$*, and*

$$\begin{array}{rcl} \widetilde{\boldsymbol{L}}^k(\boldsymbol{z}) & = & -\sum_{(i,j,h);\ (i,j,h,k) \in \Lambda} \boldsymbol{E}_{ij} z_{ijhk} + \sum_{(i,j,\ell);\ (i,j,k,\ell) \in \Lambda} \boldsymbol{E}_{ij} z_{ijk\ell} \\ & & \text{for every } \boldsymbol{z} = (z_{ijk\ell} : (i,j,k,\ell) \in \Lambda) \ (k = 1,2,\ldots,p). \end{array} \tag{24}$$

Proof: In the previous discussions, we have shown that $\boldsymbol{A} \in \mathbb{S}^n(E,0)$ is positive semidefinite if and only if the SDP (22) has the optimal value 0. The dual of the SDP (22) is

$$\text{maximize } 0 \text{ subject to (23).} \tag{25}$$

The primal SDP (22) attains the objective value 0 at a trivial feasible solution $(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_p) = (\boldsymbol{O}, \boldsymbol{O}, \ldots, \boldsymbol{O})$. If the dual SDP (25) is feasible or the system of LMIs (23) has a solution, then the primal SDP (22) has the optimal value 0 by the week duality theorem. Thus we have shown the "if part" of the theorem. Now suppose that the primal SDP (22) has the optimal value 0. The primal SDP (22) has an interior-feasible solution; for example, take $\boldsymbol{X}^k$ to be the $\#C_k \times \#C_k$ identity matrix in $\mathbb{S}^{C_k}$ $(k = 1,2,\ldots,p)$. By the strong duality theorem (Theorem 4.2.1 of [21]), the optimal value of the dual SDP (25) is zero, which implies that (25) is feasible. ∎

As a corollary, we obtain the following.

Theorem 4.2. (Theorem 2.3 of [1]) $\boldsymbol{A} \in \mathbb{S}^n(E,0)$ *is positive semidefinite if and only if there exist* $\boldsymbol{Y}^k \in \mathbb{S}^{C_k}_+$ $(k = 1,2,\ldots,p)$ *which decompose* $\boldsymbol{A}$ *as* $\boldsymbol{A} = \sum_{k=1}^{p} \boldsymbol{Y}^k$.

Proof: Since the "if part" is straightforward, we prove the "only if" part. Assume that $\boldsymbol{A}$ is positive semidefinite. By Theorem 4.1, the LMI (23) has a solution $\tilde{\boldsymbol{z}}$. Let

$\boldsymbol{Y}^k = \widetilde{\boldsymbol{A}}^k - \widetilde{\boldsymbol{L}}^k(\tilde{\boldsymbol{z}})$ $(k = 1, 2, \ldots, p)$. Then $\boldsymbol{Y}^k \in \mathbb{S}^{C_k}_+$ $(k = 1, 2, \ldots, p)$. Since $\displaystyle\sum_{k=1}^{p} \widetilde{\boldsymbol{L}}^k(\tilde{\boldsymbol{z}}) = \boldsymbol{O}$ by construction, we obtain the desired result. ▮

Conversely, Theorem 4.1 can be derived from Theorem 4.2. In the paper [1], Theorem 4.2 was proved by Theorem 7 of Grone et al. [7] (Lemma 2.1 of the this paper).

We conclude this section by applying Theorem 4.1 to the case of the chordal graph $G(N, E)$ given in (a) of Figure 3. The maximal cliques are $C_1 = \{3, 4, 6\}$, $C_2 = \{4, 5, 6\}$, $C_3 = \{1, 6\}$ and $C_4 = \{2, 6\}$, so that $\boldsymbol{A} \in \mathbb{S}^6(E, 0)$ is decomposed into 4 matrices

$$
\begin{aligned}
\widetilde{\boldsymbol{A}}^1 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{33} & A_{34} & 0 & A_{36} \\ 0 & 0 & A_{43} & A_{44} & 0 & A_{46} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{63} & A_{64} & 0 & A_{66} \end{pmatrix} \in \mathbb{S}^{\{3,4,6\}}, \\
\widetilde{\boldsymbol{A}}^2 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{45} & 0 \\ 0 & 0 & 0 & A_{54} & A_{55} & A_{56} \\ 0 & 0 & 0 & 0 & A_{65} & 0 \end{pmatrix} \in \mathbb{S}^{\{4,5,6\}}, \\
\widetilde{\boldsymbol{A}}^3 &= \begin{pmatrix} A_{11} & 0 & 0 & 0 & 0 & A_{16} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ A_{61} & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{S}^{\{1,6\}}, \\
\widetilde{\boldsymbol{A}}^4 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{22} & 0 & 0 & 0 & A_{26} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{62} & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{S}^{\{2,6\}},
\end{aligned}
$$

or,

$$
\left.
\begin{aligned}
\widetilde{\boldsymbol{A}}^1 &= \begin{pmatrix} A_{33} & A_{34} & A_{36} \\ A_{43} & A_{44} & A_{46} \\ A_{63} & A_{64} & A_{66} \end{pmatrix} \in \mathbb{S}^{\{3,4,6\}},\ \widetilde{\boldsymbol{A}}^2 = \begin{pmatrix} 0 & A_{45} & 0 \\ A_{54} & A_{55} & A_{56} \\ 0 & A_{65} & 0 \end{pmatrix} \in \mathbb{S}^{\{4,5,6\}}, \\
\widetilde{\boldsymbol{A}}^3 &= \begin{pmatrix} A_{11} & A_{16} \\ A_{61} & 0 \end{pmatrix} \in \mathbb{S}^{\{1,6\}},\ \widetilde{\boldsymbol{A}}^4 = \begin{pmatrix} A_{22} & A_{26} \\ A_{62} & 0 \end{pmatrix} \in \mathbb{S}^{\{2,6\}}
\end{aligned}
\right\} \tag{26}
$$

in the compact representation. We note that this decomposition is not unique. For example, we can move the $(6, 6)$ element A_{66} from $\widetilde{\boldsymbol{A}}^1$ to any other $\widetilde{\boldsymbol{A}}^k$. We showed two clique trees with $\mathcal{K} = \{C_1, C_2, C_3, C_4\}$ in Figure 4. For the left clique tree, we have

$\Lambda = \{(4,4,1,2),(4,6,1,2),(6,6,1,2),(6,6,1,3),(6,6,1,4))\}$. Thus, the system of LMIs (23) becomes

$$\left.\begin{array}{l} \begin{pmatrix} A_{33} & A_{34} & A_{36} \\ A_{43} & A_{44}-z_{4412} & A_{46}-z_{4612} \\ A_{63} & A_{64}-z_{4612} & A_{66}-z_{6612}-z_{6613}-z_{6614} \end{pmatrix} \succeq \boldsymbol{O}, \\ \begin{pmatrix} z_{4412} & A_{45} & z_{4612} \\ A_{54} & A_{55} & A_{56} \\ z_{4612} & A_{65} & z_{6612} \end{pmatrix} \succeq \boldsymbol{O}, \\ \begin{pmatrix} A_{11} & A_{16} \\ A_{61} & z_{6613} \end{pmatrix} \succeq \boldsymbol{O}, \ \begin{pmatrix} A_{22} & A_{26} \\ A_{62} & z_{6614} \end{pmatrix} \succeq \boldsymbol{O}. \end{array}\right\} \tag{27}$$

For the right clique tree, we have $\Lambda = \{(4,4,1,2),(4,6,1,2),(6,6,1,2),(6,6,2,3),(6,6,3,4)\}$ and

$$\left.\begin{array}{l} \begin{pmatrix} A_{33} & A_{34} & A_{36} \\ A_{43} & A_{44}-z_{4412} & A_{46}-z_{4612} \\ A_{63} & A_{64}-z_{4612} & A_{66}-z_{6612} \end{pmatrix} \succeq \boldsymbol{O}, \\ \begin{pmatrix} z_{4412} & A_{45} & z_{4612} \\ A_{54} & A_{55} & A_{56} \\ z_{4612} & A_{65} & z_{6612}-z_{6623} \end{pmatrix} \succeq \boldsymbol{O}, \\ \begin{pmatrix} A_{11} & A_{16} \\ A_{61} & z_{6623}-z_{6634} \end{pmatrix} \succeq \boldsymbol{O}, \ \begin{pmatrix} A_{22} & A_{26} \\ A_{62} & z_{6634} \end{pmatrix} \succeq \boldsymbol{O}. \end{array}\right\} \tag{28}$$

5 Exploiting the range-space sparsity

In this section, we present two range-space conversion methods, *the r-space conversion method using clique trees* based on Theorem 4.1 and *the r-space conversion method using matrix decomposition* based on Theorem 4.2.

5.1 The range-space conversion method using clique trees

Let

$$F \;=\; \{(i,j) \in N \times N : M_{ij}(\boldsymbol{y}) \neq 0 \text{ for some } \boldsymbol{y} \in \mathbb{R}^s,\ i \neq j\}.$$

We call F *the r-space sparsity pattern* and $G(N,F)$ *the r-space sparsity pattern graph* of the mapping $\boldsymbol{M} : \mathbb{R}^s \to \mathbb{S}^n$. Apparently, $\boldsymbol{M}(\boldsymbol{y}) \in \mathbb{S}^n(F,0)$ for every $\boldsymbol{y} \in \mathbb{R}^s$, but the graph $G(N,F)$ may not be chordal. Let $G(N,E)$ be a chordal extension of $G(N,F)$. Then

$$\boldsymbol{M}(\boldsymbol{y}) \in \mathbb{S}^n(E,0) \text{ for every } \boldsymbol{y} \in \mathbb{R}^s. \tag{29}$$

Let $C_1, C_2, \ldots, C_p$ be the maximal cliques of $G(N,E)$.

To apply Theorem 4.1, we choose mappings $\widetilde{\boldsymbol{M}}^k$ $(k = 1,2,\ldots,p)$ to decompose the mapping $\boldsymbol{M} : \mathbb{R}^s \to \mathbb{S}^n$ such that

$$\boldsymbol{M}(\boldsymbol{y}) \;=\; \sum_{k=1}^{p} \widetilde{\boldsymbol{M}}^k(\boldsymbol{y}) \text{ for every } \boldsymbol{y} \in \mathbb{R}^s,\ \widetilde{\boldsymbol{M}}^k : \mathbb{R}^s \to \mathbb{S}^{C_k}\ (k = 1,2,\ldots,p). \tag{30}$$

Let $\mathcal{T}(\mathcal{K},\mathcal{E})$ be a clique tree where $\mathcal{K} = \{C_1, C_2, \ldots, C_p\}$ and $\mathcal{E} \subset \mathcal{K} \times \mathcal{K}$. By Theorem 4.1, $\boldsymbol{y}$ is a solution of (1) if and only if it is a solution of

$$\widetilde{\boldsymbol{M}}^k(\boldsymbol{y}) - \widetilde{\boldsymbol{L}}^k(\boldsymbol{z}) \succeq \boldsymbol{O} \ (k = 1, 2, \ldots, p) \tag{31}$$

for some $\boldsymbol{z} = (z_{ghk\ell} : (g, h, k, \ell) \in \Lambda)$, where Λ is given in (16) and $\widetilde{\boldsymbol{L}}^k$ in (24).

We may regard the r-space conversion method using clique trees described above as a dual of the d-space conversion method using clique trees applied to the SDP

$$\text{minimize } \boldsymbol{M}(\boldsymbol{y}) \bullet \boldsymbol{X} \text{ subject to } \boldsymbol{X} \succeq \boldsymbol{O}, \tag{32}$$

where $\boldsymbol{X} \in \mathbb{S}^n$ denotes a variable matrix and $\boldsymbol{y} \in \mathbb{R}^s$ a fixed vector. We know that $\boldsymbol{M}(\boldsymbol{y}) \succeq \boldsymbol{O}$ if and only if the optimal value of the SDP (32) is zero, so that (32) serves as a dual of the matrix inequality $\boldsymbol{M}(\boldsymbol{y}) \succeq \boldsymbol{O}$. Each element $z_{ijk\ell}$ of the vector variable $\boldsymbol{z}$ corresponds to a dual variable of the equality constraint $\boldsymbol{E}_{ij} \bullet \boldsymbol{X}^k - \boldsymbol{E}_{ij} \bullet \boldsymbol{X}^\ell = 0$ in the problem (15), while each matrix variable $\boldsymbol{X}^k \in \mathbb{S}^{C_k}$ in the problem (15) corresponds to a dual matrix variable of the kth matrix inequality $\widetilde{\boldsymbol{M}}^k(\boldsymbol{y}) - \widetilde{\boldsymbol{L}}^k(\boldsymbol{z}) \succeq \boldsymbol{O}$.

Remark 5.1. On the flexibilities in implementing the r-space conversion method using clique trees, the comments in Remark 3.1 are valid if we replace the sizes of the matrix variable $\boldsymbol{X}^k$ by the size of the mapping $\widetilde{\boldsymbol{M}}^k : \mathbb{R}^s \to \mathbb{S}^{C_k}$ and the number of equalities by the number of elements $z_{ijk\ell}$ of the vector variable $\boldsymbol{z}$. The correlative sparsity of (31) depends on the choice of the clique tree and the decomposition (30). This is illustrated in Remark 6.1.

As an example, we consider the case where $\boldsymbol{M}$ is tridiagonal, *i.e.*, the (i,j)th element M_{ij} of $\boldsymbol{M}$ is zero if $|i-j| \geq 2$, to illustrate the range space conversion of the matrix inequality (1) into the system of matrix inequalities (31). By letting $E = \{(i,j) : |i-j| = 1\}$, we have a simple chordal graph $G(N,E)$ with no cycle satisfying (29), its maximal cliques $C_k = \{k, k+1\}$ $(k = 1, 2, \ldots, n-1)$, and a clique tree $\mathcal{T}(\mathcal{K},\mathcal{E})$ with

$$\mathcal{K} = \{C_1, C_2, \ldots, C_{n-1}\} \text{ and } \mathcal{E} = \{(C_k, C_{k+1}) \in \mathcal{K} \times \mathcal{K} : k = 1, 2, \ldots, n-2\}.$$

For every $\boldsymbol{y} \in \mathbb{R}^s$, let

$$\widetilde{\boldsymbol{M}}^k(\boldsymbol{y}) = \begin{cases} \begin{pmatrix} M_{kk}(\boldsymbol{y}) & M_{k,k+1}(\boldsymbol{y}) \\ M_{k+1,k}(\boldsymbol{y}) & 0 \end{pmatrix} \in \mathbb{S}^{C_k} & \text{if } 1 \leq k \leq n-2, \\[2ex] \begin{pmatrix} M_{n-1,n-1}(\boldsymbol{y}) & M_{n-1,n}(\boldsymbol{y}) \\ M_{n,n-1}(\boldsymbol{y}) & M_{nn}(\boldsymbol{y}) \end{pmatrix} \in \mathbb{S}^{C_k} & \text{if } k = n-1. \end{cases}$$

Then, we can decompose $\boldsymbol{M} : \mathbb{R}^s \to \mathbb{S}^n(E,0)$ into $\widetilde{\boldsymbol{M}}^k : \mathbb{R}^s \to \mathbb{S}^{C_k}$ $(k = 1, 2, \ldots, n-1)$ as in (30) with $p = n-1$. We also see that

$$\begin{aligned} \Lambda &= \{(k+1, k+1, k, k+1) : k = 1, 2, \ldots, n-2\}, \\ \widetilde{\boldsymbol{L}}^k(\boldsymbol{z}) &= \begin{cases} \boldsymbol{E}_{22}\ z_{2212} \in \mathbb{S}^{C_1} & \text{if } k = 1, \\ -\boldsymbol{E}_{k,k}\ z_{k,k,k-1,k} + \boldsymbol{E}_{k+1,k+1}\ z_{k+1,k+1,k,k+1} \in \mathbb{S}^{C_k} & \text{if } k = 2, 3, \ldots, n-2, \\ -\boldsymbol{E}_{n-1,n-1}\ z_{n-1,n-1,n-2,n-1} \in \mathbb{S}^{C_{n-1}} & \text{if } k = n-1, \end{cases} \end{aligned}$$

Thus the resulting system of matrix inequalities (31) is

$$\left.\begin{array}{l}
\begin{pmatrix} M_{11}(\boldsymbol{y}) & M_{12}(\boldsymbol{y}) \\ M_{21}(\boldsymbol{y}) & -z_{2212} \end{pmatrix} \succeq \boldsymbol{O}, \\
\begin{pmatrix} M_{kk}(\boldsymbol{y}) + z_{k,k,k-1,k} & M_{k,k+1}(\boldsymbol{y}) \\ M_{k+1,k}(\boldsymbol{y}) & -z_{k+1,k+1,k,k+1} \end{pmatrix} \succeq \boldsymbol{O} \ (k = 2, 3, \ldots, n-2), \\
\begin{pmatrix} M_{n-1,n-1}(\boldsymbol{y}) + z_{n-1,n-1,n-2,n-1} & M_{n-1,n}(\boldsymbol{y}) \\ M_{n,n-1}(\boldsymbol{y}) & M_{nn}(\boldsymbol{y}) \end{pmatrix} \succeq \boldsymbol{O}.
\end{array}\right\}$$

5.2 The range-space conversion method using matrix decomposition

By Theorem 4.2, we obtain that $\boldsymbol{y} \in \mathbb{R}^s$ is a solution of the matrix inequality (1) if and only if there exist $\boldsymbol{Y}^k \in \mathbb{S}^{C_k}$ $(k = 1, 2, \ldots, p)$ such that

$$\sum_{k=1}^{p} \boldsymbol{Y}^k = \boldsymbol{M}(\boldsymbol{y}) \ \text{ and } \boldsymbol{Y}^k \in \mathbb{S}_+^{C_k} \ (k = 1, 2, \ldots, p).$$

Let $\overline{J} = \cup_{k=1}^{p} J(C_k)$ and $\Gamma(i,j) = \{k : i \in C_k, \ j \in C_k\}$ $((i,j) \in \overline{J})$. Then we can rewrite the condition above as

$$\sum_{k\in\Gamma(i,j)} \boldsymbol{E}_{ij} \bullet \boldsymbol{Y}^k - \boldsymbol{E}_{ij} \bullet \boldsymbol{M}(\boldsymbol{y}) = 0 \ ((i,j) \in \overline{J}) \ \text{ and } \boldsymbol{Y}^k \in \mathbb{S}_+^{C_k} \ (k = 1, 2, \ldots, p). \qquad (33)$$

We may regard the r-space conversion method using matrix decomposition as a dual of the d-space conversion method using basis representation applied to the SDP (32) with a fixed $\boldsymbol{y} \in \mathbb{R}^s$. Each variable X_{ij} $((i,j) \in \overline{J})$ in the problem (17) corresponds to a dual real variable of the (i,j)th equality constraint of the problem (33), while each matrix variable $\boldsymbol{Y}^k$ in the problem (33) corresponds to a dual matrix variable of the constraint $\displaystyle\sum_{(i,j)\in J(C_k)} \boldsymbol{E}_{ij} X_{ij} \in \mathbb{S}_+^{C_k}$.

Remark 5.2. On the flexibilities in implementing the r-space conversion method using matrix decomposition, the comments in Remark 3.3 are valid if we replace the sizes of the semidefinite constraints by the sizes of the matrix variables $\boldsymbol{Y}^k$ $(k = 1, 2, \ldots, p)$.

We illustrate the r-space conversion method using matrix decomposition with the same example where $\boldsymbol{M}$ is tridiagonal as in Section 5.1. In this case, we see that

$$\begin{array}{rcl}
p & = & n-1, \\
C_k & = & \{k, k+1\} \ (k = 1, 2, \ldots, n-1), \\
J(C_k) & = & \{(k,k), \ (k,k+1), \ (k+1,k+1)\} \ (k = 1, 2, \ldots, n-1), \\
\overline{J} & = & \{(k,k) : k = 1, 2, \ldots, n\} \bigcup \{(k,k+1) : k = 1, 2, \ldots, n-1\}, \\
\Gamma(i,j) & = & \left\{\begin{array}{ll} \{1\} & \text{if } i = j = 1, \\ \{k\} & \text{if } i = k, \ j = k+1 \text{ and } 1 \leq k \leq n-1, \\ \{k-1, k\} & \text{if } i = j = k \text{ and } 2 \leq k \leq n-1, \\ \{n-1\} & \text{if } i = j = n. \end{array}\right.
\end{array}$$

Hence, the matrix inequality (1) with the tridiagonal $\boldsymbol{M} : \mathbb{R}^s \to \mathbb{S}^n$ is converted into

$$\begin{array}{l}
\boldsymbol{E}_{11} \bullet \boldsymbol{Y}^1 - \boldsymbol{E}_{11} \bullet \boldsymbol{M}(\boldsymbol{y}) = 0, \\
\boldsymbol{E}_{k,k+1} \bullet \boldsymbol{Y}^k - \boldsymbol{E}_{k,k+1} \bullet \boldsymbol{M}(\boldsymbol{y}) = 0 \ (k = 1, 2, \ldots, n-1), \\
\boldsymbol{E}_{kk} \bullet \boldsymbol{Y}^{k-1} + \boldsymbol{E}_{kk} \bullet \boldsymbol{Y}^k - \boldsymbol{E}_{kk} \bullet \boldsymbol{M}(\boldsymbol{y}) = 0 \ (k = 2, \ldots, n-1), \\
\boldsymbol{E}_{nn} \bullet \boldsymbol{Y}^{n-1} - \boldsymbol{E}_{nn} \bullet \boldsymbol{M}(\boldsymbol{y}) = 0, \\
\boldsymbol{Y}^k = \begin{pmatrix} Y^k_{kk} & Y^k_{k,k+1} \\ Y^k_{k+1,k} & Y^k_{k+1,k+1} \end{pmatrix} \in \mathbb{S}^{C_k}_+ \ \ (k = 1, 2, \ldots, n-1).
\end{array}$$

6 Correlative sparsity

When we are concerned with the SDP relaxation of polynomial SDPs (including ordinary polynomial optimization problems) and linear SDPs, another type of sparsity called the correlative sparsity plays an important role in solving the SDPs efficiently. The correlative sparsity was dealt with extensively in the paper [13]. It is known that the sparse SDP relaxation [16, 18, 23, 24] for a correlatively sparse polynomial optimization problem leads to an SDP that can maintain the sparsity for primal-dual interior-point methods. See Section 6 of [13]. In this section, we focus on how the d-space and r-space conversion methods enhance the correlative sparsity. We consider a polynomial SDP of the form

$$\text{maximize} \quad f_0(\boldsymbol{y}) \quad \text{subject to} \quad \boldsymbol{F}_k(\boldsymbol{y}) \in \mathbb{S}^{m_k}_+ \ (k = 1, 2, \ldots, p). \tag{34}$$

Here f_0 denotes a real valued polynomial function in $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{F}_k$ a mapping from $\mathbb{R}^n$ into $\mathbb{S}^{m_k}$ with

all polynomial components in $\boldsymbol{y} \in \mathbb{R}^n$. For simplicity, we assume that f_0 is a linear function of the form $f_0(\boldsymbol{y}) = \boldsymbol{b}^T\boldsymbol{y}$ for some $\boldsymbol{b} \in \mathbb{R}^n$. The correlative sparsity pattern graph is defined as a graph $G(N, E)$ with the node set $N = \{1, 2, \ldots, n\}$ and the edge set

$$E = \left\{ (i,j) \in N \times N : \begin{array}{l} i \neq j, \text{ both values } y_i \text{ and } y_j \text{ are necessary} \\ \text{to evaluate the value of } \boldsymbol{F}_k(\boldsymbol{y}) \text{ for some } k \end{array} \right\}.$$

When a chordal extension $G(N, \overline{E})$ of the correlative sparsity pattern graph $G(N, E)$ is sparse or all the maximal cliques of $G(N, \overline{E})$ are small-sized, we can effectively apply the sparse SDP relaxation [16, 18, 23, 24] to the polynomial SDP (34). As a result, we have a linear SDP satisfying a correlative sparsity characterized by the same chordal graph structure as $G(N, \overline{E})$. More details can be found in Section 6 of [13]. Even when the correlative sparsity pattern graph $G(N, E)$ or its chordal extension $G(N, \overline{E})$ is not sparse, the polynomial SDP may have “a hidden correlative sparsity” that can be recognized by applying the d-space and/or r-space conversion methods to the problem to decompose a large size matrix variable (and/or inequality) into multiple smaller size matrix variables (and/or inequalities). To illustrate this, let us consider a polynomial SDP of the form

$$\text{minimize } \boldsymbol{b}^T\boldsymbol{y} \text{ subject to } \boldsymbol{F}(\boldsymbol{y}) \in \mathbb{S}^n_+,$$

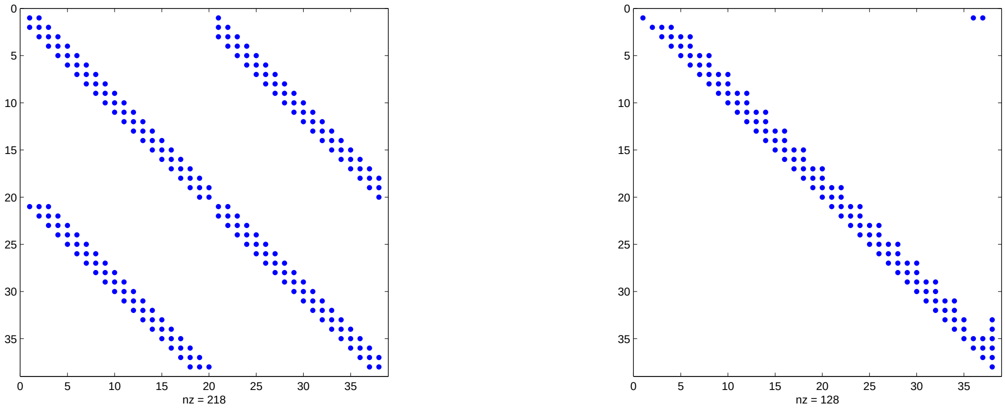

Figure 5: The correlative sparsity pattern of the polynomial SDP (35) with $n = 20$, and its Cholesky factor with a symmetric minimum degree ordering of its rows and columns.

where $\boldsymbol{F}$ denotes a mapping from $\mathbb{R}^n$ into $\mathbb{S}^n$ defined by

$$\boldsymbol{F}(\boldsymbol{y}) \;=\; \begin{pmatrix} 1-y_1^4 & 0 & 0 & \dots & 0 & y_1y_2 \\ 0 & 1-y_2^4 & 0 & \dots & 0 & y_2y_3 \\ 0 & 0 & \ddots & & 0 & y_3y_4 \\ \dots & \dots & \dots & \ddots & \dots & \dots \\ 0 & 0 & 0 & & 1-y_{n-1}^4 & y_{n-1}y_n \\ y_1y_2 & y_2y_3 & y_3y_4 & \dots & y_{n-1}y_n & 1-y_n^4 \end{pmatrix}.$$

This polynomial SDP is not correlatively sparse at all (*i.e.*, $G(N,E)$ becomes a complete graph) because all variables $y_1, y_2, \dots, y_n$ are involved in the single matrix inequality $\boldsymbol{F}(\boldsymbol{y}) \in \mathbb{S}^n_+$. Hence, the sparse SDP relaxation [16] is not effective for this problem. Applying the r-space conversion method using clique trees to the polynomial SDP under consideration, we have a polynomial SDP

$$\left.\begin{array}{ll} \text{minimize} & \boldsymbol{b}^T\boldsymbol{y} \\ \text{subject to} & \begin{pmatrix} 1-y_1^4 & y_1y_2 \\ y_1y_2 & z_1 \end{pmatrix} \succeq \boldsymbol{O}, \\ & \begin{pmatrix} 1-y_i^4 & y_iy_{i+1} \\ y_iy_{i+1} & -z_{i-1}+z_i \end{pmatrix} \succeq \boldsymbol{O} \ (i = 2,3,\dots,n-2), \\ & \begin{pmatrix} 1-y_{n-1}^4 & y_{n-1}y_n \\ y_{n-1}y_n & 1-y_n^4-z_{n-2} \end{pmatrix} \succeq \boldsymbol{O}, \end{array}\right\} \qquad (35)$$

which is equivalent to the original polynomial SDP. The resulting polynomial SDP now satisfies the correlative sparsity as shown in Figure 5. Thus the sparse SDP relaxation [16] is efficient for solving (35).

The correlative sparsity is important in linear SDPs, too. We have seen such a case in

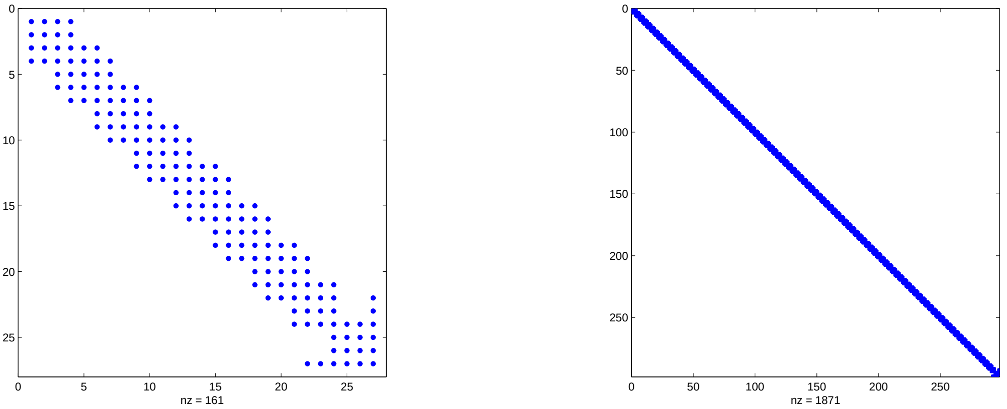

Figure 6: The correlative sparsity pattern of the SDP (37) induced from (4) with $n = 10$ and $n = 100$, and its Cholesky factor with a symmetric minimum degree ordering of its rows and columns.

Section 2.1. We can rewrite the SDP (3) as

$$\left.\begin{array}{ll}\text{maximize} & -\displaystyle\sum_{i=1}^{n-1}\left(A^0_{ii}X_{ii} + 2A^0_{i,i+1}X_{i,i+1}\right) - A^0_{nn}X_{nn} \\ \text{subject to} & \boldsymbol{I} - \displaystyle\sum_{i=1}^{n}\boldsymbol{E}_{ii}X_{ii} + \sum_{i=1}^{n-1}\boldsymbol{E}_{in}X_{i,i+1} \succeq \boldsymbol{O}, \\ & \displaystyle\sum_{1\le i\le j\le n}\boldsymbol{E}_{ij}X_{ij} \succeq \boldsymbol{O}, \end{array}\right\} \tag{36}$$

where $\boldsymbol{I}$ denotes the $n \times n$ identity matrix. Since the coefficient matrices of all real variables X_{ij} $(1 \le i \le j \le n)$ are nonzero in the last constraint, the correlative sparsity pattern graph $G(N, E)$ forms a complete graph. Applying the d-space conversion method using basis representation and the r-space conversion method using clique trees to the original SDP (3), we have reduced it to the SDP (4) in Section 2.1. We rewrite the constraints of the SDP (4) as an ordinary LMI form:

$$\left.\begin{array}{ll}\text{maximize} & \boldsymbol{b}^T\boldsymbol{y} \\ \text{subject to} & \boldsymbol{A}^k_0 - \displaystyle\sum_{h=1}^{s}\boldsymbol{A}^k_h y_h \succeq \boldsymbol{O} \ (k = 1, 2, \ldots, p). \end{array}\right\} \tag{37}$$

Here $p = 2n - 2$, $s = 3n - 3$, each $\boldsymbol{A}^k_h$ is 2×2 matrix $(k = 1, 2, \ldots, p,\ h = 0, 1, \ldots, 3n - 3)$, $\boldsymbol{b} \in \mathbb{R}^{3n-3}$, $\boldsymbol{y} \in \mathbb{R}^{3n-3}$, and each element y_h of $\boldsymbol{y}$ corresponds to some X_{ij} or some z_i. Comparing the SDP (36) with the SDP (37), we notice that the number of variables is reduced from $n(n + 1)/2$ to $3n - 3$, and the maximum size of the matrix inequality is reduced from n to 2. Furthermore, the correlative sparsity pattern graph becomes sparse. See Figure 6.

Now we consider an SDP of the form (37) in general. The edge set E of the correlative sparsity pattern graph $G(N, E)$ is written as

$$E = \left\{(g, h) \in N \times N : \ g \neq h,\ \boldsymbol{A}^k_g \neq \boldsymbol{O} \ \text{ and } \boldsymbol{A}^k_h \neq \boldsymbol{O} \text{ for some } k \ \right\},$$

where $N = \{1, 2, \ldots, s\}$. It is known that the graph $G(N, E)$ characterizes the sparsity pattern of the Schur complement matrix of the SDP (37). More precisely, if $\boldsymbol{R}$ denotes the $s \times s$ sparsity pattern of the Schur complement matrix, then $R_{gh} = 0$ if $(g, h) \notin E^{\bullet}$. Furthermore, if the graph $G(N, E)$ is chordal, then there exists a perfect elimination ordering, a simultaneous row and column ordering of the Schur complement matrix that allows a Cholesky factorization with no fill-in. For the SDP induced from (4), we have seen the correlative sparsity pattern with a symmetric minimum degree ordering of its rows and columns in Figure 6, which coincides with the sparsity pattern of the Schur complement matrix whose symbolic Cholesky factorization is shown in Figure 1.

Remark 6.1. As mentioned in Remark 5.1, the application of r-space conversion method using clique trees to reduce the SDP (3) to the SDP (4) can be implemented in many different ways. In practice, it should be implemented to have a better correlative sparsity in the resulting problem. For example, we can reduce the SDP (3) to

$$\left.\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{i=1}^{n-1} \left(A^0_{ii} X_{ii} + 2A^0_{i,i+1} X_{i,i+1}\right) + A^0_{nn} X_{nn} \\
\text{subject to} & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} X_{11} & -X_{12} \\ -X_{21} & -z_1 \end{pmatrix} \succeq \boldsymbol{O}, \\
& \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} X_{ii} & -X_{i,i+1} \\ -X_{i+1,i} & -z_i \end{pmatrix} \succeq \boldsymbol{O} \ (i = 2, 3, \ldots, n-2), \\
& \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} X_{n-1,n-1} & -X_{n-1,n} \\ -X_{n,n-1} & X_{n,n} + \sum_{i=1}^{n-2} z_i \end{pmatrix} \succeq \boldsymbol{O}, \\
& \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} -X_{ii} & -X_{i,i+1} \\ -X_{i+1,i} & -X_{i+1,i+1} \end{pmatrix} \succeq \boldsymbol{O} \ (i = 1, 2, \ldots, n-1),
\end{array}\right\} \tag{38}$$

which is different from the SDP (4). This is obtained by choosing a different clique tree in the r-space conversion method using clique trees for the SDP (3). In this case, all auxiliary variables z_i $(i = 1, 2, \ldots, n-2)$ are contained in a single matrix inequality. This implies that the corresponding correlative sparsity pattern graph $G(N, E)$ involves a clique with the size $n - 2$. See Figure 7. Thus the correlative sparsity becomes worse than the previous conversion. Among various ways of implementing the d- and r-space conversion methods, determining which one is effective for a better correlative sparsity will be a subject which requires further study.

7 Preliminary numerical results

We present numerical results to show the effectiveness of the d- and r-space conversion methods. For test problems, we first consider a quadratic SDP of the form

$$\text{minimize} \ \sum_{i=1}^{s} c_i x_i \ \text{subject to} \ \boldsymbol{M}(\boldsymbol{x}) \succeq \boldsymbol{O}, \tag{39}$$

where $c_i \in [0, 1]$ $(i = 1, 2, \ldots, s)$, $\boldsymbol{M} : \mathbb{R}^s \to \mathbb{S}^n$, and each non-zero element of M_{ij} of the mapping $\boldsymbol{M}$ is a polynomial in $\boldsymbol{x} = (x_1, x_2, \ldots, x_s) \in \mathbb{R}^s$ with degree at most 2. Then, three

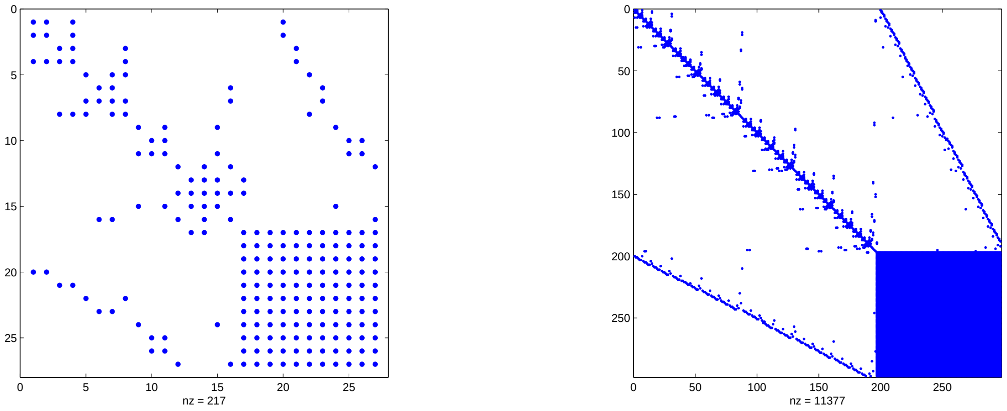

Figure 7: The correlative sparsity pattern of the SDP (37) induced from (38) with $n = 10$ and $n = 100$ where the rows and columns are simultaneously reordered by the Matlab function symamd (a symmetric minimum degree ordering).

max-cut problems, a Lovas theta problem, and a box-constrained quadratic problem from SDPLIB [5] are considered. A sensor network localization problem and discretized partial differential equations (PDE) with Neumann and Dirichlet condition are also tested.

7.1 SDP relaxations of a quadratic SDP

In this subsection, we apply the d- and r-space conversion methods to the quadratic SDP (39), and derive 4 kinds of SDP relaxations:

(a) a dense SDP relaxation without exploiting any sparsity.

(b) a sparse SDP relaxation by applying the d-space conversion method using basis representation given in Section 3.2.

(c) a sparse SDP relaxation by applying the r-space conversion method using clique trees in Section 5.1.

(d) a sparse SDP relaxation by applying both of the d-space conversion method using basis representation and the r-space conversion method using clique trees.

Some preliminary numerical results on these SDP relaxations are provided to compare their efficiency in Sections 7.2 and 7.3.

We write each non-zero element $M_{ij}(\boldsymbol{x})$ as

$$M_{ij}(\boldsymbol{x}) \quad = \quad \boldsymbol{Q}_{ij} \bullet \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{x}\boldsymbol{x}^T \end{pmatrix} \text{ for every } \boldsymbol{x} \in \mathbb{R}^s.$$

for some $\boldsymbol{Q}_{ij} \in \mathbb{S}^{1+s}$. Assume that the rows and columns of each $\boldsymbol{Q}_{ij}$ are indexed from 0 to s. Let us introduce a linearization (or lifting) $\widehat{M}_{ij} : \mathbb{R}^s \times \mathbb{S}^s \to \mathbb{R}$ of the quadratic function

$M_{ij} : \mathbb{R}^s \rightarrow \mathbb{R}$:

$$\widehat{M}_{ij}(\boldsymbol{x}, \boldsymbol{X}) \;=\; \boldsymbol{Q}_{ij} \bullet \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \text{for every } \boldsymbol{x} \in \mathbb{R}^s \text{ and } \boldsymbol{X} \in \mathbb{S}^s,$$

which induces a linearization (or lifting) $\widehat{\boldsymbol{M}} : \mathbb{R}^s \times \mathbb{S}^s \rightarrow \mathbb{S}^n$ of $\boldsymbol{M} : \mathbb{R}^s \rightarrow \mathbb{S}^n$ whose (i, j)th element is $\widehat{M}_{ij}$. Then we can describe the dense SDP relaxation (a) as

$$\text{minimize} \sum_{i=1}^{n} c_i x_i \text{ subject to } \widehat{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{X}) \succeq \boldsymbol{O} \text{ and } \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \succeq \boldsymbol{O}.$$

For simplicity, we rewrite the dense SDP relaxation above as

$$\text{(a)} \qquad \text{minimize} \sum_{i=1}^{n} c_i W_{0i} \text{ subject to } \widehat{\boldsymbol{M}}(\boldsymbol{W}) \succeq \boldsymbol{O},\ W_{00} = 1 \text{ and } \boldsymbol{W} \succeq \boldsymbol{O},$$

where

$$(W_{01}, W_{02}, \ldots, W_{0s}) \;=\; \boldsymbol{x}^T \in \mathbb{R}^s \text{ and } \boldsymbol{W} = \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \in \mathbb{S}^{1+s}.$$

Let $G(N', F')$ be the d-space sparsity pattern graph for the SDP (a) with $N' = \{0, 1, \ldots, s\}$, and F' = the set of distinct row and column index pairs (i, j) of W_{ij} that is necessary to evaluate the objective function $\sum_{i=1}^{n} c_i W_{0i}$ and/or the LMI $\boldsymbol{M}(\boldsymbol{W}) \succeq \boldsymbol{O}$. Let $G(N', E')$ be a chordal extension of $G(N', F')$, and $C'_1, C'_2, \ldots, C'_r$ be the maximal cliques of $G(N', E')$. Applying the d-space conversion method using basis representation described in Section 3.2, we obtain the SDP relaxation

$$\text{(b)} \qquad \begin{cases} \text{minimize} & \displaystyle\sum_{i=1}^{n} c_i W_{0i} \\ \text{subject to} & \widehat{\boldsymbol{M}}((W_{ij} : (i, j) \in \overline{J})) \succeq \boldsymbol{O},\ W_{00} = 1, \\ & \displaystyle\sum_{(i,j) \in J(C'_k)} \boldsymbol{E}_{ij} W_{ij} \in \mathbb{S}^{C'_k}_{+} \ (k = 1, 2, \ldots, r). \end{cases}$$

Here $\overline{J} = \cup_{k=1}^{r} J(C'_k)$, $(W_{ij} : (i, j) \in \overline{J})$ = the vector variable of the elements W_{ij} $((i, j) \in \overline{J})$ and

$$\widehat{\boldsymbol{M}}((W_{ij} : (i, j) \in \overline{J})) \;=\; \widehat{\boldsymbol{M}}(\boldsymbol{W}) \text{ for every } \boldsymbol{W} \in \mathbb{S}^s(E', 0).$$

To apply the r-space conversion method using clique trees given in Section 5.1 to the quadratic SDP (39), we assume that $\boldsymbol{M} : \mathbb{R}^s \rightarrow \mathbb{S}^n(E, 0)$ for some chordal graph $G(N, E)$ where $N = \{1, 2, \ldots, n\}$ and $E \subseteq N \times N$. Then, we convert the matrix inequality $\boldsymbol{M}(\boldsymbol{x}) \succeq \boldsymbol{O}$ in (39) into an equivalent system of matrix inequalities (31). The application of the LMI relaxation described above to (31) leads to the SDP relaxation

$$\text{(c)} \qquad \begin{cases} \text{minimize} & \displaystyle\sum_{i=1}^{n} c_i W_{0i} \\ \text{subject to} & \overline{\boldsymbol{M}}^k(\boldsymbol{W}) - \widetilde{\boldsymbol{L}}^k(\boldsymbol{z}) \succeq \boldsymbol{O} \ (k = 1, 2, \ldots, p),\ W_{00} = 1,\ \boldsymbol{W} \succeq \boldsymbol{O}, \end{cases}$$

where $\overline{\boldsymbol{M}}^k : \mathbb{S}^{1+s} \rightarrow \mathbb{S}^{C_k}$ denotes a linearization (or lifting) of $\widetilde{\boldsymbol{M}}^k : \mathbb{R}^s \rightarrow \mathbb{S}^{C_k}$. We may apply the linearization to (39) first to derive the dense SDP relaxation (a), and then apply the r-space conversion method using clique trees to (a). This results in the same sparse SDP relaxation (c) of (39). Note that both $\boldsymbol{M}$ and $\widehat{\boldsymbol{M}}$ take values from $\mathbb{S}^n(E,0)$, thus, they provide the same r-space sparsity pattern characterized by the chordal graph $G(N,E)$.

Finally, the sparse SDP relaxation (d) is derived by applying the d-space conversion method using basis representation to the the sparse LMI relaxation (c). We note that the d-space sparsity pattern graph for the SDP (c) with respect to the matrix variable $\boldsymbol{W} \in \mathbb{S}^{1+s}$ is the same as the one for the SDP (a). Hence, the sparse SDP relaxation (d) is obtained in the same way as the SDP (b) is obtained from the SDP (a). Consequently, we have the sparse SDP relaxation

$$\text{(d)} \quad \left\{ \begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{n} c_i W_{0i} \\ \text{subject to} & \overline{\boldsymbol{M}}^k((W_{ij} : (i,j) \in \overline{J})) - \widetilde{\boldsymbol{L}}^k(\boldsymbol{z}) \succeq \boldsymbol{O} \ (k = 1,2,\ldots,p), \ W_{00} = 1, \\ & \displaystyle\sum_{(\alpha,\beta)\in J(C'_k)} \boldsymbol{E}_{\alpha\beta} W_{\alpha\beta} \in \mathbb{S}_+^{C'_j} \ (j = 1,2,\ldots,r). \end{array} \right.$$

Here $\overline{J} = \cup_{k=1}^r J(C'_k)$ and

$$\overline{\boldsymbol{M}}^k((W_{ij} : (i,j) \in \overline{J})) \ = \ \overline{\boldsymbol{M}}^k(\boldsymbol{W}) \text{ for every } \boldsymbol{W} \in \mathbb{S}^s(E',0).$$

7.2 A tridiagonal quadratic SDP

In this and next subsection, two sparse cases of the quadratic SDP (39) are considered with numerical results on the SDP relaxations (a) $\sim$ (d). The SDP relaxation problems were solved by SeDuMi on 2.66 GHz Dual-Core Intel Xeon with 12GB memory.

For every $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,n$, define an $(1+s)\times(1+s)$ symmetric matrix $\boldsymbol{Q}_{ij}$ such that

$$\begin{array}{rcl} \boldsymbol{Q}_{ij} & = & \left\{ \begin{array}{ll} \begin{pmatrix} 1 & \boldsymbol{0}^T \\ \boldsymbol{0} & -\boldsymbol{D}_i \end{pmatrix} & \text{if } i = j, \\ \begin{pmatrix} 0 & \boldsymbol{a}_i^T/2 \\ \boldsymbol{a}_i/2 & \boldsymbol{O} \end{pmatrix} & \text{if } j = i+1 \ \text{ and } i = 1,2,\ldots,n-1, \\ \begin{pmatrix} 0 & \boldsymbol{a}_j^T/2 \\ \boldsymbol{a}_j/2 & \boldsymbol{O} \end{pmatrix} & \text{if } j = i-1 \ \text{ and } i = 2,3,\ldots,n, \\ \boldsymbol{O} & \text{otherwise.} \end{array} \right. \\ \boldsymbol{D}_i & = & \text{an } s \times s \text{ diagonal matrix with diagonal elements chosen} \\ & & \text{randomly from the interval } (0,1). \\ \boldsymbol{a}_i & = & \text{an } s \text{ dimensional column vector with elements chosen} \\ & & \text{randomly from the interval } (-1,1). \end{array}$$

We see that, for every $\boldsymbol{x} \in \mathbb{R}^s$,

$$\begin{array}{rcl} M_{ii}(\boldsymbol{x}) & = & \boldsymbol{Q}_{ii} \bullet \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{x}\boldsymbol{x}^T \end{pmatrix} = 1 - \displaystyle\sum_{j=1}^{s} [\boldsymbol{D}_i]_{jj} x_j^2, \\ M_{ij}(\boldsymbol{x}) & = & \boldsymbol{Q}_{ij} \bullet \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{x}\boldsymbol{x}^T \end{pmatrix} = \boldsymbol{a}_i^T \boldsymbol{x} \text{ if } j = i+1, \ i = 1,2,\ldots,n-1 \end{array}$$

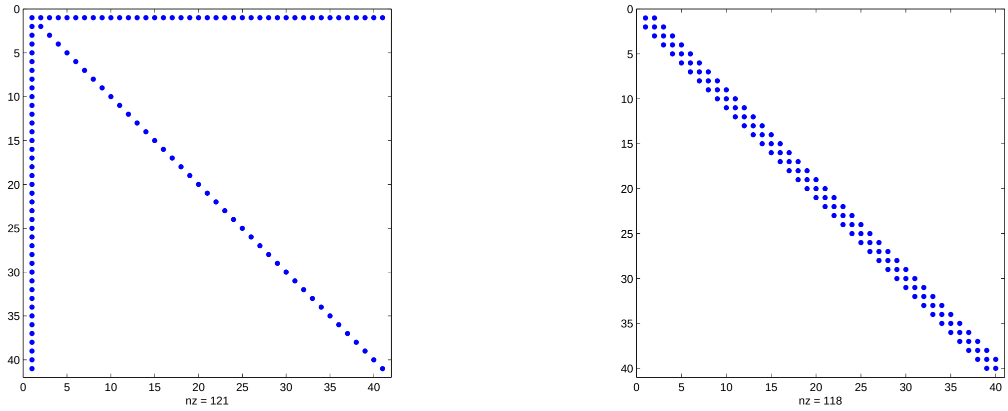

Figure 8: The d-space sparsity pattern (the left figure) and the r-space sparsity pattern (the right figure) of the the tridiagonal quadratic SDP with $s = 40$ and $n = 40$.

$$\begin{aligned} M_{ij}(\boldsymbol{x}) &= \boldsymbol{Q}_{ij} \bullet \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{x}\boldsymbol{x}^T \end{pmatrix} = \boldsymbol{a}_j^T \boldsymbol{x} \text{ if } j = i-1, \ i = 2, 3, \ldots, n, \\ M_{ij}(\boldsymbol{x}) &= M_{ji}(\boldsymbol{x}) \ (i = 1, 2, \ldots, n, j = 1, 2, \ldots, n). \end{aligned}$$

Figure 8 shows the d- and r-space sparsity patterns when $s = 40$ and $n = 40$.

Table 2 shows the SeDuMi CPU time in seconds, the size of the Schur complement matrix, and the maximum size of matrix variables of the SDP relaxation problems (a), (b), (c) and (d) of the tridiagonal quadratic SDP. The size of the Schur complement matrix and the maximum size of matrix variables are essential factors affecting the CPU time. Comparing the CPU time of (b) with that of (a), we observe that the d-space conversion method using basis representation used in (b) works very effectively. In (b), the $(1+s) \times (1+s)$ matrix variable $\boldsymbol{W}$ is decomposed into s 2×2 matrix variables, while the size of the matrix inequality remains the same. The reduction in the SeDuMi CPU time to solve the SDP relaxation problem is mainly due to the reduction in the size of the Schur complement matrix. On the other hand, using only the r-space conversion method using clique trees in (c) fails in reducing the maximum size of matrix variables, the size of the Schur complement matrix, and the SeDuMi CPU time. But when combined with the d-space conversion method using basis representation as in (d), both of the $(1+s) \times (1+s)$ matrix variable $\boldsymbol{W}$ and the matrix inequality are decomposed into 2×2 matrices, and the size of the Schur complement matrix decreases more than those in (a) and (c). These contribute to a further reduction in the SeDuMi CPU time from (b).

Table 3 shows numerical results on the tridiagonal quadratic SDP with the dimension s of the domain space of the variable vector $\boldsymbol{x}$ fixed to 40. In this case, we observe that the r-space conversion method using clique trees in (c) works more effectively than the d-space conversion method using basis representation as the dimension n of the range space of the matrix inequality $\boldsymbol{M}(\boldsymbol{x}) \succeq \boldsymbol{O}$ becomes larger. We also see that (d) attains the shortest SeDuMi CPU time.

		SeDuMi CPU time in seconds (the size of the Schur complement matrix, the max. size of matrix variables)			
s	n	(a)	(b)	(c)	(d)
40	40	8.38	0.97	8.83	0.68
		(860, 41)	(80, 40)	(898, 41)	(118, 2)
80	80	384.43	11.72	402.86	1.58
		(3320, 81)	(160, 80)	(3398, 81)	(238, 2)
160	160	-	33.26	-	4.71
			(320, 160)		(478, 2)
320	320	-	100.36	-	24.57
			(640, 320)		(958, 2)

Table 2: Numerical results on the tridiagonal SDP with $s = n$ and "-" indicates out of memory error in Matlab.

		SeDuMi CPU time in seconds (the size of the Schur complement matrix, the max. size of matrix variables)			
s	n	(a)	(b)	(c)	(d)
40	80	30.76	6.27	28.70	1.52
		(860, 80)	(80, 80)	(938, 41)	(158, 2)
40	160	41.9	21.86	32.44	2.82
		(860, 160)	(80, 160)	(1081, 41)	(238, 2)
40	320	95.46	69.98	40.15	5.25
		(860, 320)	(80, 320)	(1178, 41)	(398, 2)
40	640	474.51	393.23	46.26	11.60
		(860, 640)	(80, 640)	(1498, 41)	(718, 2)

Table 3: Numerical results on the tridiagonal SDP with the dimension s of the domain space of the variable vector $\boldsymbol{x}$ fixed to 40.

7.3 Quadratic SDPs with d- and r-space sparsity from randomly generated sparse graphs

Quadratic SDP problem were constructed by first generating two graphs $G(N_d, E_d)$ with $N_d = \{1, 2, \ldots, 1+s\}$ and $G(N_r, E_r)$ with $N_r = \{1, 2, \ldots, n\}$ and using the Matlab program generateProblem.m [12], which was developed for sensor network localization problems. Sparse chordal extensions $G(N_d, \overline{E}_d)$ and $G(N_r, \overline{E}_r)$ were then obtained by the Matlab functions symamd.m and chol.m. Next, we generated data matrices $\boldsymbol{Q}_{ij} \in \mathbb{S}^{1+s}$ $(i = 1, 2, \ldots, n, j = 1, 2, \ldots, n)$ and a data vector $\boldsymbol{c} \in \mathbb{R}^s$ so that the d- and r-space sparsity pattern graphs of the resulting quadratic SDP coincide with $G(N_d, E_d)$ and $G(N_r, E_r)$, respectively. Some characteristics of the chordal extensions $G(N_d, \overline{E}_d)$ of $G(N_r, E_r)$ and $G(N_r, \overline{E}_r)$ of $G(N_r, E_r)$ used in the experiments are shown in Table 4.

For the problem with $s = 40$ and $n = 640$, the d- and r-space sparsity pattern obtained from the symmetric approximate minimum degree permutation of rows and columns by the Matlab function symamd.m is displayed in Figure 9.

		Domain space sparsity				Range space sparsity			
s	n	$\#\overline{E}_d$	NoC	MaxC	MinC	$\#\overline{E}_r$	NoC	MaxC	MinC
80	80	143	63	3	3	216	72	7	3
320	320	649	260	7	3	840	301	9	3
40	160	70	30	3	3	426	150	7	3
40	640	70	30	3	3	1732	616	13	3

Table 4: Some characteristics of d- and r-space sparsity patterns of the tested quadratic SDPs. $\#\overline{E}_d$ (or $\#\overline{E}_r$) denotes the number of edges of $G(N_d, \overline{E}_d)$ (or $G(N_r, \overline{E}_r)$), NoC the number of the maximal cliques of $G(N_d, \overline{E}_d)$ (or $G(N_r, \overline{E}_r)$), MaxC the size of the largest maximal cliques of $G(N_d, \overline{E}_d)$ (or $G(N_r, \overline{E}_r)$), and MinC the size of the smallest maximal cliques of $G(N_d, \overline{E}_d)$ (or $G(N_r, \overline{E}_r)$).

Table 5 shows numerical results on the quadratic SDPs whose d- and r-space sparsity characteristics are given in Table 4. We observe that both the d-space conversion method using basis representation in (b) and the r-space conversion method using clique tree in (c) work effectively, and that their combination in (d) results in the shortest CPU time among the four methods.

7.4 SDPs from some applications

For additional numerical experiments, we selected five SDP problems from SDPLIB [5]. Moreover, as SDP problems from applications, we tested a sensor network localization problem and discretized PDE with Neumann and Dirichlet boundary condition. The test problems in Table 6 are

mcp500-1, maxG11, maxG32: An SDP relaxation of the max cut problem from SDPLIB.

thetaG11: An SDP relaxation of the Lovasz theta problem from SDPLIB.

qpG11: An SDP relaxation of the box constrained quadratic problem form SDPLIB.

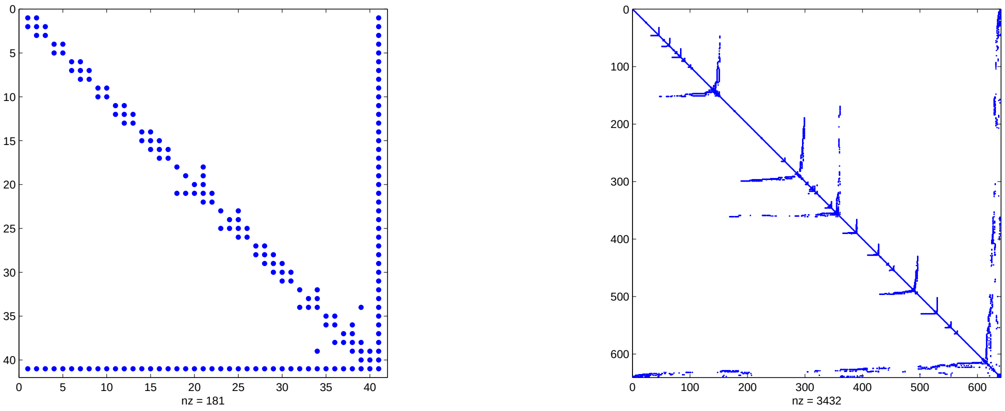

Figure 9: The d-space sparsity pattern of the quadratic SDP with $s = 40$ and $n = 640$ on the left and the r-space sparsity pattern on the right.

		SeDuMi CPU time in seconds (the size of the Schur complement matrix, the max. size of matrix variables)			
s	n	(a)	(b)	(c)	(d)
80	80	296.51 (3321, 81)	1.38 (224, 80)	1.58 (801, 81)	0.73 (252, 19)
320	320	-	74.19 (970, 320)	80.09 (322, 321)	35.20 (1216, 20)
40	160	6.70 (861, 160)	4.22 (111, 160)	2.91 (1626, 41)	0.74 (207, 21)
40	640	158.95 (861, 640)	151.20 (111, 640)	120.86 (6776, 41)	5.71 (772, 21)

Table 5: Numerical results on the quadratic SDPs with d- and r- sparsity from randomly generated sparse graphs. “-” indicates out of memory error in Matlab.

d2n01s1000a100FSDP: A full SDP relaxation [4] of the sensor network localization problem with 1000 sensors, 100 anchors distributed in $[0,1]^2$, radio range = 0.1, and noise = 10%. The method (iii) in Table 6 for this problem is equivalent to the method used in SFSDP [11], a sparse version of full SDP relaxation.

pdeEllipticNeum11: An SDP relaxation of the discretized nonlinear, elliptic PDE (4.4) (Case II, Neumann boundary condition) of [19]. We choose a 11 × 11 grid for the domain $[0,1]^2$ of the PDE.

pdeEllipticBifur20: An SDP relaxation of the discretized nonlinear, elliptic PDE (4.5) (Dirichlet boundary condition) of [19]. We choose a 20 × 20 grid for the domain $[0,1]^2$ of the PDE.

The SDP relaxations (i), (ii) and (iii) in Table 6 indicate

(i) a dense SDP relaxation without exploiting any sparsity.

(ii) a sparse SDP relaxation by applying the d-space conversion method using clique trees given in Section 3.1.

(iii) a sparse SDP relaxation by applying the d-space conversion method using basis representation given in Section 3.2.

Table 6 shows that CPU time spent by (ii) is shorter than that by (i) and (iii) for all tested problems except for mcp500-1 and pdeEllipticNeum11. Notice that it took shorter CPU time to solve (iii) than (i) except for maxG32 and thetaG11. We confirm that applying at least one of the d-space conversion methods greatly reduces CPU time for the test problems. The d-space sparsity patterns for the test problems are displayed in Figures 10 and 11.

	SeDuMi CPU time (size.SC.mat., Max.size.mat.var.)		
Problem	(i)	(ii)	(iii)
mcp500-1	65.5 (500, 500)	94.5 (7222, 44)	15.9 (2878, 44)
maxG11	220.5 (800, 800)	12.1 (2432, 80)	26.8 (8333, 24)
maxG32	5373.8 (2000, 2000)	971.4 (13600, 210)	Out of memory
thetaG11	345.9 (2401, 801)	23.9 (4237, 81)	458.5 (9134, 25)
qpG11	2628.5 (800, 1600)	16.0 (2432, 80)	72.5 (9133, 24)
d2n01s1000a100FSDP	5193.5 (4949, 1002)	16.9 (7260, 45)	19.5 (15691, 17)
pdeEllipticNeum11	216.1 (1453, 485)	2.2 (1483, 17)	2.1 (1574, 4)
pdeEllipticBifur20	1120.4 (2401, 801)	4.3 (2451, 17)	5.3 (2001, 3)

Table 6: Numerical results on SDPs from some applications. size.SC.mat. denotes the size of the Schur complement matrix and Max.size.mat.var. the maximum size of matrix variables.

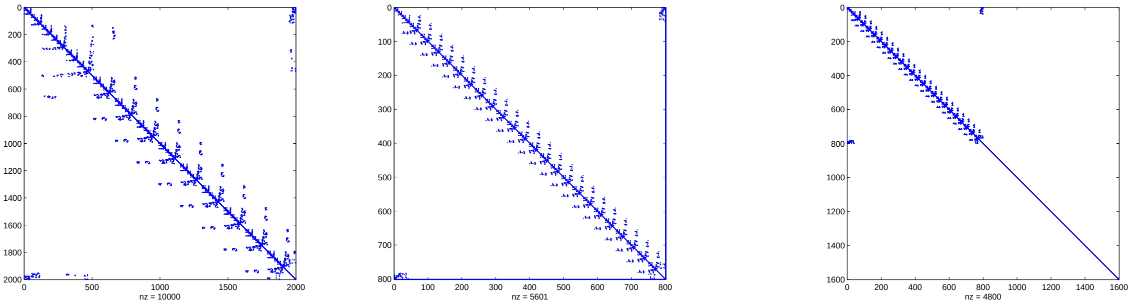

Figure 10: The d-space sparsity pattern with the symmetric approximate minimum degree permutation of rows and columns provided by the Matlab function symamd.m for maxG32(left), thetaG11(middle) and qpG11(right)

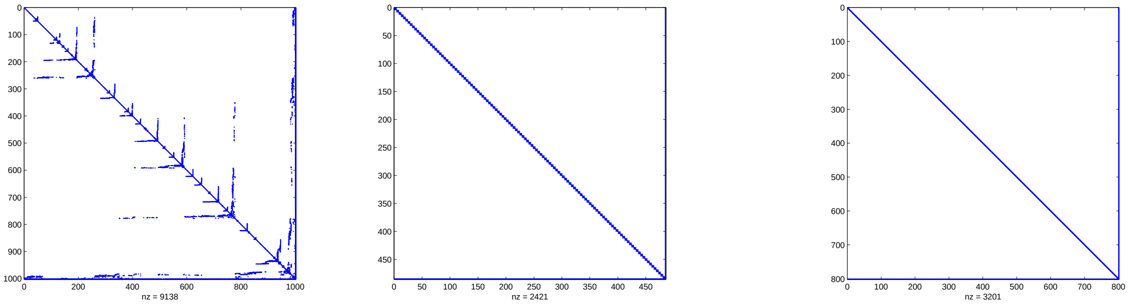

Figure 11: The d-space sparsity pattern with the symmetric approximate minimum degree permutation of rows and columns provided by the Matlab function symamd.m for d2n01s1000a100FSDP(left), pdeEllipticNeum11(middle) and pdeEllipticBifur20(right).

8 Concluding discussions

Our focus has been on developing a theoretical framework consisting of the d- and r-space conversion methods to exploit structured sparsity, characterized by a chordal graph structure, via the positive semidefinite matrix completion for an optimization problem involving linear and nonlinear matrix inequalities. The two d-space conversion methods are provided for a matrix variable $\boldsymbol{X}$ in objective and/or constraint functions of the problem, which is required to be positive semidefinite. The methods decompose $\boldsymbol{X}$ into multiple smaller matrix variables. The two r-space conversion methods are aimed at a matrix inequality in the constraint of the problem. In these methods, the matrix inequality is converted into multiple smaller matrix inequalities. We have also discussed how the conversion methods enhance the correlative sparsity in linear and polynomial SDPs.

We have not described technical details of practical implementation in this paper. As mentioned in Remarks 3.1, 5.1 and 6.1, the d-space conversion method using clique trees and the r-space conversion method using clique trees have plenty of flexibilities in implementation. This should be explored further for computational efficiency. In addition, how the four methods should be combined to solve a given optimization problem in practice needs to be studied.

In the preliminary numerical experiments in Section 7, we have observed that the computational performance is greatly improved by applying d- and r-space conversion methods. Extensive numerical experiments are necessary before presenting a practical version of the four conversion methods.

Acknowledgments

The authors would like to thank Dr. Naonori Kakimura for informing them that Theorem 4.2 was presented in the paper [1].

References

[1] J. Agler, J. W. Helton, S. McCullough, and L. Rodman (1988) Positive semidefinite matrices with a given sparsity pattern, *Linear Algebra Appl.*, **107**, 101-149.

[2] C. Ashcraft, D. Pierce, D. K. Wah and J. Wu (1999) "The reference manual for SPOOLES, release 2.3: An object oriented software library for solving linear systems of equations," Boeing Shared Services Group, Seattle, WA, January.

[3] J. R. S. Blair and B. Peyton (1993) "An introduction to chordal graphs and clique trees," In: A. George, J. R. Gilbert and J. W. H. Liu eds., *Graph Theory and Sparse Matrix Computation*, Springer, New York, pp.1-29.

[4] P. Biswas and Y. Ye (2004) "Semidefinite programming for ad hoc wireless sensor network localization," in *Proceedings of the third international symposium on information processing in sensor networks*, ACM press, 46–54.

[5] B. Borchers (1999) "SDPLIB 1.2, A library of semidefinite programming test problems," *Optim. Methods Softw.*, **11-12**, 683-689.

[6] M. Fukuda, M. Kojima, K. Murota and K. Nakata (2000) "Exploiting sparsity in semidefinite programming via matrix completion I: General framework," *SIAM J. Optim.*, **11**, 647-674.

[7] R. Grone, C. R. Johnson, E. M. Sá and H. Wolkowitz (1984) "Positive definite completions of a partial hermitian matrices," *Linear Algebra Appl.*, **58**, 109-124.

[8] D. Henrion and J. B. Lasserre (2006) "Convergent relaxations of polynomial matrix inequalities and static output feedback," *IEEE Trans. Automatic Control*, **51**, 192-202.

[9] C. W. J. Hol and C. W. Scherer, "Sum of squares relaxations for polynomial semidefinite programming," *Proc. Symp. on Mathematical Theory of Networks and Systems (MTNS)*, Leuven, Belgium, 2004.

[10] G. Karypis and V. Kumar (1998), "METIS — A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing ordering of sparse matrices, version 4.0 —," Department of Computer Science/Army HPC Research Center, University of Minnesota, Minneapolis, MN, September.

[11] S. Kim, M. Kojima and H. Waki (2009) "Exploiting Sparsity in SDP Relaxation for Sensor Network Localization," *SIAM J. Optim.,* **20**, 1, 192-215.

[12] S. Kim, M. Kojima H. Waki and M. Yamashita (2009) "SFSDP: a Sparse Version of Full Semidefinite Programming Relaxation for Sensor Network Localization Problems," Research Report B-457, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Meguro, Tokyo 152-8552.

[13] K. Kobayashi, S. Kim and M. Kojima (2008) "Correlative sparsity in primal-dual interior-point methods for LP, SDP and SOCP," *Appl. Math. Optim.*, **58**, 69-88.

[14] M. Kojima, "Sums of Squares Relaxations of Polynomial Semidefinite Programs," Research Report B-397, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152-8552, Japan, November 2003.

[15] M. Kojima and M. Muramatsu (2007) "An Extension of Sums of Squares Relaxations to Polynomial Optimization Problems over Symmetric Cones, " *Math. Program.,* **110**, 315-336.

[16] M. Kojima and M. Muramatsu (2009) "A note on sparse SOS and SDP relaxations for polynomial optimization problems over symmetric cones ," *Comput. Optim. Appl.*, **42**, 31-41.

[17] J. B. Lasserre (2001) "Global optimization with polynomials and the problems of moments," *SIAM J. Optim.*, **11**, 796–817.

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

[19] M. Mevissen, M. Kojima, J. Nie and N. Takayama (2008) "Solving partial differential equations via sparse SDP relaxations," *Paci. J. Optim.*, **4**, 2, 213-241.

[20] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima and K. Murota (2003) "Exploiting sparsity in semidefinite programming via matrix completion II: Implementation and numerical results," *Math. Program.*, **95**, 303-327.

[21] Ju. E. Nesterov and A. S. Nemirovski (1994) *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, SIAM, Philadelphia, PA.

[22] J. F. Strum (1999) "SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods Softw.*, **11 & 12**, 625-653.

[23] 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**, 218–242.

[24] H. Waki, S. Kim, M. Kojima, M. Muramatsu and H. Sugimoto (2008) "SparsePOP : a Sparse Semidefinite Programming Relaxation of Polynomial Optimization Problems," *ACM Trans. Math. Software*, **35**, 2, 15.