EXPLOITING SPARSITY IN SEMIDEFINITE PROGRAMMING VIA MATRIX COMPLETION I: GENERAL FRAMEWORK*

MITUHIRO FUKUDA†¶, MASAKAZU KOJIMA†, KAZUO MUROTA‡, AND KAZUHIDE NAKATA§

Abstract. A critical disadvantage of primal-dual interior-point methods compared to dual interior-point methods for large scale SDPs (semidefinite programs) has been that the primal positive semidefinite matrix variable becomes fully dense in general even when all data matrices are sparse. Based on some fundamental results about positive semidefinite matrix completion, this article proposes a general method of exploiting the aggregate sparsity pattern over all data matrices to overcome this disadvantage. Our method is used in two ways. One is a conversion of a sparse SDP having a large scale positive semidefinite matrix variable into an SDP having multiple but smaller size positive semidefinite matrix variables to which we can effectively apply any interior-point method for SDPs employing a standard block-diagonal matrix data structure. The other way is an incorporation of our method into primal-dual interior-point methods which we can apply directly to a given SDP. In Part II of this article, we will investigate an implementation of such a primal-dual interior-point method based on positive definite matrix completion, and report some numerical results.

Key words. semidefinite programming, primal-dual interior-point method, matrix completion problem, chordal graph

AMS subject classifications. 90C22, 90C51, 05C50

1. Introduction.

Let $\boldsymbol{R}^n$ denote the n-dimensional Euclidean space, and $\mathcal{S}^n$ the space of $n \times n$ symmetric matrices with the Frobenius inner product $\boldsymbol{X} \bullet \boldsymbol{Y} = \sum_{i=1}^n \sum_{j=1}^n X_{ij} Y_{ij}$ for $\boldsymbol{X},\ \boldsymbol{Y} \in \mathcal{S}^n$. We will use the notation $\boldsymbol{X} \in \mathcal{S}^n_+$ and $\boldsymbol{X} \in \mathcal{S}^n_{++}$ to designate that $\boldsymbol{X} \in \mathcal{S}^n$ is positive semidefinite and positive definite, respectively. Given $\boldsymbol{A}_p \in \mathcal{S}^n$ $(p = 0, 1, \ldots, m)$ and $\boldsymbol{b} \in \boldsymbol{R}^m$, we are concerned with the standard equality form SDP (semidefinite program)

$$
\left.\begin{array}{ll} \text{minimize} & \boldsymbol{A}_0 \bullet \boldsymbol{X} \\ \text{subject to} & \boldsymbol{A}_p \bullet \boldsymbol{X} = b_p \ (p = 1, 2, \ldots, m),\ \boldsymbol{X} \in \mathcal{S}^n_+, \end{array}\right\} \tag{1.1}
$$

and its dual

$$
\left.\begin{array}{ll} \text{maximize} & \displaystyle\sum_{p=1}^m b_p z_p \\ \text{subject to} & \displaystyle\sum_{p=1}^m \boldsymbol{A}_p z_p + \boldsymbol{Y} = \boldsymbol{A}_0,\ \boldsymbol{Y} \in \mathcal{S}^n_+. \end{array}\right\} \tag{1.2}
$$

In recent years, many interior-point methods have been proposed for SDPs. Among others, primal-dual interior-point methods have been studied intensively and extensively [1, 13, 15, 16, 20, 21, 24, 27, etc.]. They generate a sequence $\{(\boldsymbol{X}^k, \boldsymbol{Y}^k, \boldsymbol{z}^k) \in$

*

†Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan (mituhiro@is.titech.ac.jp, kojima@is.titech.ac.jp).

‡Research Institute for Mathematical Sciences, Kyoto University, Kyoto 606-8502 Japan (murota@kurims.kyoto-u.ac.jp).

§Department of Applied Physics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8565 Japan (nakata@zzz.t.u-tokyo.ac.jp).

¶This author was supported by Ministry of Education, Science, Sports and Culture of Japan.

$\mathcal{S}^n \times \mathcal{S}^n \times \boldsymbol{R}^m\}$ such that $\boldsymbol{X}^k \in \mathcal{S}^n_{++}$ and $\boldsymbol{Y}^k \in \mathcal{S}^n_{++}$. At each iteration, they first compute a search direction $(d\boldsymbol{X}, d\boldsymbol{Y}, d\boldsymbol{z}) \in \mathcal{S}^n \times \mathcal{S}^n \times \boldsymbol{R}^m$, and then they choose a step length $\alpha^k > 0$ such that the next iterate defined by

$$(\boldsymbol{X}^{k+1}, \boldsymbol{Y}^{k+1}, \boldsymbol{z}^{k+1}) = (\boldsymbol{X}^k, \boldsymbol{Y}^k, \boldsymbol{z}^k) + \alpha^k (d\boldsymbol{X}, d\boldsymbol{Y}, d\boldsymbol{z}) \tag{1.3}$$

still satisfies $\boldsymbol{X}^{k+1} \in \mathcal{S}^n_{++}$ and $\boldsymbol{Y}^{k+1} \in \mathcal{S}^n_{++}$.

The computation of a search direction $(d\boldsymbol{X}, d\boldsymbol{Y}, d\boldsymbol{z})$ is usually reduced to an $m \times m$ square system of linear equations $\boldsymbol{B}d\boldsymbol{z} = \boldsymbol{s}$, which is often called the Schur complement equation. Here the coefficient matrix $\boldsymbol{B}$ (hence the search direction $(d\boldsymbol{X}, d\boldsymbol{Y}, d\boldsymbol{z})$) varies with the individual method. See [15, 21, 27, etc.] for more details on various search directions used in primal-dual interior-point methods. The size m of the matrix $\boldsymbol{B}$ coincides with the number of equality constraints in the primal SDP (1.1), so that m can be as large as $n(n+1)/2$ even if the constraint matrices $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_m$ are assumed to be linearly independent. For a fixed n, as m becomes larger, more CPU time is spent in

(a) the computation of the coefficient matrix $\boldsymbol{B}$, and
(b) the computation of the solution $d\boldsymbol{z}$ of $\boldsymbol{B}d\boldsymbol{z} = \boldsymbol{s}$.

See [5, 23]. Fujisawa-Kojima-Nakata [7] proposed an efficient method for computing the coefficient matrix $\boldsymbol{B}$ when the data matrices $\boldsymbol{A}_p \in \mathcal{S}^n$ $(p = 1, 2, \ldots, m)$ are sparse. Also, the computation of $\boldsymbol{B}$ can be carried out efficiently, when the data matrices $\boldsymbol{A}_p \in \mathcal{S}^n$ $(p = 1, 2, \ldots, m)$ are of rank 1 or 2 [2, 12].

In general, the matrix $\boldsymbol{B}$ is fully dense. Therefore, as m becomes larger, it becomes more difficult to apply direct methods such as the Cholesky factorization to the computation of the solution $d\boldsymbol{z}$ of $\boldsymbol{B}d\boldsymbol{z} = \boldsymbol{s}$. If m is larger than ten thousand, it is even impossible to store the coefficient matrix in standard workstations. [19, 23] studied the use of iterative methods such as the conjugate gradient method to overcome the storage problem for such large and dense systems of linear equations.

Another difficulty in applying primal-dual interior-point methods to large scale SDPs arises from the fact that

(c) the $n \times n$ primal positive semidefinite matrix variable $\boldsymbol{X}$ is fully dense in general even when all the data matrices $\boldsymbol{A}_p \in \mathcal{S}^n$ $(p = 0, 1, \ldots, m)$ are sparse.

On the other hand, the dual positive semidefinite matrix variable $\boldsymbol{Y}$, which is computed by

$$\boldsymbol{Y} = \boldsymbol{A}_0 - \sum_{p=1}^{m} \boldsymbol{A}_p z_p,$$

inherits the sparsity of the data matrices $\boldsymbol{A}_p \in \mathcal{S}^n$ $(p = 0, 1, \ldots, m)$. This difference has been a critical disadvantage of primal-dual interior-point methods compared to the dual interior-point method [2] which generates a sequence $\{(\boldsymbol{Y}^k, \boldsymbol{z}^k)\}$ only in the dual space.

The purpose of the current paper is to resolve the difficulty (c). Let V denote the set $\{1, 2, \ldots, n\}$ of row/column indices of the data matrices $\boldsymbol{A}_0, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_m$. For every pair of subsets S and T of V, we use the notation $\boldsymbol{X}_{ST}$ for the submatrix of $\boldsymbol{X}$ obtained by deleting all rows $i \notin S$ and all columns $j \notin T$. To outline the basic idea behind our method, we introduce the *aggregate sparsity pattern* E of the data matrices given by

$$E = \{(i, j) \in V \times V : [\boldsymbol{A}_p]_{ij} \neq 0 \text{ for some } p \in \{0, 1, 2, \ldots, m\}\}. \tag{1.4}$$

Here $[\boldsymbol{A}_p]_{ij}$ denotes the (i,j)th entry of $\boldsymbol{A}_p$. Geometrically, it is convenient to identify the aggregate sparsity pattern E with the *aggregate sparsity pattern matrix* $\boldsymbol{A}$ having unspecified nonzero numerical values in E. Since the matrices $\boldsymbol{A}_0, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_m$ are all symmetric, $(i,j) \in E$ if and only if $(j,i) \in E$; hence the corresponding matrix $\boldsymbol{A}$ is symmetric. (In section 2, we will represent the aggregate sparsity pattern E in terms of a graph).

Assume that a collection of nonempty subsets $C_1, C_2, \ldots, C_\ell$ of V satisfies the following two conditions:

(i) $E \subseteq F \equiv \bigcup_{r=1}^{\ell} C_r \times C_r$.

(ii) Any partial symmetric matrix $\boldsymbol{X}$ with entries $X_{ij} = \bar{X}_{ij} \in \boldsymbol{R}$ $((i,j) \in F)$ has a *positive (semi)definite matrix completion* (*i.e.,* given any $\bar{X}_{ij} \in \boldsymbol{R}$ $((i,j) \in F)$, there exists a positive (semi)definite $\boldsymbol{X} \in \mathcal{S}^n$ such that $X_{ij} = \bar{X}_{ij} \in \boldsymbol{R}$ $((i,j) \in F)$) if and only if the submatrices $\bar{\boldsymbol{X}}_{C_rC_r}$ $(r = 1, 2, \ldots, \ell)$ are all positive (semi)definite.

From condition (i), we observe that values of the objective and constraint linear functions $\boldsymbol{A}_p \bullet \boldsymbol{X}$ $(p = 0, 1, \ldots, m)$ involved in the SDP (1.1) are completely determined by values of entries X_{ij} $((i,j) \in F)$ and independent of values of entries X_{ij} $((i,j) \notin F)$. In other words, if two $\boldsymbol{X},\ \boldsymbol{X}' \in \mathcal{S}^n$ satisfy $X_{ij} = X'_{ij}$ $((i,j) \in F)$ then

$$\boldsymbol{A}_p \bullet \boldsymbol{X} = \boldsymbol{A}_p \bullet \boldsymbol{X}' \ (p = 0, 1, \ldots, m).$$

The remaining entries X_{ij} $((i,j) \notin F)$ only affect whether $\boldsymbol{X}$ is positive (semi)definite. Now we know by condition (ii) whether we can assign some appropriate values to those remaining entries X_{ij} $((i,j) \notin F)$ so that the resulting whole matrix $\boldsymbol{X}$ becomes positive (semi)definite. Therefore the SDP (1.1) is equivalent to

$$\left.\begin{array}{ll} \text{minimize} & \displaystyle\sum_{(i,j)\in F} [\boldsymbol{A}_0]_{ij} X_{ij} \\ \text{subject to} & \displaystyle\sum_{(i,j)\in F} [\boldsymbol{A}_p]_{ij} X_{ij} = b_p \ (p = 1, 2, \ldots, m), \\ & \boldsymbol{X}_{C_rC_r} \in \mathcal{S}_+^{C_r} \ (r = 1, 2, \ldots, \ell). \end{array}\right\}$$

Here $\mathcal{S}_+^{C_r}$ denotes the set of $\sharp C_r \times \sharp C_r$ positive semidefinite symmetric matrices with entries specified in $C_r \times C_r$, and $\sharp C_r$ the number of elements of C_r.

Section 2 is devoted to some fundamental results on the positive (semi)definite matrix completion problem. In particular, we present a characterization of the positive (semi)definite matrix completion in terms of chordal graphs based on the paper [11] by Gron, Johnson, Sá and Wolkowicz, and relate it to the perfect elimination ordering for the Cholesky factorization with no fill-in. Based on the former characterization, we describe in section 3 a general method of choosing a collection of subsets $C_1, C_2, \ldots, C_\ell$ satisfying conditions (i) and (ii) above. The latter perfect elimination ordering leads us to a sparse factorization formula for the maximum-determinant positive definite matrix completion in the latter part of section 2. A variation of this formula, which we will call the sparse clique-factorization formula, plays an essential role in the primal-dual interior-point method based on positive definite matrix completion which we describe in section 5.

As an illustrative example, consider the simple case

$$E = \{(i,n),\ (n,i),\ (i,i)\ : i = 1, 2, \ldots, n\},$$

i.e., the case where each $\boldsymbol{A}_p$ has possible nonzero entries only in its nth row, its nth column and its diagonal. Let S_r $(r = 1, 2, \ldots, \ell)$ be a partition of $\{1, 2, \ldots, n-1\}$, *i.e.*, $\bigcup_{r=1}^{\ell} S_r = \{1, 2, \ldots, n-1\}$ and $S_r \cap S_s = \emptyset$ $(1 \leq r < s \leq \ell)$. Let $C_r = S_r \cup \{n\}$ $(r = 1, 2, \ldots, \ell)$ and $F = \bigcup_{r=1}^{\ell} C_r \times C_r$. Then conditions (i) and (ii) hold (we will discuss more general cases in detail in section 3). In this case, we obtain the problem below which is equivalent to the SDP (1.1):

$$
\left.
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{(i,j)\in F} [\boldsymbol{A}_0]_{ij} X_{ij} \\
\text{subject to} & \displaystyle\sum_{(i,j)\in F} [\boldsymbol{A}_p]_{ij} X_{ij} = b_p \ (p = 1, 2, \ldots, m), \\
& \begin{pmatrix} \boldsymbol{X}_{S_r S_r} & \boldsymbol{X}_{S_r n} \\ \boldsymbol{X}_{n S_r} & X_{nn} \end{pmatrix} \in \mathcal{S}_+^{C_r} \ (r = 1, 2, \ldots, \ell).
\end{array}
\right\}
$$

Since the entry X_{nn} is involved commonly in all the ℓ positive semidefinite constraints above, we need to introduce additional $\ell - 1$ variables U_{rr} $(r = 1, 2, \ldots, \ell - 1)$ to rewrite the problem above as a standard SDP. Consequently, we obtain an SDP

$$
\left.
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{(i,j)\in F} [\boldsymbol{A}_0]_{ij} X_{ij} \\
\text{subject to} & \displaystyle\sum_{(i,j)\in F} [\boldsymbol{A}_p]_{ij} X_{ij} = b_p \ (p = 1, 2, \ldots, m), \\
& \begin{pmatrix} \boldsymbol{X}_{S_r S_r} & \boldsymbol{X}_{S_r n} \\ \boldsymbol{X}_{n S_r} & U_{rr} \end{pmatrix} \in \mathcal{S}_+^{C_r} \ (r = 1, 2, \ldots, \ell - 1), \\
& \begin{pmatrix} \boldsymbol{X}_{S_\ell S_\ell} & \boldsymbol{X}_{S_\ell n} \\ \boldsymbol{X}_{n S_\ell} & X_{nn} \end{pmatrix} \in \mathcal{S}_+^{C_\ell}, \\
& U_{rr} = X_{nn} \ (r = 1, 2, \ldots, \ell - 1).
\end{array}
\right\}
$$

Thus we have converted the SDP (1.1) having an $n \times n$ positive semidefinite matrix variable $\boldsymbol{X}$ into an SDP having ℓ smaller size positive semidefinite matrix variables. We can use several software packages [4, 6, 28, etc.] of primal-dual interior-point methods incorporating a standard block-diagonal matrix data structure to solve this type of SDPs quite efficiently.

The conversion mentioned above considerably reduces the size of the positive semidefinite matrix variables when we take a larger ℓ and smaller size S_r $(r = 1, 2, \ldots, \ell)$. Intuitively, it becomes easier to solve the resulting SDP as the size of each positive semidefinite matrix variable $\boldsymbol{X}_{C_r C_r}$ gets smaller. However, it is also necessary to take into account the increase in the number of equality constraints. For example, if we take $\ell = n - 1$ and $S_r = \{r\}$ $(r = 1, 2, \ldots, n-1)$, then the conversion yields $n - 2$ additional equality constraints of the form $U_{rr} = X_{nn}$ $(r = 1, 2, \ldots, n-2)$, which, in turn, causes an increase in the CPU time to solve the system of linear equations $\boldsymbol{B} d\boldsymbol{z} = \boldsymbol{s}$. Therefore we need to balance two factors, reduction in the sizes of positive semidefinite matrix variables, and increase in the number of equality constraints. We will present more details on the conversion method in section 4. Some numerical examples are presented in section 7 which show how the balance of the two factors is important.

In section 5, we propose a primal-dual interior-point method based on positive definite matrix completion which we can directly apply to the primal-dual pair of SDPs (1.1) and (1.2) without increasing the number of equality constraints. The method generates a sequence $\{(\boldsymbol{X}^k, \boldsymbol{Y}^k, \boldsymbol{z}^k)\}$ of interior-points of the primal-dual pair of SDPs (1.1) and (1.2), but we perform all matrix computations in $\boldsymbol{X}^k$, $\boldsymbol{Y}^k$ and their inverses essentially in partial matrices with entries specified in F by fully utilizing the sparse clique-factorization formula for the maximum-determinant positive definite matrix completion given in section 2.2. This method is more promising than the conversion method given in section 4. A practical implementation of this method and its numerical experiments will be the main subjects of Part II [22] of this article.

Section 6 discusses linear transformations in the primal and the dual spaces which enhance the aggregate sparsity pattern of data matrices of SDPs. In particular, we will show that an appropriate congruence transformation in the primal space makes it possible for us to apply our methods given in sections 4 and 5 to SDP relaxations of the graph equi-partition problem and the maximum clique problem.

The reader can read sections 4, 5, and 6 independently.

Finally, section 7 is devoted to some numerical examples on the conversion method given in section 4.

2. Theoretical background on positive semidefinite matrix completion.

In this section, we review some fundamental results about the positive semidefinite matrix completion problem.

2.1. Chordal graph.

Some graph-theoretic concepts needed in the subsequent discussion are introduced here. Particular emphasis is laid on chordal graphs.

We denote by $G(V,E)$ an undirected graph with the vertex set V and the edge set $E \subseteq V \times V$, where $(u,v) \in V \times V$ is identified with $(v,u) \in V \times V$. It is assumed throughout this paper that a graph has no loops, that is, $(v,v) \notin E$ for any $v \in V$. Two vertices $u,v \in V$ are said to be *adjacent* if $(u,v) \in E$. The set of the vertices adjacent to $v \in V$ is denoted by $\mathrm{Adj}(v) = \{u \in V : (u,v) \in E\}$.

A graph is called *complete* if every pair of vertices are adjacent. For a subset V' of the vertex set V of a graph $G(V,E)$, the *induced subgraph* on V' is a graph $G(V',E')$ with the vertex set V' and the edge set $E' = E \cap (V' \times V')$. A *clique* of a graph is an induced subgraph which is complete, and a clique is *maximal* if its vertices do not constitute a proper subset of another clique. In our succeeding discussions, we often call $C \subseteq V$ a clique of $G(V,E)$ whenever it induces a clique of $G(V,E)$. A vertex is called *simplicial* if its adjacent vertices induce a clique.

A graph $G(V,E)$ is said to be *chordal* if every cycle of length ≥ 4 has a chord (an edge joining two nonconsecutive vertices of the cycle). Chordal graphs have been studied extensively in many different contexts. See [3, 10, 18] for the background materials as well as the proofs of the statements given below.

The most fundamental property of a chordal graph is that it has a simplicial vertex, say v_1. Then the subgraph induced on $V \setminus \{v_1\}$ is again chordal, and therefore it has a simplicial vertex, say v_2. By repeating this, we can construct an ordering of the vertices, $(v_1, v_2, \ldots, v_n)$ (where $n = |V|$) such that $\mathrm{Adj}(v_i) \cap \{v_{i+1}, v_{i+2}, \ldots, v_n\}$ induces a clique for each $i = 1, 2, \ldots, n-1$. Such an ordering of the vertices is called a *perfect elimination ordering*. The existence of a perfect elimination ordering characterizes chordality, as follows.

THEOREM 2.1 (Fulkerson and Gross [8]). *A graph is chordal if and only if it has a perfect elimination ordering.*

It is known that a perfect elimination ordering of a chordal graph can be found efficiently, in linear time in the number of vertices and edges of the graph [26].

Maximal cliques of a chordal graph can be enumerated easily with reference to a perfect elimination ordering $(v_1, v_2, \ldots, v_n)$. A maximal clique containing v_1 is unique, which is given by $\{v_1\} \cup \mathrm{Adj}(v_1)$, and a maximal clique not containing v_1 is a maximal clique of the subgraph induced on $\{v_2, v_3, \dots, v_n\}$. Then it follows that the family $\{C_r \subseteq V : r = 1, 2, \dots, \ell\}$ of (the vertex sets of) the maximal cliques is given as the maximal members of $\{v_i\} \cup (\mathrm{Adj}(v_i) \cap \{v_{i+1}, v_{i+2}, \dots, v_n\})$ for $i = 1, 2, \dots, n$. More specifically, we have

$$C_r = \{v_i\} \cup (\mathrm{Adj}(v_i) \cap \{v_{i+1}, v_{i+2}, \dots, v_n\})$$

for $i = \min\{j : v_j \in C_r\}$. This shows, in particular, that the number ℓ of maximal cliques is bounded by n.

It is known that the maximal cliques can be indexed in such a way that for each $r = 1, 2, \dots, \ell - 1$ it holds that

$$\exists s \geq r+1 : \quad C_r \cap (C_{r+1} \cup C_{r+2} \cup \cdots \cup C_\ell) \subsetneq C_s. \tag{2.1}$$

The property (2.1) is called the *running intersection property.* An ordering of the maximal cliques satisfying the running intersection property (2.1) induces a perfect elimination ordering of the vertices. Note first that $S_1 = C_1 \backslash (C_2 \cup C_3 \cup \cdots \cup C_\ell)$ is nonempty and all the vertices in S_1 are simplicial. This means that we can start a perfect elimination ordering by numbering the vertices in S_1 with $1, 2, \cdots, |S_1|$. For each $r = 1, 2, \dots, \ell$ in general we number the vertices in $S_r = C_r \backslash (C_{r+1} \cup \cdots \cup C_\ell)$ with $\sum_{s=1}^{r-1} |S_s| + 1, \sum_{s=1}^{r-1} |S_s| + 2, \cdots, \sum_{s=1}^{r-1} |S_s| + |S_r|$. We can thus obtain a perfect elimination ordering of the vertices, in which the vertices in S_r are given consecutive numbers for each r. Throughout this paper, we assume that $(v_1, v_2, \dots, v_n)$ is a perfect elimination ordering induced in this way from an ordering of the maximal cliques satisfying the running intersection property (2.1).

The structure of the family of maximal cliques can be represented most conveniently in terms of a tree, called *clique tree*, of which the vertices are maximal cliques. In particular, the ordering of the maximal cliques with the running intersection property (2.1) can be represented by an orientation (of the edges) of the clique tree to a rooted tree. The use of clique trees will be discussed in Part II of this article where the implementation issues are treated.

In numerical linear algebra, chordal graphs have been studied in relation to the Gaussian elimination (Cholesky factorization) of sparse positive definite matrices. Given a positive definite matrix $\boldsymbol{X}$, we consider a graph $G(V, E)$ that represents the sparsity pattern of the matrix $\boldsymbol{X}$. Namely, V is the set of row/column indices and $E = \{(i, j) : X_{ij} \neq 0, i \neq j\}$. Let $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{L}^T$ be the Cholesky factorization, where $\boldsymbol{L}$ is a lower-triangular matrix. The sparsity pattern of $\boldsymbol{L}$ can be represented similarly by a graph $G(V, F)$ defined by $F = \{(i, j) : L_{ij} \neq 0 \text{ or } L_{ji} \neq 0,\ i \neq j\}$. Under the generic assumption that no numerical cancellations occur in the elimination process, the sparsity pattern of $\boldsymbol{L}$ is determined by that of the matrix $\boldsymbol{X}$, and accordingly the graph $G(V, F)$ is determined by the graph $G(V, E)$ and the ordering of the vertices. In particular, we have $F \supseteq E$, where the added edges (belonging to $F \setminus E$) correspond to the fill-in. Moreover, the graph $G(V, F)$ is a chordal graph by Theorem 2.1. Given a graph $G(V, E)$ in general (not necessarily chordal), we say that a graph $G(V, F)$ is a *chordal extension* of $G(V, E)$ if $G(V, F)$ is chordal and $F \supseteq E$.

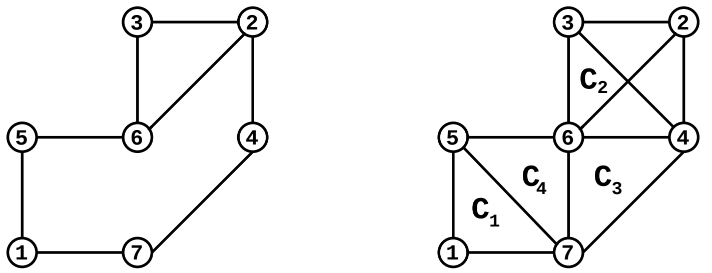
EXAMPLE 2.2. *The chordal extension is illustrated here. Let* $\boldsymbol{X}$ *be a* 7×7 *positive* *definite symmetric matrix with the nonzero pattern given by*

$$\boldsymbol{X} = \left(\begin{array}{ccccccc} e & & & & e & & e \\ & e & e & e & & e & \\ & e & e & & & e & \\ & e & & e & & & e \\ e & & & & e & e & \\ & e & e & & e & e & \\ e & & & e & & & e \end{array} \right),$$

where e denotes nonzero entries. The associated graph $G(V, E)$ is depicted in Figure 2.1(a), where $V = \{1, 2, \ldots, 7\}$. The Cholesky factorization of $\boldsymbol{X}$ yields fill-in at $(i, j) = (5, 7)$, $(3, 4), (4, 6), (6, 7)$, and the chordal extension $G(V, F)$ is shown in Figure 2.1(b). The matrix pattern for $G(V, F)$ is

$$\tilde{\boldsymbol{X}} = \left(\begin{array}{ccccccc} e & & & & e & & e \\ & e & e & e & & e & \\ & e & e & f & & e & \\ & e & f & e & & f & e \\ e & & & & e & e & f \\ & e & e & f & e & e & f \\ e & & & e & f & f & e \end{array} \right),$$

where f denotes fill-in. The natural ordering $(1, 2, \ldots, 6, 7)$ is a perfect elimination ordering of the chordal graph $G(V, F)$, whereas $(7, 6, \ldots, 2, 1)$ is not. The perfect elimination ordering is not unique; for instance, $(3, 2, 4, 6, 7, 1, 5)$ is another perfect elimination ordering. The chordal graph $G(V, F)$ has four maximal cliques, $C_1 = \{1, 5, 7\}$, $C_2 = \{2, 3, 4, 6\}$, $C_3 = \{4, 6, 7\}$, $C_4 = \{5, 6, 7\}$. Note that the running intersection property (2.1) holds with respect to this ordering of the maximal cliques.



(a) graph $G(V, E)$ *(b) chordal extension $G(V, F)$*

FIG. 2.1. *Graph $G(V, E)$ and its chordal extension $G(V, F)$; $(1, 2, \ldots, 7)$ and $(3, 2, 4, 6, 7, 1, 5)$ are perfect elimination orderings for $G(V, F)$, and C_1, C_2, C_3 and C_4 are maximal cliques of $G(V, F)$.*

The fill-in in the Cholesky factorization, and hence the resulting chordal extension $G(V, F)$, depend on the ordering of the row/column indices. It is a major issue in sparse matrix computation to find a permutation matrix $\boldsymbol{P}$ (representing an ordering) such that $\boldsymbol{P}\boldsymbol{X}\boldsymbol{P}^T$ yields as little fill-in as possible. Using the graph terminology this amounts to finding a sparse chordal extension of a given graph, since any minimal chordal extension $G(V, F)$ of $G(V, E)$ can be obtained through the Cholesky factorization process for some ordering. The problem of finding a permutation matrix $\boldsymbol{P}$

that results in the minimum number of fill-in, or equivalently, the problem of finding a chordal extension with the minimum number of edges, is known to be NP-complete. Several heuristic algorithms such as the minimum-degree ordering and the nested dissection have been proposed for this problem [9]. In the most favorable case where the given graph $G(V, E)$ is chordal, the perfect elimination ordering yields the Cholesky factorization with no fill-in.

2.2. Positive semidefinite matrix completion. A *partial symmetric matrix* means a symmetric matrix in which only part of the entries are specified. More precisely, an $n \times n$ partial symmetric matrix $\bar{\boldsymbol{X}}$ is given as a collection of real numbers $(\bar{X}_{ij} = \bar{X}_{ji} : (i,j) \in F)$ for some $F \subseteq V \times V$ such that $(i,j) \in F$ if and only if $(j,i) \in F$, where $V = \{1, 2, \ldots, n\}$. A *completion* of a partial symmetric matrix $\bar{\boldsymbol{X}}$ means a symmetric matrix $\boldsymbol{X}$ (of the same size) such that $X_{ij} = \bar{X}_{ij}$ for $(i,j) \in F$. The *positive (semi)definite matrix completion problem* is to find a positive (semi)definite matrix which is a completion of a given partial symmetric matrix. See [14, 17] for surveys on matrix completion problems.

In considering this problem we may assume without loss of generality that the diagonal entries are all specified, *i.e.*,

$$F \supseteq \{(i,i) : i = 1, 2, \ldots, n\}, \tag{2.2}$$

since unspecified diagonal entries, if any, may be given sufficiently large values to realize positive (semi)definiteness. We adopt the convention (2.2) throughout this section.

We use the following notation:

- $\mathcal{S}^n(F, ?)$: the set of $n \times n$ partial symmetric matrices with entries specified in F,
- $\mathcal{S}^n_+(F, ?)$: the set of $n \times n$ partial symmetric matrices with specified entries in F which can be completed to positive semidefinite symmetric matrices; *i.e.*, $\mathcal{S}^n_+(F, ?) = \{\bar{\boldsymbol{X}} \in \mathcal{S}^n(F, ?) : \exists \boldsymbol{X} \in \mathcal{S}^n_+, \bar{X}_{ij} = X_{ij} \text{ for } (i,j) \in F\}$,
- $\mathcal{S}^n_{++}(F, ?)$: the set of $n \times n$ partial symmetric matrices with specified entries in F which can be completed to positive definite symmetric matrices; *i.e.*, $\mathcal{S}^n_{++}(F, ?) = \{\bar{\boldsymbol{X}} \in \mathcal{S}^n(F, ?) : \exists \boldsymbol{X} \in \mathcal{S}^n_{++}, \bar{X}_{ij} = X_{ij} \text{ for } (i,j) \in F\}$,
- $\mathcal{S}^n(F, 0)$: the set of $n \times n$ symmetric matrices with vanishing entries outside F; *i.e.*, $\mathcal{S}^n(F, 0) = \{\boldsymbol{X} \in \mathcal{S}^n : X_{ij} = 0 \text{ if } (i,j) \notin F\}$,
- $\mathcal{S}^n_+(F, 0)$: the set of $n \times n$ positive semidefinite symmetric matrices with vanishing entries outside F; *i.e.*, $\mathcal{S}^n_+(F, 0) = \mathcal{S}^n_+ \cap \mathcal{S}^n(F, 0) = \{\boldsymbol{X} \in \mathcal{S}^n_+ : X_{ij} = 0 \text{ if } (i,j) \notin F\}$,
- $\mathcal{S}^n_{++}(F, 0)$: the set of $n \times n$ positive definite symmetric matrices with vanishing entries outside F; *i.e.*, $\mathcal{S}^n_{++}(F, 0) = \mathcal{S}^n_{++} \cap \mathcal{S}^n(F, 0) = \{\boldsymbol{X} \in \mathcal{S}^n_{++} : X_{ij} = 0 \text{ if } (i,j) \notin F\}$,
- $\mathcal{S}^C, \mathcal{S}^C_+, \mathcal{S}^C_{++}$: the sets of $\sharp C \times \sharp C$ symmetric matrices, positive semidefinite symmetric matrices, positive definite symmetric matrices, respectively, with rows and columns indexed by $C \subseteq V$, where $\sharp C$ means the number of elements of C.

For $E, F \subseteq V \times V$ in general, we define

$$F^\circ = F \setminus \{(i,i) : i = 1, 2, \ldots, n\}, \tag{2.3}$$

$$E^\bullet = E \cup \{(i,i) : i = 1, 2, \ldots, n\}. \tag{2.4}$$

Then, the structure F of a partial symmetric matrix can be represented by a graph

$G(V, E)$ with $E = F^\circ$. Conversely, a graph $G(V, E)$ is associated with the class of partial symmetric matrices $\mathcal{S}^n(E^\bullet, ?)$.

Suppose we are given a partial symmetric matrix $\bar{\boldsymbol{X}} \in \mathcal{S}^n(F, ?)$, and let $G(V, E)$ be the associated graph, where $E = F^\circ$. Denote by $\{C_r \subseteq V : r = 1, 2, \ldots, \ell\}$ the family of all maximal cliques of $G(V, E)$. An obvious necessary condition for $\bar{\boldsymbol{X}}$ to have a positive semidefinite matrix completion is that each $\bar{\boldsymbol{X}}_{C_r C_r}$ is positive semidefinite, *i.e.*,

$$\bar{\boldsymbol{X}}_{C_r C_r} \in \mathcal{S}_+^{C_r} \qquad (r = 1, 2, \ldots, \ell), \tag{2.5}$$

where it is noted that all the entries of the submatrix $\bar{\boldsymbol{X}}_{C_r C_r}$ are specified. Similarly, an obvious necessary condition for $\bar{\boldsymbol{X}}$ to have a positive definite matrix completion is that each $\bar{\boldsymbol{X}}_{C_r C_r}$ is positive definite, *i.e.*,

$$\bar{\boldsymbol{X}}_{C_r C_r} \in \mathcal{S}_{++}^{C_r} \qquad (r = 1, 2, \ldots, \ell). \tag{2.6}$$

We refer to (2.5) and (2.6) as the *clique-PSD condition* and the *clique-PD condition*, respectively.

The following two theorems are most fundamental concerning the positive (semi)definite matrix completion problem.

THEOREM 2.3 (Gron, Johnson, Sá and Wolkowicz [11, Theorem 7]). *Let $G(V, E)$ be a graph.*

(i) Any partial symmetric matrix $\bar{\boldsymbol{X}} \in \mathcal{S}^n(E^\bullet, ?)$ satisfying the clique-PSD condition (2.5) can be completed to a positive semidefinite symmetric matrix $\boldsymbol{X}$ if and only if $G(V, E)$ is chordal.

(ii) Any partial symmetric matrix $\bar{\boldsymbol{X}} \in \mathcal{S}^n(E^\bullet, ?)$ satisfying the clique-PD condition (2.6) can be completed to a positive definite symmetric matrix $\boldsymbol{X}$ if and only if $G(V, E)$ is chordal.

THEOREM 2.4 (Gron, Johnson, Sá and Wolkowicz [11, Theorem 2]). *Suppose that a partial symmetric matrix $\bar{\boldsymbol{X}} \in \mathcal{S}^n(F, ?)$ has a positive definite matrix completion. Then there exists a unique positive definite matrix completion $\boldsymbol{X} = \hat{\boldsymbol{X}}$ that maximizes the determinant, i.e., such that*

$$\det(\hat{\boldsymbol{X}}) = \max\{\det(\boldsymbol{X}) : \boldsymbol{X} \text{ is a positive definite matrix completion of } \bar{\boldsymbol{X}}\}.$$

Moreover, such $\hat{\boldsymbol{X}}$ is characterized by the condition:

$$[\hat{\boldsymbol{X}}^{-1}]_{ij} = 0 \quad ((i, j) \notin F), \quad \textit{i.e.}, \quad \hat{\boldsymbol{X}}^{-1} \in \mathcal{S}^n(F, 0).$$

We refer to the completion $\hat{\boldsymbol{X}}$ in Theorem 2.4 as the *maximum-determinant positive definite matrix completion* of $\bar{\boldsymbol{X}}$.

The sufficiency part in Theorem 2.3 can be restated in the following form convenient for our subsequent use.

THEOREM 2.5. *Let $G(V, E)$ be a chordal graph.*

(i) A partial symmetric matrix $\bar{\boldsymbol{X}} \in \mathcal{S}^n(E^\bullet, ?)$ can be completed to a positive semidefinite symmetric matrix $\boldsymbol{X}$ if and only if it satisfies the clique-PSD condition (2.5).

(ii) A partial symmetric matrix $\bar{\boldsymbol{X}} \in \mathcal{S}^n(E^\bullet, ?)$ can be completed to a positive definite symmetric matrix $\boldsymbol{X}$ if and only if it satisfies the clique-PD condition (2.6).

In what follows we shall give a concrete expression of the maximum-determinant positive definite matrix completion in case (ii) above. This expression forms the basis of our computational scheme for sparse semidefinite programs, to be described in section 5. Also it serves as a constructive proof of the "if" part in (ii), while the "only if" part is obvious.

We start with a fundamental lemma showing an elementary construction of the maximum-determinant positive definite matrix completion.

LEMMA 2.6. *Let S and T be disjoint nonempty subsets of V and $\bar{\boldsymbol{X}}$ be a partial symmetric matrix with the entries in $(S \times T) \cup (T \times S)$ unspecified, i.e., $\bar{\boldsymbol{X}} \in \mathcal{S}^n(F, ?)$ for $F = (V \times V) \setminus ((S \times T) \cup (T \times S))$. Then $\bar{\boldsymbol{X}}$ admits a positive definite matrix completion if and only if the two submatrices*

$$\begin{pmatrix} \bar{\boldsymbol{X}}_{SS} & \bar{\boldsymbol{X}}_{SU} \\ \bar{\boldsymbol{X}}_{US} & \bar{\boldsymbol{X}}_{UU} \end{pmatrix} \text{ and } \begin{pmatrix} \bar{\boldsymbol{X}}_{UU} & \bar{\boldsymbol{X}}_{UT} \\ \bar{\boldsymbol{X}}_{TU} & \bar{\boldsymbol{X}}_{TT} \end{pmatrix} \tag{2.7}$$

are both positive definite, where $U = V \setminus (S \cup T)$. If this is the case, the matrix $\hat{\boldsymbol{X}}$ defined by

$$\hat{\boldsymbol{X}} = \begin{pmatrix} \bar{\boldsymbol{X}}_{SS} & \bar{\boldsymbol{X}}_{SU} & \bar{\boldsymbol{X}}_{SU}\bar{\boldsymbol{X}}_{UU}^{-1}\bar{\boldsymbol{X}}_{UT} \\ \bar{\boldsymbol{X}}_{US} & \bar{\boldsymbol{X}}_{UU} & \bar{\boldsymbol{X}}_{UT} \\ \bar{\boldsymbol{X}}_{TU}\bar{\boldsymbol{X}}_{UU}^{-1}\bar{\boldsymbol{X}}_{US} & \bar{\boldsymbol{X}}_{TU} & \bar{\boldsymbol{X}}_{TT} \end{pmatrix} \tag{2.8}$$

has the following properties: (i) *$\hat{\boldsymbol{X}}$ is positive definite,* (ii) $(\hat{\boldsymbol{X}}^{-1})_{ST} = \boldsymbol{O}$, (iii) *$\hat{\boldsymbol{X}}$ is the unique maximizer of the determinant among all positive definite matrix completions of $\bar{\boldsymbol{X}}$. Here we adopt the convention: $\bar{\boldsymbol{X}}_{SU}\bar{\boldsymbol{X}}_{UU}^{-1}\bar{\boldsymbol{X}}_{UT} = \boldsymbol{O}$ and $\bar{\boldsymbol{X}}_{TU}\bar{\boldsymbol{X}}_{UU}^{-1}\bar{\boldsymbol{X}}_{US} = \boldsymbol{O}$ if $U = \emptyset$.*

Proof. The necessity of the positive definiteness of the two submatrices in (2.7) is obvious. For the sufficiency, we note

$$\begin{aligned} &\begin{pmatrix} \boldsymbol{I} & -\bar{\boldsymbol{X}}_{SU}\bar{\boldsymbol{X}}_{UU}^{-1} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I} \end{pmatrix} \hat{\boldsymbol{X}} \begin{pmatrix} \boldsymbol{I} & \boldsymbol{O} & \boldsymbol{O} \\ -\bar{\boldsymbol{X}}_{UU}^{-1}\bar{\boldsymbol{X}}_{US} & \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I} \end{pmatrix} \\ &= \begin{pmatrix} \bar{\boldsymbol{X}}_{SS} - \bar{\boldsymbol{X}}_{SU}\bar{\boldsymbol{X}}_{UU}^{-1}\bar{\boldsymbol{X}}_{US} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \bar{\boldsymbol{X}}_{UU} & \bar{\boldsymbol{X}}_{UT} \\ \boldsymbol{O} & \bar{\boldsymbol{X}}_{TU} & \bar{\boldsymbol{X}}_{TT} \end{pmatrix}, \end{aligned} \tag{2.9}$$

in which

$$\bar{\boldsymbol{X}}_{SS} - \bar{\boldsymbol{X}}_{SU}\bar{\boldsymbol{X}}_{UU}^{-1}\bar{\boldsymbol{X}}_{US} \in \mathcal{S}_{++}^{S}$$

by the positive definiteness of the first matrix in (2.7). Hence (i) follows. Let $\boldsymbol{D}$ denote the matrix on the right-hand side of (2.9). Then (ii) can be shown as

$$(\hat{\boldsymbol{X}}^{-1})_{ST} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{O} & \boldsymbol{O} \end{pmatrix} \boldsymbol{D}^{-1} \begin{pmatrix} \boldsymbol{O} \\ \boldsymbol{O} \\ \boldsymbol{I} \end{pmatrix} = \boldsymbol{O}.$$

Finally, (iii) follows from (ii) by Theorem 2.4. □

A recursive application of Lemma 2.6 in accordance with the perfect elimination ordering yields an explicit construction of the positive definite matrix completion in Theorem 2.5(ii). For simplicity of notation, let us assume that $(1, 2, \ldots, n)$ is a perfect elimination ordering of the chordal graph $G(V, E)$. Suppose (recursively) that the $(n-1)\times(n-1)$ submatrix corresponding to $\{2,3,\dots,n\}$ has been completed to a positive definite matrix. Then we can apply Lemma 2.6 with $S=\{1\}$, $T=\{i : i>1\}\setminus \mathrm{Adj}(1)$ and $U=\mathrm{Adj}(1)$, to obtain a positive definite matrix completion of the whole matrix. Note that the first matrix in (2.7) is positive definite by the assumed clique-PD condition (2.6), and the second by the recursive assumption. Let $\hat{\boldsymbol{X}}$ be the completion obtained by the recursive application of this procedure.

We shall show in Lemma 2.7 below that the matrix $\hat{\boldsymbol{X}}$ constructed above is indeed the maximum-determinant positive definite matrix completion of $\bar{\boldsymbol{X}}$, and moreover, that it admits a factorization

$$\boldsymbol{P}\hat{\boldsymbol{X}}\boldsymbol{P}^T = \boldsymbol{L}_1^T\boldsymbol{L}_2^T\cdots\boldsymbol{L}_{n-1}^T\boldsymbol{D}\boldsymbol{L}_{n-1}\cdots\boldsymbol{L}_2\boldsymbol{L}_1 \tag{2.10}$$

with "sparse" triangular matrices $\boldsymbol{L}_k$ $(k=1,2,\dots,n-1)$ and a positive definite diagonal matrix $\boldsymbol{D}$, where $\boldsymbol{P}=\boldsymbol{I}$ under our tentative assumption that $(1,2,\dots,n)$ is a perfect elimination ordering. We define

$$U_k = \mathrm{Adj}(k)\cap\{i : i>k\} \qquad (k=1,2,\dots,n). \tag{2.11}$$

It follows from the repeated use of (2.9) that $\boldsymbol{L}_k$ is a lower triangular matrix

$$\boldsymbol{L}_k = \begin{pmatrix} \boldsymbol{I}_{k-1} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 & 0 & \cdots & 0 \\ \boldsymbol{0}^T & [\boldsymbol{L}_k]_{k+1,k} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0}^T & [\boldsymbol{L}_k]_{nk} & 0 & \cdots & 1 \end{pmatrix} \tag{2.12}$$

with unit diagonal entries $[\boldsymbol{L}_k]_{ii}=1$ $(i=1,2,\dots,n)$ and other possible nonzero entries at $\{(i,k) : i\in U_k\}$ in the kth column; to be specific,

$$[\boldsymbol{L}_k]_{ij} = \begin{cases} 1 & (i=j) \\ [\bar{\boldsymbol{X}}_{U_kU_k}^{-1}\bar{\boldsymbol{X}}_{U_kk}]_{ik} & (i\in U_k,\, j=k) \\ 0 & (\text{otherwise}) \end{cases} \tag{2.13}$$

for $k=1,2,\dots,n-1$. Expressions of the diagonal entries of $\boldsymbol{D}$ are also known from (2.9) as

$$D_{kk} = \begin{cases} \bar{X}_{kk} - \bar{\boldsymbol{X}}_{kU_k}\bar{\boldsymbol{X}}_{U_kU_k}^{-1}\bar{\boldsymbol{X}}_{U_kk} & (k=1,2,\dots,n-1), \\ \bar{X}_{nn} & (k=n). \end{cases} \tag{2.14}$$

We have $D_{kk}>0$ for $k=1,2,\dots,n$ by the clique-PD condition (2.6). Henceforth we refer to (2.10) as the *sparse factorization formula.*

It is mentioned that the sparse factorization formula (2.10) of $\hat{\boldsymbol{X}}$ depends on the perfect elimination ordering, represented by $\boldsymbol{P}$, used in the construction, whereas $\hat{\boldsymbol{X}}$ itself is independent of it because of the uniqueness of the maximum-determinant positive definite matrix completion. Note also that the factorization (2.10) is equivalent to

$$\boldsymbol{P}\hat{\boldsymbol{X}}^{-1}\boldsymbol{P}^T = \boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdots\boldsymbol{L}_{n-1}^{-1}\boldsymbol{D}^{-1}\boldsymbol{L}_{n-1}^{-T}\cdots\boldsymbol{L}_2^{-T}\boldsymbol{L}_1^{-T}, \tag{2.15}$$

which is the product form of the (LDL^T) Cholesky factorization of $\boldsymbol{P}\hat{\boldsymbol{X}}^{-1}\boldsymbol{P}^T$.

LEMMA 2.7. *Let $G(V,E)$ be a chordal graph, and $\bar{\boldsymbol{X}} \in \mathcal{S}^n(E^\bullet, ?)$ be a partial symmetric matrix satisfying the clique-PD condition (2.6). Let $\boldsymbol{P}$ be a permutation matrix representing a perfect elimination ordering of $G(V,E)$ in such a way that $(1,2,\ldots,n)$ is a perfect elimination ordering for $\boldsymbol{P}\bar{\boldsymbol{X}}\boldsymbol{P}^T$. Then the maximum-determinant positive definite matrix completion $\hat{\boldsymbol{X}}$ of $\bar{\boldsymbol{X}}$ can be expressed in terms of the sparse factorization formula (2.10), where $\boldsymbol{L}_k$ is a lower triangular matrix given by (2.12) and (2.13) and $\boldsymbol{D}$ is a positive definite diagonal matrix given by (2.14).*

Proof. The positive definiteness of $\hat{\boldsymbol{X}}$ follows from the factorization formula (2.10) together with the positive definiteness of $\boldsymbol{D}$. For the maximum-determinant property it suffices, by Theorem 2.4, to show that $[\hat{\boldsymbol{X}}^{-1}]_{ij} = 0$ for $(i,j) \notin E^\bullet$. Referring to (2.15) we define $\boldsymbol{M} = \boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdots\boldsymbol{L}_{n-1}^{-1}$, which is a lower triangular matrix with unit diagonal entries. The kth column of $\boldsymbol{M}$ coincides, except for the diagonal entry, with the negative of the kth column of $\boldsymbol{L}_k$. Therefore, $\boldsymbol{M}$ has nonzero off-diagonal entries only at $(i,j) \in E$. Suppose that $[\hat{\boldsymbol{X}}^{-1}]_{ij} \neq 0$ and assume $\boldsymbol{P} = \boldsymbol{I}$ in (2.15). Then $M_{ik} \neq 0$ and $M_{jk} \neq 0$ for some $k \leq \min(i,j)$. Hence $(k,i) \in E^\bullet$ and $(k,j) \in E^\bullet$. This means $(i,j) \in E^\bullet$ because $(1,2,\ldots,n)$ is a perfect elimination ordering. ∎

REMARK 2.8. *Here is a minor remark on the computations of (2.13) and (2.14). For each k, the subset $\{k\} \cup U_k$ induces a clique in $G(V,E)$, and the maximal members of such cliques are exactly the maximal cliques of $G(V,E)$, which are denoted as $\{C_r \subseteq V : r = 1,2,\ldots,\ell\}$. Moreover, for each r, those subsets U_k which are contained in C_r form a nested family; define $K_r = \{k : U_k \subseteq C_r\}$. Hence the Cholesky factorizations of $\bar{\boldsymbol{X}}_{U_kU_k}$ for all $k \in K_r$ needed in the computations in (2.13) and (2.14) are embedded in the Cholesky factorization of $\bar{\boldsymbol{X}}_{C_rC_r}$ with an appropriate ordering.*

The sparse factorization formula (2.10) can be made conceptually more transparent and practically more efficient if it is constructed with reference to an ordering of maximal cliques rather than to a perfect elimination ordering of vertices. Let $(C_1, C_2, \ldots, C_\ell)$ be an ordering of maximal cliques that enjoys the running intersection property (2.1). A similar argument based on Lemma 2.6 yields a variant of the sparse factorization formula of the form

$$\boldsymbol{P}\hat{\boldsymbol{X}}\boldsymbol{P}^T = \boldsymbol{L}_1^T\boldsymbol{L}_2^T\cdots\boldsymbol{L}_{\ell-1}^T\boldsymbol{D}\boldsymbol{L}_{\ell-1}\cdots\boldsymbol{L}_2\boldsymbol{L}_1, \tag{2.16}$$

where $\boldsymbol{L}_r$ $(r = 1,2,\ldots,\ell-1)$ are "sparse" triangular matrices and $\boldsymbol{D}$ is a positive definite block-diagonal matrix consisting of ℓ diagonal blocks. We will call (2.16) the *sparse clique-factorization formula*. The concrete expressions of $\boldsymbol{L}_r$ $(r = 1,2,\ldots,\ell-1)$ and $\boldsymbol{D}$ can be obtained as straightforward extensions of (2.12) $\sim$ (2.14). Namely, define

$$\begin{array}{rcll} S_r & = & C_r \backslash (C_{r+1} \cup C_{r+2} \cup \cdots \cup C_\ell) & (r = 1,2,\ldots,\ell), \\ U_r & = & C_r \cap (C_{r+1} \cup C_{r+2} \cup \cdots \cup C_\ell) & (r = 1,2,\ldots,\ell). \end{array}$$

Then the factors in (2.16) are given by

$$[\boldsymbol{L}_r]_{ij} = \begin{cases} 1 & (i = j) \\ [\bar{\boldsymbol{X}}_{U_rU_r}^{-1}\bar{\boldsymbol{X}}_{U_rS_r}]_{ij} & (i \in U_r, j \in S_r) \\ 0 & (\text{otherwise}) \end{cases} \tag{2.17}$$

for $r = 1,2,\ldots,\ell-1$, and

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{D}_{S_1S_1} & & & \\ & \boldsymbol{D}_{S_2S_2} & & \\ & & \ddots & \\ & & & \boldsymbol{D}_{S_\ell S_\ell} \end{pmatrix} \tag{2.18}$$

with

$$\boldsymbol{D}_{S_rS_r} = \begin{cases} \bar{\boldsymbol{X}}_{S_rS_r} - \bar{\boldsymbol{X}}_{S_rU_r}\bar{\boldsymbol{X}}_{U_rU_r}^{-1}\bar{\boldsymbol{X}}_{U_rS_r} & (r = 1, 2, \dots, \ell - 1), \\ \bar{\boldsymbol{X}}_{S_\ell S_\ell} & (r = \ell). \end{cases} \tag{2.19}$$

It should be remarked that we can compute all nonzero submatrices $\bar{\boldsymbol{X}}_{U_rU_r}^{-1}\bar{\boldsymbol{X}}_{U_rS_r}$ and $\boldsymbol{D}_{S_rS_r}$ above in parallel although we need an induction argument to derive the sparse clique-factorization formula (2.16).

3. Chordal extension of aggregate sparsity pattern. In this section, we apply the discussions given in the previous section to the standard equality form SDP (1.1). Let E denote the aggregate sparsity pattern of the data matrices $\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2, \dots, \boldsymbol{A}_m$ given in (1.4). We first choose a chordal extension $G(V, F^\circ)$ of the graph $G(V, E^\circ)$. Let $\{C_r \subseteq V : r = 1, 2, \dots, \ell\}$ be the family of maximal cliques of the graph $G(V, F^\circ)$, where $F \supseteq E$. Then (i) the values of the objective and constraint linear functions $\boldsymbol{A}_p \bullet \boldsymbol{X}$ $(p = 0, 1, \dots, m)$ of the SDP (1.1) are determined by X_{ij} $((i, j) \in F)$ regardless of X_{ij} $((i, j) \notin F)$, and (ii) any $\boldsymbol{X} \in \mathcal{S}^n(F, ?)$ has a positive semidefinite (or positive definite, respectively) matrix completion if and only if the submatrices $\boldsymbol{X}_{C_rC_r}$ $(r = 1, 2, \dots, \ell)$ are positive semidefinite (or positive definite, respectively) — the clique-PSD condition (2.5) (or the clique-PD condition (2.6), respectively). Therefore we can replace the constraint and the objective function $\boldsymbol{A}_0 \bullet \boldsymbol{X}$ of the SDP (1.1) by the constraint

$$\sum_{(i,j)\in F} [\boldsymbol{A}_p]_{ij} X_{ij} = b_p \ (p = 1, 2, \dots, m) \ \text{ and } \ \boldsymbol{X}_{C_rC_r} \in \mathcal{S}_+^{C_r} \ (r = 1, 2, \dots, \ell) \tag{3.1}$$

and the objective function $\sum_{(i,j)\in F} [\boldsymbol{A}_0]_{ij} X_{ij}$, respectively. More precisely, if $\boldsymbol{X} = \bar{\boldsymbol{X}} \in \mathcal{S}^n$ satisfies the constraint of (1.1) then the partial symmetric matrix $\boldsymbol{X}' \in \mathcal{S}^n(F, ?)$ with entries $X'_{ij} = \bar{X}_{ij}$ $((i, j) \in F)$ satisfies (3.1) and their objective values $\boldsymbol{A}_0 \bullet \boldsymbol{X}$ and $\sum_{(i,j)\in F} [\boldsymbol{A}_0]_{ij} X'_{ij}$ coincide with each other. Conversely, any partial symmetric matrix $\boldsymbol{X}' \in \mathcal{S}^n(F, ?)$ satisfying (3.1) has a positive semidefinite matrix completion $\boldsymbol{X} \in \mathcal{S}^n$ that satisfies the constraint of (1.1) and has the same objective value as $\boldsymbol{X}' \in \mathcal{S}_+^n(F, ?)$.

We will propose two methods with the use of (3.1) for solving the SDP (1.1). The first one is a conversion of the SDP (1.1) having a single matrix variable $\boldsymbol{X} \in \mathcal{S}_+^n$ into an SDP having ℓ matrix variables in $\mathcal{S}_+^{C_r}$ $(r = 1, 2, \dots, \ell)$ in section 4. The other is a primal-dual interior-point method based on positive definite matrix completion in section 5. Roughly speaking, matrix operations such as finding the Cholesky factorization of $\boldsymbol{X}$, the minimum eigenvalue of $\boldsymbol{X}$, and matrix-matrix multiplications, are replaced by the corresponding matrix operations on smaller matrices in $\mathcal{S}^{C_r}$ $(r = 1, 2, \dots, \ell)$ in both methods. There are also overheads depending on the maximal cliques C_r $(r = 1, 2, \dots, \ell)$. In particular, the number of additional equality constraints required in the former method is determined by the intersections of two distinct maximal cliques C_r and C_s $(r < s)$, while the amount of arithmetic operations to compute the search direction in the latter method depends not only on the maximal cliques C_r $(r = 1, 2, \dots, \ell)$, but also on the number m of equality constraints and the sparsity pattern of data matrices $\boldsymbol{A}_p$ $(p = 0, 1, 2, \dots, m)$. The effectiveness of both methods relies entirely on a suitable choice of a chordal extension $G(V, F^\circ)$ of the graph $G(V, E^\circ)$. (Through simple numerical examples in section 7, we will see how crucial a better choice of a chordal extension is to the conversion method.) It seems quite difficult, however, to determine (or even define) an "optimal" chordal extension that would minimize the amount of computational work in each method, because various consequences of the use of (3.1), including those mentioned above, are too complicated to be evaluated accurately. In addition, even if we could set up an appropriate objective function to be minimized over the chordal extensions of the graph $G(V, E^\circ)$, such a minimization problem would be a very difficult combinatorial optimization problem.

As we have seen in the previous section, the chordal extension is closely related to the Cholesky factorization. Specifically, the chordal extension that minimizes the total number of edges in $G(V, F^\circ)$ is obtained via the Cholesky factorization of the aggregate sparsity pattern matrix $\boldsymbol{A}$ with the minimum fill-in. Therefore it seems reasonable (or at least attractive) in practice to employ various existing heuristic methods, such as the minimum degree ordering for less fill-in, the (nested) dissection ordering for less fill-in and the reverse Cuthill-McKee ordering for reducing bandwidth, developed for the Cholesky factorization [9]. We briefly illustrate below how we construct a chordal extension $G(V, F^\circ)$ of the graph $G(V, E^\circ)$ using some of those existing methods.

Suppose that we have reordered the row/column indices symmetrically by applying a dissection ordering so that the resulting aggregate sparsity pattern matrix $\boldsymbol{A}$ has the following bordered block-diagonal form:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{S_1S_1} & \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{A}_{S_1S_0} \\ \boldsymbol{O} & \boldsymbol{A}_{S_2S_2} & \cdots & \boldsymbol{O} & \boldsymbol{A}_{S_2S_0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{A}_{S_\ell S_\ell} & \boldsymbol{A}_{S_\ell S_0} \\ \boldsymbol{A}_{S_0S_1} & \boldsymbol{A}_{S_0S_2} & \cdots & \boldsymbol{A}_{S_0S_\ell} & \boldsymbol{A}_{S_0S_0} \end{pmatrix}$$

and

$$E \subseteq \left(\bigcup_{r=1}^{\ell} S_r \times S_r \right) \bigcup \left(\bigcup_{r=0}^{\ell} S_r \times S_0 \right) \bigcup \left(S_0 \times \bigcup_{r=0}^{\ell} S_r \right).$$

Let

$$C_r = S_0 \cup S_r \ (r = 1, 2, \dots, \ell) \ \text{ and } F = \bigcup_{r=1}^{\ell} C_r \times C_r. \tag{3.2}$$

Obviously $E \subseteq F$. We also see that $G(V, F^\circ)$ is a chordal extension of $G(V, E^\circ)$ and that $\{C_r \subseteq V : r = 1, 2, \dots, \ell\}$ forms the family of maximal cliques of $G(V, F^\circ)$. Furthermore, $(1, 2, \dots, n)$ is a perfect elimination ordering, and the running intersection property (2.1) holds for any $s \geq r + 1$.

Another chordal extension can be obtained through the reordering of row/column indices by the reverse Cuthill-McKee ordering that yields the aggregate sparsity pattern matrix $\boldsymbol{A}$ having a small bandwidth;

$$A_{ij} = 0 \ \text{ if } |j - i| > \beta \quad \text{and } E = \{(i, j) \in V \times V : |i - j| \leq \beta\},$$

where β is a small positive integer. In this case, we can take a collection of subsets $C_1, C_2, \dots, C_\ell$ and $F \supseteq E$ such that

$$C_r = \{i \in V : (r-1)\kappa < i \leq \beta + r\kappa\} \ (r = 1, 2, \dots, \ell) \ \text{ and } \ F = \bigcup_{r=1}^{\ell} C_r \times C_r, \tag{3.3}$$

where κ denotes a positive integer and ℓ the smallest positive integer satisfying $\beta+\ell\kappa \geq n$. Then $G(V, F^\circ)$ is a chordal extension of $G(V, E^\circ)$ and $\{C_r \subseteq V : r = 1, 2, \ldots, \ell\}$ forms the family of maximal cliques of $G(V, F^\circ)$. In this case, $(1, 2, \ldots, n)$ is a perfect elimination ordering, and the running intersection property (2.1) holds for $s = r + 1$.

It is not difficult to extend the discussions above to more sophisticated cases where the aggregate sparsity pattern matrix $\boldsymbol{A}$ forms a nested bordered block-diagonal matrix or a bordered band matrix. In our succeeding paper [22], we will discuss more details about how we choose a chordal extension of $G(V, E^\circ)$ in general.

In the remainder of the paper, we assume that

- an appropriate chordal extension $G(V, F^\circ)$ of $G(V, E^\circ)$ and the family $\{C_r \subseteq V : r = 1, 2, \ldots, \ell\}$ of maximal cliques of $G(V, F^\circ)$ are available to us, and
- $(v_1, v_2, \ldots, v_n)$ is a perfect elimination ordering induced from an ordering of the maximal cliques satisfying the running intersection property (2.1).

Hence, in view of the discussions in the previous section, we can factorize the maximum-determinant positive definite matrix completion $\hat{\boldsymbol{X}}$ of each $\bar{\boldsymbol{X}} \in \mathcal{S}^n(F; ?)$ as in the sparse factorization formula (2.10) (and also as in the sparse clique-factorization formula (2.16)), and any $\boldsymbol{Y} \in \mathcal{S}^n_{++}(F; 0)$ is factorized as $\boldsymbol{Y} = \boldsymbol{R}\boldsymbol{R}^T$ for some $n \times n$ lower triangular matrix $\boldsymbol{R}$ without any fill-in. We also know that the number ℓ of maximal cliques of $G(V, F^\circ)$ does not exceed n.

REMARK 3.1. *We also assume in the remainder of the paper that* $(i, i) \in E$ $(i = 1, 2, \ldots, n)$. *Assume, to the contrary, that some* $(i, i) \notin E$, *for example,*

$$(i, i) \notin E \ (i = 1, 2, \ldots, k) \quad \textit{and} \ (j, j) \in E \ (j = k + 1, k + 2, \ldots, n).$$

Then we can rewrite the SDP (1.1) as

$$\left.\begin{array}{ll}
\textit{minimize} & \boldsymbol{A}_0' \bullet \boldsymbol{X}' + 2\displaystyle\sum_{i=1}^{k}\sum_{j=i+1}^{n} [\boldsymbol{A}_0]_{ij} X_{ij}, \\
\textit{subject to} & \boldsymbol{A}_p' \bullet \boldsymbol{X}' + 2\displaystyle\sum_{i=1}^{k}\sum_{j=i+1}^{n} [\boldsymbol{A}_p]_{ij} X_{ij} = b_p \ (p = 1, 2, \ldots, m), \\
& X_{ij} \in \boldsymbol{R} \ (i = 1, 2, \ldots, k, \ i < j \leq n), \\
& \boldsymbol{X}' = \begin{pmatrix} X_{k+1,k+1} & X_{k+1,k+2} & \cdots & X_{k+1,n} \\ X_{k+2,k+1} & X_{k+2,k+2} & \cdots & X_{k+2,n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n,k+1} & X_{n,k+2} & \cdots & X_{nn} \end{pmatrix} \in \mathcal{S}^U_+,
\end{array}\right\}$$

where

$$\boldsymbol{A}_p' = \begin{pmatrix} [\boldsymbol{A}_p]_{k+1,k+1} & [\boldsymbol{A}_p]_{k+1,k+2} & \cdots & [\boldsymbol{A}_p]_{k+1,n} \\ [\boldsymbol{A}_p]_{k+2,k+1} & [\boldsymbol{A}_p]_{k+2,k+2} & \cdots & [\boldsymbol{A}_p]_{k+2,n} \\ \vdots & \vdots & \ddots & \vdots \\ [\boldsymbol{A}_p]_{n,k+1} & [\boldsymbol{A}_p]_{n,k+2} & \cdots & [\boldsymbol{A}_p]_{nn} \end{pmatrix} \in \mathcal{S}^U \ (p = 0, 1, \ldots, m),$$

and $U = \{k+1, k+2, \ldots, n\}$. *In the transformed problem above, none of* $X_{ij} \in \boldsymbol{R}$ $(i = 1, 2, \ldots, k, \ i < j \leq n)$ *are involved in the positive semidefinite constraint* $\boldsymbol{X}' \in \mathcal{S}^U_+$, *and therefore they are free variables. We can easily adapt the methods described in sections 4 and 5 for the SDP (1.1) satisfying the assumption* $(i, i) \in E$ $(i = 1, 2, \ldots, n)$ *to the transformed problem.*

4. Conversion to an SDP having multiple but smaller size positive semidefinite matrix variables. In the previous section, we have shown that the SDP (1.1) is equivalent to the problem of minimizing the objective function $\sum_{(i,j)\in F}[\boldsymbol{A}_0]_{ij}X_{ij}$ over the constraint (3.1). This problem involves less variables and smaller size positive semidefinite constraints than the original SDP (1.1). This feature certainly makes the conversion attractive in practice because such a problem is expected to be solved easier. It should be noted, however, that two distinct positive semidefinite constraints $\boldsymbol{X}_{C_rC_r} \in \mathcal{S}_+^{C_r}$ and $\boldsymbol{X}_{C_sC_s} \in \mathcal{S}_+^{C_s}$ in (3.1) share variables X_{ij} $((i,j) \in (C_r \cap C_s) \times (C_r \cap C_s))$ whenever $C_r \cap C_s \neq \emptyset$. Hence the problem is not a standard SDP. In this section, we show how to convert the problem to a standard SDP to which we can apply interior-point methods, and discuss some advantages and disadvantages of the resulting SDP.

For every $r = 1, 2, \ldots, \ell$, let

$$E_r = \{(i,j) \in C_r \times C_r : (i,j) \in C_s \times C_s \ \text{ for some } s < r \ \}.$$

By definition, $E_1 = \emptyset$, and if $(i,j) \in E_r$ then the positive semidefinite constraint $\boldsymbol{X}_{C_rC_r} \in \mathcal{S}_+^{C_r}$ shares variables X_{ij} $((i,j) \in E_r)$ with the positive semidefinite constraint $\boldsymbol{X}_{C_sC_s} \in \mathcal{S}_+^{C_s}$ for some $s < r$. To make such a pair of dependent positive semidefinite constraints independent, we introduce auxiliary variables U_{ij}^r $((i,j) \in E_r,\ r = 2, 3, \ldots, \ell)$, and we rewrite the constraint (3.1) as

$$\left.\begin{array}{l} \displaystyle\sum_{(i,j)\in F} [\boldsymbol{A}_p]_{ij} X_{ij} = b_p \ (p = 1, 2, \ldots, m), \\ U_{ij}^r = X_{ij} \ ((i,j) \in E_r,\ i \geq j,\ r = 2, 3, \ldots, \ell), \\ \boldsymbol{X}^r \in \mathcal{S}_+^{C_r} \ (r = 1, 2, \ldots, \ell), \end{array}\right\} \tag{4.1}$$

where

$$[\boldsymbol{X}^r]_{ij} = \begin{cases} U_{ij}^r & \text{if } (i,j) \in E_r, \\ X_{ij} & \text{otherwise.} \end{cases}$$

Then we may regard the minimization of the objective function $\sum_{(i,j)\in F}[\boldsymbol{A}_0]_{ij}X_{ij}$ over the constraint (4.1) as a standard SDP. In fact, if we further introduce a block-diagonal symmetric matrix variable of the form

$$\boldsymbol{X}' = \begin{pmatrix} \boldsymbol{X}^1 & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{X}^2 & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{X}^\ell \end{pmatrix},$$

and if we appropriately rearrange all the coefficients of the linear equality constraints in (4.1) and the objective function $\sum_{(i,j)\in F}[\boldsymbol{A}_0]_{ij}X_{ij}$ to reconstruct data matrices with the same block-diagonal structure as $\boldsymbol{X}'$, we obtain an standard equality form SDP.

There are two major advantages of this conversion. First, when the sizes of all positive semidefinite matrix variables in (4.1) are small, their Cholesky factorizations, computation of their minimum eigenvalues, and matrix multiplications require less CPU time than those of the original positive semidefinite matrix variable $\boldsymbol{X}$ in (1.1). Second, once we have converted the SDP (1.1) into the SDP with the block-diagonal positive semidefinite matrix variable $\boldsymbol{X}'$, we can apply effectively any interior-point method incorporating a block-diagonal matrix data structure ([4, 6, 28, etc.]) for SDPs.

We should note, however, that the conversion above from the SDP (1.1) to the SDP with the block-diagonal symmetric matrix variable $\boldsymbol{X}'$ increases the number of equality constraints from m to the number

$$m' = m + \sum_{r=2}^{\ell} \sharp\{(i,j) \in E_r : i \geq j\}.$$

When we apply interior-point methods to a standard form SDP having m equality constraints, we solve a system of linear equations with a fully dense $m \times m$ coefficient matrix $\boldsymbol{B}$ to generate a search direction at each iteration. This requires $\mathcal{O}(m^3)$ arithmetic operations. So the increase in the number of equality constraints in the converted problem may worsen the total computational efficiency. Therefore the reduction in the sizes of positive semidefinite matrix variables should be properly balanced with the increase in the number of equality constraints in (4.1) when we choose a chordal extension $G(V, F^\circ)$ of $G(V, E^\circ)$. In section 7, we will show by simple numerical examples how this balance is crucial.

5. Primal-dual interior-point method based on positive definite matrix completion. One disadvantage of the conversion of the SDP (1.1) to the SDP with multiple but smaller size positive semidefinite matrix variables (4.1) is an increase in the number of equality constraints. In this section, we propose a primal-dual interior-point method based on positive semidefinite matrix completion which exploits the mechanism of positive definite completion to compute the search directions and step lengths, and does not add any equality constraints in the original SDP formulation. Various search directions [1, 13, 15, 16, 20, 21, 24, 27] have been proposed so far for primal-dual interior-point methods. Among others, we restrict ourselves to the HRVW/KSH/M search direction [13, 16, 20], although we can adapt some of the discussions below to some other search directions.

There are two places below where we effectively utilize the equivalence between the constraint on the symmetric matrix variable $\boldsymbol{X} \in \mathcal{S}^n$ of the original problem (1.1) and the constraint (3.1) on the partial symmetric matrix $\boldsymbol{X} \in \mathcal{S}^n(F, ?)$ with entries specified in F. One is the computation of a search direction, and the other is the computation of a step length. Recall that E denotes the aggregate sparsity pattern of the data matrices $\boldsymbol{A}_0, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_m$, and that $G(V, F^\circ)$ denotes a chordal extension of $G(V, E^\circ)$.

Let $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}, \bar{\boldsymbol{z}})$ be a point obtained at the kth iteration of a primal-dual interior-point method using the HRVW/KSH/M search direction ($k \geq 1$) or given initially ($k = 0$). We assume that $\bar{\boldsymbol{X}} \in \mathcal{S}^n_{++}(F, ?)$ and $\bar{\boldsymbol{Y}} \in \mathcal{S}^n_{++}(E, 0)$. Here the feasibility of the point $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}, \bar{\boldsymbol{z}})$ is not assumed; $\bar{\boldsymbol{X}}$ and $(\bar{\boldsymbol{Y}}, \bar{\boldsymbol{z}})$ need not satisfy the equality constraints of the SDPs (1.1) and (1.2), respectively.

In order to compute the HRVW/KSH/M search direction, we use the whole matrix values for both $\bar{\boldsymbol{X}} \in \mathcal{S}^n_{++}(F, ?)$ and $\bar{\boldsymbol{Y}} \in \mathcal{S}^n_{++}(E, 0)$, so that we need to make a positive definite matrix completion of $\bar{\boldsymbol{X}} \in \mathcal{S}^n_{++}(F, ?)$. Let $\hat{\boldsymbol{X}} \in \mathcal{S}^n_{++}$ be the maximum-determinant positive definite matrix completion of $\bar{\boldsymbol{X}} \in \mathcal{S}^n_{++}(F, ?)$. See section 2.2. Then we compute the HRVW/KSH/M search direction $(d\boldsymbol{X}, d\boldsymbol{Y}, d\boldsymbol{z})$ by solving the system of linear equations

$$
\left.\begin{array}{l}
\boldsymbol{A}_p \bullet d\boldsymbol{X} = g_p \ (p = 1, 2, \ldots, m), \ d\boldsymbol{X} \in \mathcal{S}^n \\
\displaystyle\sum_{p=1}^{m} \boldsymbol{A}_p dz_p + d\boldsymbol{Y} = \boldsymbol{H}, \ d\boldsymbol{Y} \in \mathcal{S}^n(E, 0), \ d\boldsymbol{z} \in \boldsymbol{R}^m, \\
\widetilde{d\boldsymbol{X}} \bar{\boldsymbol{Y}} + \hat{\boldsymbol{X}} d\boldsymbol{Y} = \boldsymbol{K}, \ d\boldsymbol{X} = (\widetilde{d\boldsymbol{X}} + \widetilde{d\boldsymbol{X}}^T)/2,
\end{array}\right\} \tag{5.1}
$$

where $g_p = b_p - \boldsymbol{A}_p \bullet \hat{\boldsymbol{X}} \in \boldsymbol{R}$ $(p = 1, 2, \ldots, m)$ (the primal residual), $\boldsymbol{H} = \boldsymbol{A}_0 - \sum_{p=1}^{m} \boldsymbol{A}_p \bar{z}_p - \bar{\boldsymbol{Y}} \in \mathcal{S}^n(E, 0)$ (the dual residual), $\boldsymbol{K} = \mu \boldsymbol{I} - \hat{\boldsymbol{X}} \bar{\boldsymbol{Y}}$ (an $n \times n$ constant matrix), and $\widetilde{d\boldsymbol{X}}$ denotes an $n \times n$ auxiliary matrix variable. The search direction parameter μ is usually chosen to be $\beta \hat{\boldsymbol{X}} \bullet \bar{\boldsymbol{Y}}/n$ for some $\beta \in [0, 1]$. We can reduce the system of linear equations (5.1) to

$$
\left.\begin{array}{l}
\boldsymbol{B} d\boldsymbol{z} = \boldsymbol{s}, \ d\boldsymbol{Y} = \boldsymbol{H} - \displaystyle\sum_{p=1}^{m} \boldsymbol{A}_p dz_p, \\
\widetilde{d\boldsymbol{X}} = (\boldsymbol{K} - \hat{\boldsymbol{X}} d\boldsymbol{Y}) \bar{\boldsymbol{Y}}^{-1}, \ d\boldsymbol{X} = (\widetilde{d\boldsymbol{X}} + \widetilde{d\boldsymbol{X}}^T)/2,
\end{array}\right\} \tag{5.2}
$$

where

$$
\left.\begin{array}{l}
B_{pq} = \text{Trace } \boldsymbol{A}_p \hat{\boldsymbol{X}} \boldsymbol{A}_q \bar{\boldsymbol{Y}}^{-1} \ (p = 1, 2, \ldots, m, \ q = 1, 2, \ldots, m), \\
s_p = g_p - \text{Trace } \boldsymbol{A}_p (\boldsymbol{K} - \hat{\boldsymbol{X}} \boldsymbol{H}) \bar{\boldsymbol{Y}}^{-1} \ (p = 1, 2, \ldots, m).
\end{array}\right\}
$$

Note that $\boldsymbol{B}$ is a positive definite symmetric matrix.

Now recall that the maximum-determinant positive definite matrix completion $\hat{\boldsymbol{X}}$ of $\bar{\boldsymbol{X}} \in \mathcal{S}^n_{++}(F, ?)$ is expressed in terms of the sparse clique-factorization formula (2.16). Since we have assumed that $(1, 2, \ldots, n)$ is a perfect elimination ordering of the chordal graph $G(V, F^\circ)$ as in section 2.1, we can take the identity $\boldsymbol{I}$ for the permutation matrix $\boldsymbol{P}$ in (2.16). Hence the sparse clique-factorization formula (2.16) turns out to be

$$
\hat{\boldsymbol{X}} = \boldsymbol{L}_1^T \boldsymbol{L}_2^T \ldots \boldsymbol{L}_{\ell-1}^T \boldsymbol{D} \boldsymbol{L}_{\ell-1} \ldots \boldsymbol{L}_2 \boldsymbol{L}_1, \tag{5.3}
$$

where $\boldsymbol{L}_r$ $(r = 1, 2, \ldots, \ell - 1)$ and $\boldsymbol{D}$ are given by (2.17), (2.18) and (2.19). Also $\bar{\boldsymbol{Y}} \in \mathcal{S}^n_{++}(E, 0)$ is factorized as $\bar{\boldsymbol{Y}} = \boldsymbol{N}\boldsymbol{N}^T$ without any fill-in except for entries in $F \backslash E$, where $\boldsymbol{N}$ is a lower triangular matrix. We can effectively utilize these factorizations of $\hat{\boldsymbol{X}}$ and $\bar{\boldsymbol{Y}}$ for the computation of the search direction $(d\boldsymbol{X}, d\boldsymbol{Y}, d\boldsymbol{z})$. In particular, the coefficients B_{pq} $(p = 1, 2, \ldots, m, \ q = 1, 2, \ldots, m)$ in the system (5.2) of linear equations are computed by

$$
\begin{array}{l}
B_{pq} = \text{Trace } \boldsymbol{A}_p (\boldsymbol{L}_1^T \boldsymbol{L}_2^T \ldots \boldsymbol{L}_{\ell-1}^T \boldsymbol{D} \boldsymbol{L}_{\ell-1} \ldots \boldsymbol{L}_2 \boldsymbol{L}_1) \boldsymbol{A}_q (\boldsymbol{N}^{-T} \boldsymbol{N}^{-1}) \\
\qquad (p = 1, 2, \ldots, m, \ q = 1, 2, \ldots, m).
\end{array}
$$

If we utilize those factorizations also for the computation of s_p $(p = 1, 2, \ldots, m)$ and $\widetilde{d\boldsymbol{X}}$, we do not need even to store the whole dense matrix $\hat{\boldsymbol{X}}$ in the memory but only its sparse clique-factorizations in terms of $\boldsymbol{L}_1, \ \boldsymbol{L}_2, \ \ldots, \ \boldsymbol{L}_{\ell-1}$ and $\boldsymbol{D}$. As we will see below in the computation of a step length and a next iterate, we need the partial symmetric matrix with entries $[d\boldsymbol{X}]_{ij}$ specified in F, but not the whole search direction matrix $d\boldsymbol{X} \in \mathcal{S}^n$ in the primal space (hence the partial symmetric matrix with entries $[\widetilde{d\boldsymbol{X}}]_{ij}$ specified in F but not the whole matrix $\widetilde{d\boldsymbol{X}}$). Hence it is possible to carry out all the matrix computation above using only partial matrices with entries specified in F. Therefore we can expect to save both CPU time and memory in our computation of the search direction. To clarify the distinction between the whole primal search direction matrix $d\boldsymbol{X} \in \mathcal{S}^n$ and the corresponding partial symmetric matrix with entries specified in F in the discussions below, we use the notation $d\hat{\boldsymbol{X}}$ for the former whole matrix in $\mathcal{S}^n$ and $d\bar{\boldsymbol{X}}$ for the latter partial symmetric matrix in $\mathcal{S}^n(F;?)$. Now, suppose that we have computed the HRVW/KSH/M search direction $(d\bar{\boldsymbol{X}}, d\boldsymbol{Y}, d\boldsymbol{z}) \in \mathcal{S}^n \times \mathcal{S}^n(E,0) \times \boldsymbol{R}^m$. We describe how to compute a step length $\alpha > 0$ and the next iterate $(\boldsymbol{X}', \boldsymbol{Y}', \boldsymbol{z}') \in \mathcal{S}^n \times \mathcal{S}^n(E,0) \times \boldsymbol{R}^m$. Usually we compute the maximum $\hat{\alpha}$ of α's satisfying

$$\hat{\boldsymbol{X}} + \alpha d\hat{\boldsymbol{X}} \in \mathcal{S}^n_+ \ \text{ and } \bar{\boldsymbol{Y}} + \alpha d\boldsymbol{Y} \in \mathcal{S}^n_+, \tag{5.4}$$

and let $(\boldsymbol{X}', \boldsymbol{Y}', \boldsymbol{z}') = (\hat{\boldsymbol{X}}, \bar{\boldsymbol{Y}}, \bar{\boldsymbol{z}}) + \gamma\hat{\alpha}(d\hat{\boldsymbol{X}}, d\boldsymbol{Y}, d\boldsymbol{z})$ for some $\gamma \in (0,1)$. Then $\boldsymbol{X}' \in \mathcal{S}^n_{++}$ and $\boldsymbol{Y}' \in \mathcal{S}^n_{++}(E,0)$. The computation of $\hat{\alpha}$ is necessary to know how long we can take the step length along the search direction $(d\hat{\boldsymbol{X}}, d\boldsymbol{Y}, d\boldsymbol{z})$. The computation of $\hat{\alpha}$ is usually carried out by calculating the minimum eigenvalues of the matrices

$$\hat{\boldsymbol{M}}^{-1} d\hat{\boldsymbol{X}} \hat{\boldsymbol{M}}^{-T} \ \text{ and } \boldsymbol{N}^{-1} d\boldsymbol{Y} \boldsymbol{N}^{-T},$$

where $\hat{\boldsymbol{X}} = \hat{\boldsymbol{M}}\hat{\boldsymbol{M}}^T$ and $\bar{\boldsymbol{Y}} = \boldsymbol{N}\boldsymbol{N}^T$ denote the factorizations of $\hat{\boldsymbol{X}}$ and $\bar{\boldsymbol{Y}}$, respectively.

Instead of (5.4), we propose to employ

$$\bar{\boldsymbol{X}}_{C_rC_r} + \alpha d\bar{\boldsymbol{X}}_{C_rC_r} \in \mathcal{S}^{C_r}_+ \ (r = 1,2,\dots,\ell) \ \text{ and } \bar{\boldsymbol{Y}} + \alpha d\boldsymbol{Y} \in \mathcal{S}^n_+(E,0). \tag{5.5}$$

Recall that $\{C_r \subseteq V : r = 1,2,\dots,\ell\}$ denotes the family of maximal cliques of $G(V,F^\circ)$ and $\ell \le n$. Let $\bar{\alpha}$ be the maximum of α's satisfying (5.5), and let

$$(\boldsymbol{X}', \boldsymbol{Y}', \boldsymbol{z}') = (\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}, \bar{\boldsymbol{z}}) + \gamma\bar{\alpha}(d\bar{\boldsymbol{X}}, d\boldsymbol{Y}, d\boldsymbol{z}) \in \mathcal{S}^n(F,?) \times \mathcal{S}^n_{++}(E,0) \times \boldsymbol{R}^m$$

for some $\gamma \in (0,1)$. By Theorem 2.3, $\boldsymbol{X}' \in \mathcal{S}^n(F,?)$ has a positive definite matrix completion, so that the point $(\boldsymbol{X}', \boldsymbol{Y}', \boldsymbol{z}') \in \mathcal{S}^n_{++}(F,?) \times \mathcal{S}^n_{++}(E,0) \times \boldsymbol{R}^m$ can be the next iterate. In this case, the computation of $\bar{\alpha}$ is reduced to the computation of the minimum eigenvalues of the matrices

$$\bar{\boldsymbol{M}}_r^{-1} d\bar{\boldsymbol{X}}_{C_rC_r} \bar{\boldsymbol{M}}_r^{-T} \ (r = 1,2,\dots,\ell) \ \text{ and } \boldsymbol{N}^{-1} d\boldsymbol{Y} \boldsymbol{N}^{-T},$$

where $\bar{\boldsymbol{X}}_{C_rC_r} = \bar{\boldsymbol{M}}_r \bar{\boldsymbol{M}}_r^T$ denotes a factorization of $\bar{\boldsymbol{X}}_{C_rC_r}$ $(r = 1,2,\dots,\ell)$. Thus the computation of the minimum eigenvalue of $\hat{\boldsymbol{M}}^{-1}\hat{\boldsymbol{X}}\hat{\boldsymbol{M}}^{-T}$ has been replaced by the computation of the minimum eigenvalues of ℓ smaller submatrices $\bar{\boldsymbol{M}}_r^{-1} d\bar{\boldsymbol{X}}_{C_rC_r} \bar{\boldsymbol{M}}_r^{-T}$ $(r = 1,2,\dots,\ell)$.

We mention some important effects of the maximum-determinant positive definite matrix completion $\hat{\boldsymbol{X}} \in \mathcal{S}^n_{++}$ of $\bar{\boldsymbol{X}} \in \mathcal{S}^n_{++}(F,?)$ on the theoretical and practical convergence of the primal-dual interior-point method with the modification above. We first observe that

$$\boldsymbol{X} \bullet \bar{\boldsymbol{Y}} = \hat{\boldsymbol{X}} \bullet \bar{\boldsymbol{Y}} \ \text{ and } \det \boldsymbol{X} \le \det \hat{\boldsymbol{X}}$$

for any positive definite matrix completion $\boldsymbol{X} \in \mathcal{S}^n_{++}$ of $\bar{\boldsymbol{X}} \in \mathcal{S}^n_{++}(F,?)$. This implies that $\hat{\boldsymbol{X}} \in \mathcal{S}^n_{++}$ minimizes the value of the primal-dual potential function

$$\rho \log \boldsymbol{X} \bullet \bar{\boldsymbol{Y}} - \log\det(\boldsymbol{X}\bar{\boldsymbol{Y}})$$

over all positive definite matrix completions $\boldsymbol{X} \in \mathcal{S}^n_{++}$ of $\bar{\boldsymbol{X}} \in \mathcal{S}^n_{++}(F, ?)$, where ρ is a positive number. If we combine this fact with the primal-dual interior-point potential reduction method given in the paper [16] for SDPs, it is easy to design a polynomial-time primal-dual interior-point potential reduction method based on positive definite matrix completion for SDPs.

We also see that $\hat{\boldsymbol{X}}$ optimizes (maximizes) a centrality measure $\frac{\left(\det(\boldsymbol{X}\bar{\boldsymbol{Y}})\right)^{1/n}}{(\boldsymbol{X}\bullet\bar{\boldsymbol{Y}})/n}$ over all positive definite matrix completions $\boldsymbol{X} \in \mathcal{S}^n_{++}$ of $\bar{\boldsymbol{X}} \in \mathcal{S}_{++}(F, ?)$. Thus the maximum-determinant positive definite matrix completion is expected to work positively in both theoretical and practical convergence. It is not necessarily true, however, that $\hat{\boldsymbol{X}} \in \mathcal{S}^n_{++}$ optimizes (minimizes) the standard centrality measure $\left\|\boldsymbol{X}^{1/2}\bar{\boldsymbol{Y}}\boldsymbol{X}^{1/2} - \boldsymbol{X}\bullet\bar{\boldsymbol{Y}}/n\right\|$ over all positive definite matrix completions $\boldsymbol{X} \in \mathcal{S}^n_{++}$ of $\bar{\boldsymbol{X}} \in \mathcal{S}_{++}(F, ?)$. Here $\|\cdot\|$ denotes the Frobenius norm of a matrix.

Another positive effect of our modification is that the maximum $\bar{\alpha}$ of α's satisfying (5.5) is larger than or equal to the maximum $\hat{\alpha}$ of α's satisfying (5.4). So we are able to choose a larger step length if we use (5.5) instead of (5.4).

6. Linear transformation in the primal and dual spaces. When we are given an SDP to be solved, we may be able to transform it to a sparser SDP to which we apply more effectively the conversion method in section 4 and/or the primal-dual interior-point method based on positive definite matrix completion in section 5. As we will see in the later part of this section, certain semidefinite programming relaxations of some combinatorial optimization problems including the graph equipartition problem and the maximum clique problem are such cases.

We introduce a general framework for transformation of a given SDP which induces an equivalence class of SDPs. For every $\boldsymbol{\mathcal{A}} = (\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_m) \in \prod_{p=0}^m \mathcal{S}^n$ and $\boldsymbol{b} \in \boldsymbol{R}^m$, we use the notation $\mathrm{P}(\boldsymbol{\mathcal{A}}, \boldsymbol{b})$ for the standard equality form SDP (1.1) and the notation $\mathrm{D}(\boldsymbol{\mathcal{A}}, \boldsymbol{b})$ for its dual (1.2).

Let $\boldsymbol{P}$ be an arbitrary $n \times n$ nonsingular matrix. Performing the congruence transformation $\boldsymbol{X} = \boldsymbol{P}\boldsymbol{X}'\boldsymbol{P}^T$ from $\boldsymbol{X}$ to $\boldsymbol{X}'$ in the primal space, we obtain an SDP $\mathrm{P}(\boldsymbol{\mathcal{A}}^{\mathrm{p}}, \boldsymbol{b})$ and its dual $\mathrm{D}(\boldsymbol{\mathcal{A}}^{\mathrm{p}}, \boldsymbol{b})$, where

$$\boldsymbol{\mathcal{A}}^{\mathrm{p}} = (\boldsymbol{P}^T\boldsymbol{A}_0\boldsymbol{P}, \boldsymbol{P}^T\boldsymbol{A}_1\boldsymbol{P}, \boldsymbol{P}^T\boldsymbol{A}_2\boldsymbol{P}, \ldots, \boldsymbol{P}^T\boldsymbol{A}_m\boldsymbol{P}) \in \prod_{k=0}^m \mathcal{S}^n.$$

Let $\boldsymbol{D}$ be an $m \times m$ arbitrary nonsingular matrix, and $\boldsymbol{\zeta}$ an arbitrary vector in $\boldsymbol{R}^m$. Performing the affine transformation

$$\boldsymbol{z} = \boldsymbol{D}\boldsymbol{z}' - \boldsymbol{\zeta}$$

from $\boldsymbol{z}$ to $\boldsymbol{z}'$ in the dual space, we obtain an SDP $\mathrm{D}(\boldsymbol{\mathcal{A}}^{\mathrm{d}}, \boldsymbol{b}^{\mathrm{d}})$ and the corresponding primal SDP $\mathrm{P}(\boldsymbol{\mathcal{A}}^{\mathrm{d}}, \boldsymbol{b}^{\mathrm{d}})$, where

$$\begin{aligned}
&\boldsymbol{b}^{\mathrm{d}} = \boldsymbol{D}^T\boldsymbol{b} \in \boldsymbol{R}^m, \\
&\boldsymbol{A}_0^{\mathrm{d}} = \boldsymbol{A}_0 + \sum_{p=1}^m \boldsymbol{A}_p\zeta_p \in \mathcal{S}^n, \ \boldsymbol{A}_k^{\mathrm{d}} = \sum_{p=1}^m \boldsymbol{A}_p D_{pk} \in \mathcal{S}^n \ (k = 1, 2, \ldots, m), \\
&\boldsymbol{\mathcal{A}}^{\mathrm{d}} = (\boldsymbol{A}_0^{\mathrm{d}}, \boldsymbol{A}_1^{\mathrm{d}}, \boldsymbol{A}_2^{\mathrm{d}}, \ldots, \boldsymbol{A}_m^{\mathrm{d}}) \in \prod_{k=0}^m \mathcal{S}^n.
\end{aligned}$$

If we perform the primal transformation and the dual transformation simultaneously, we obtain another primal-dual pair of SDPs $\mathrm{P}(\boldsymbol{\mathcal{A}}^{\mathrm{pd}}, \boldsymbol{b}^{\mathrm{pd}})$ and $\mathrm{D}(\boldsymbol{\mathcal{A}}^{\mathrm{pd}}, \boldsymbol{b}^{\mathrm{pd}})$, where

$$
\begin{aligned}
\boldsymbol{b}^{\mathrm{pd}} &= \boldsymbol{D}^T \boldsymbol{b} \in \boldsymbol{R}^m, \ \boldsymbol{A}_0^{\mathrm{pd}} = \boldsymbol{P}^T \boldsymbol{A}_0 \boldsymbol{P} + \sum_{p=1}^{m} \boldsymbol{P}^T \boldsymbol{A}_p \boldsymbol{P} \zeta_p \in \mathcal{S}^n, \\
\boldsymbol{A}_k^{\mathrm{pd}} &= \sum_{p=1}^{m} \boldsymbol{P}^T \boldsymbol{A}_p \boldsymbol{P} D_{pk} \in \mathcal{S}^n \ (k = 1, 2, \dots, m), \\
\boldsymbol{\mathcal{A}}^{\mathrm{pd}} &= (\boldsymbol{A}_0^{\mathrm{pd}}, \boldsymbol{A}_1^{\mathrm{pd}}, \boldsymbol{A}_2^{\mathrm{pd}}, \dots, \boldsymbol{A}_m^{\mathrm{pd}}) \in \prod_{k=0}^{m} \mathcal{S}^n.
\end{aligned}
$$

By construction, all the primal-dual pairs, $\mathrm{P}(\boldsymbol{\mathcal{A}}, \boldsymbol{b})$ and $\mathrm{D}(\boldsymbol{\mathcal{A}}, \boldsymbol{b})$, $\mathrm{P}(\boldsymbol{\mathcal{A}}^{\mathrm{p}}, \boldsymbol{b}^{\mathrm{p}})$ and $\mathrm{D}(\boldsymbol{\mathcal{A}}^{\mathrm{p}}, \boldsymbol{b}^{\mathrm{p}})$, $\mathrm{P}(\boldsymbol{\mathcal{A}}^{\mathrm{d}}, \boldsymbol{b}^{\mathrm{d}})$ and $\mathrm{D}(\boldsymbol{\mathcal{A}}^{\mathrm{d}}, \boldsymbol{b}^{\mathrm{d}})$, $\mathrm{P}(\boldsymbol{\mathcal{A}}^{\mathrm{pd}}, \boldsymbol{b}^{\mathrm{pd}})$ and $\mathrm{D}(\boldsymbol{\mathcal{A}}^{\mathrm{pd}}, \boldsymbol{b}^{\mathrm{pd}})$ are equivalent to each other. The important issue here is how we choose $\boldsymbol{P}$, $\boldsymbol{D}$ and $\boldsymbol{\zeta}$

- to improve the aggregate sparsity pattern of the data matrices, and also,
- to reduce the total number of nonzeros in the data matrices, which affects the computation of the coefficient matrix $\boldsymbol{B}$ of the linear equation in (5.2) to determine a search direction. See also [7].

It should be noted that any transformation using an $m \times m$ nonsingular matrix $\boldsymbol{D}$ and an m-dimensional vector $\boldsymbol{\zeta}$ in the dual space never changes the aggregate sparsity pattern of the data matrices, but it may be useful to decrease the total number of nonzeros in the data matrices specially when some data matrices are 0-1 or integral (see also (D) of section 8 for further discussion on this transformation). We will show below two cases in which an appropriate congruence transformation $\boldsymbol{P}$ in the primal space improves the aggregate sparsity pattern of data matrices.

First consider a structured SDP with data matrices having the following sparsity pattern

$$
\boldsymbol{A}_0 = \begin{pmatrix} * & \boldsymbol{O} & \boldsymbol{O} & * \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & * \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & * \\ * & * & * & * \end{pmatrix}, \ \boldsymbol{A}_1 = \begin{pmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & * \\ \boldsymbol{O} & * & \boldsymbol{O} & * \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & * \\ * & * & * & * \end{pmatrix},
$$
$$
\boldsymbol{A}_2 = \begin{pmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & * \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & * \\ \boldsymbol{O} & \boldsymbol{O} & * & * \\ * & * & * & * \end{pmatrix}, \ \boldsymbol{A}_p = \begin{pmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & * \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & * \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & * \\ * & * & * & * \end{pmatrix} \ (p = 3, 4, \dots, m).
$$

Here $*$ denotes a (possibly) nonzero matrix. In this case, the aggregate sparsity pattern matrix turns out to be a bordered block-diagonal matrix

$$
\begin{pmatrix} * & \boldsymbol{O} & \boldsymbol{O} & * \\ \boldsymbol{O} & * & \boldsymbol{O} & * \\ \boldsymbol{O} & \boldsymbol{O} & * & * \\ * & * & * & * \end{pmatrix}.
$$

Since each of the first three nonzero blocks in the diagonal is due to $\boldsymbol{A}_0$, $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, respectively, and no other data matrices $\boldsymbol{A}_p$ $(p = 3, 4, \dots, m)$ contain any nonzeros in those diagonal-blocks, we can choose a nonsingular matrix $\boldsymbol{P}$ of the form

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{P}_{11} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{P}_{22} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{P}_{33} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{P}_{44} \end{pmatrix}$$

such that the transformed data matrices $\boldsymbol{P}^T\boldsymbol{A}_p\boldsymbol{P}$ $(p = 0, 1, \ldots, m)$ get the aggregate sparsity pattern

$$\begin{pmatrix} \diamond & \boldsymbol{O} & \boldsymbol{O} & * \\ \boldsymbol{O} & \diamond & \boldsymbol{O} & * \\ \boldsymbol{O} & \boldsymbol{O} & \diamond & * \\ * & * & * & * \end{pmatrix}.$$

Here each $\diamond$ denotes a diagonal matrix. Thus the aggregate sparsity pattern has been improved along the diagonal.

Now we consider the SDP relaxation of the graph equi-partition problem, which is formulated as

$$\left.\begin{array}{ll} \text{minimize} & \boldsymbol{A}_0 \bullet \boldsymbol{X} \\ \text{subject to} & \boldsymbol{E}_p \bullet \boldsymbol{X} = \dfrac{1}{4} \ (p = 1, 2, \ldots, n), \\ & \boldsymbol{E} \bullet \boldsymbol{X} = 0, \ \boldsymbol{X} \in \mathcal{S}^n_+. \end{array}\right\}$$

Here $\boldsymbol{A}_0 = \operatorname{diag}(\boldsymbol{Ce}) - \boldsymbol{C}$, $\boldsymbol{C}$ denotes an $n \times n$ symmetric cost matrix, $\operatorname{diag}(\boldsymbol{Ce})$ denotes the diagonal matrix whose entries are $\boldsymbol{Ce}$, $\boldsymbol{E}_p$ denotes the $n \times n$ matrix with all entries 0 except $[\boldsymbol{E}_p]_{pp} = 1$, and $\boldsymbol{E}$ denotes the $n \times n$ matrix with all entries 1. When the graph under consideration is sparse, the matrix $\boldsymbol{C}$ (hence the matrix $\boldsymbol{A}_0$) is sparse. But the aggregate sparsity pattern matrix is fully dense due to the only one fully dense matrix $\boldsymbol{E}$. To improve the sparsity pattern, we perform the congruence transformation using

$$\boldsymbol{P} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$$

to the data matrices $\boldsymbol{A}_0$, $\boldsymbol{E}_p$ $(p = 1, 2, \ldots, n)$ and $\boldsymbol{E}$ in the primal space to obtain

$$\boldsymbol{P}^T\boldsymbol{A}_0\boldsymbol{P}, \boldsymbol{P}^T\boldsymbol{E}_p\boldsymbol{P} \ (p = 1, 2, \ldots, n) \ \text{ and } \boldsymbol{P}^T\boldsymbol{E}\boldsymbol{P}.$$

Then all entries of $\boldsymbol{P}^T\boldsymbol{E}\boldsymbol{P}$ vanish except $[\boldsymbol{P}^T\boldsymbol{E}\boldsymbol{P}]_{11} = 1$. We can also verify that

$$\begin{array}{l} (\text{the total number of nonzeros of the matrices } \boldsymbol{P}^T\boldsymbol{A}_0\boldsymbol{P},\ \boldsymbol{P}^T\boldsymbol{E}_p\boldsymbol{P}\ (p = 1, 2, \ldots, n)) \\ \quad \leq 4 \times (\text{the total number of nonzeros of the matrices } \boldsymbol{A}_0,\ \boldsymbol{E}_p\ (p = 1, 2, \ldots, n)). \end{array}$$

Therefore if $\boldsymbol{A}_0$ is sparse this transformation reduces the total number of nonzeros in data matrices, and improves the aggregate sparsity pattern.

We can apply the same congruence transformation above to the SDP relaxation of the maximum clique problem.

7. Numerical examples. In this section, we give three numerical examples which show the effectiveness, advantages and disadvantages of the conversion method described in section 4. This conversion can be interpreted as a preprocessing scheme to the existing software [4, 6, 28] which can handle standard equality form SDPs (1.1) and (1.2) with block-diagonal data matrices. In particular, we use SDPA 5.0 [6] to solve the SDPs of this section on a DEC Alpha machine (300MHz with 256MB of memory). The first two examples illustrate remarkable effectiveness of the conversion method, and also the importance of determining an "optimal" chordal extension of the aggregate sparsity pattern for a given SDP. The third example exhibits a crucial disadvantage of employing this conversion compared with the primal-dual interior-point method based on positive definite matrix completion proposed in section 5.

We start by describing the first two examples which are randomly generated SDPs with high sparsity and special structures. The first problem is the example given in section 1. Let V denote the set $\{1, 2, \ldots, n\}$ of row/column indices of the data matrices $\boldsymbol{A}_p$ $(p = 0, 1, \ldots, m)$, and $E_{\mathrm{b}} = \{(i, n), (n, i), (i, i) : i \in V\}$ be the aggregate sparsity pattern of the data matrices. We call this example the *1-bordered diagonal SDP*. In the second example, the aggregate sparsity pattern is replaced by $E_{\mathrm{t}} = \{(i, j) \in V \times V : |i - j| \leq 1\}$ instead. We call this example the *tridiagonal SDP*. Notice that the graphs associated with the aggregate sparsity patterns, $G(V, E_{\mathrm{b}}^{\circ})$ and $G(V, E_{\mathrm{t}}^{\circ})$, are already chordal. Nevertheless, we can consider other chordal extensions which include them, namely, the graphs corresponding to the bordered block-diagonal matrix (3.2), and the graphs corresponding to (3.3), respectively. For both examples, we fixed the dimensions of the symmetric matrices $\boldsymbol{A}_p \in \mathcal{S}^n$ $(p = 0, 1, \ldots, m)$ to be equal to $n = 2^9 + 1 = 513$, and the numbers of equality constraints in the primal SDP formulation to be equal to $m = 79$. For each of the matrices $\boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_m$ of the 1-bordered diagonal SDP (tridiagonal SDP, respectively), we randomly generated three nonzero entries at some $(i, j) \in E_{\mathrm{b}}$ ($(i, j) \in E_{\mathrm{t}}$, respectively), and, for $\boldsymbol{A}_0$, we generated nonzero elements for all (i, j)th entries in E_{b} (E_{t}, respectively).

Since a similar discussion for the 1-bordered diagonal SDP will be also valid for the tridiagonal SDP, we focus on the former example for the moment. According to the notation in sections 1, 3, and 4, for each $k \in \{0, 1, \ldots, 9\}$, let us define $S_0 = \{n\}$, and $S_r = \{1 + (r-1)2^{9-k}, 2 + (r-1)2^{9-k}, \ldots, r2^{9-k}\}$ $\quad (r = 1, 2, \ldots, 2^k)$. Defining now $C_r = S_0 \cup S_r$ $\quad (r = 1, 2, \ldots, 2^k)$, and $F_{\mathrm{b}} = \bigcup_{r=1}^{2^k} C_r \times C_r$, $G(V, F_{\mathrm{b}}^{\circ})$ will be a chordal extension of $G(V, E_{\mathrm{b}}^{\circ})$. Using the formula (4.1), we can convert the 1-bordered diagonal SDP to equivalent SDPs whose sizes are specified in Table 7.1. Observe that $k = 0$ gives the original SDP.

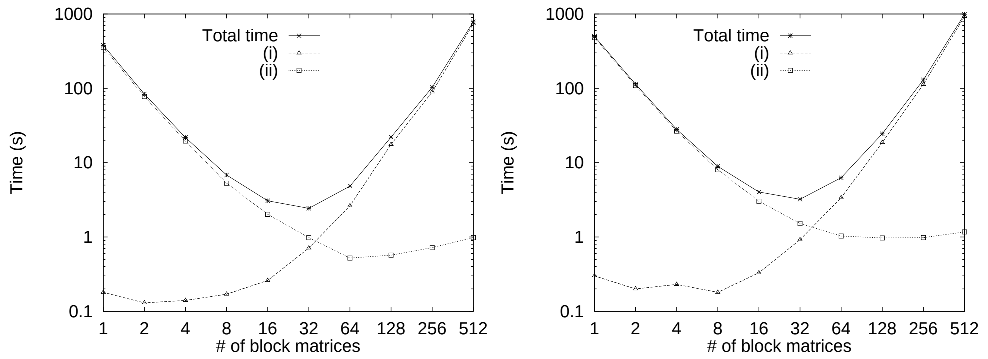
TABLE 7.1
Sizes of the equivalent SDPs to the 1-bordered diagonal SDP and the tridiagonal SDP.

k	# block matrices (2^k)	dimension of each block	# constraints m'
0	1	513×513	79 + 0
1	2	257×257	79 + 1
2	4	129×129	79 + 3
3	8	65×65	79 + 7
4	16	33×33	79 + 15
5	32	17×17	79 + 31
6	64	9×9	79 + 63
7	128	5×5	79 + 127
8	256	3×3	79 + 255
9	512	2×2	79 + 511

Figure 7.1(a) shows in log scale the total time (solid line) to solve the equivalent SDPs listed in Table 7.1 using SDPA. Most of the total time is spent in two major subroutines in SDPA:

(i) Time to compute the search direction $(d\boldsymbol{X}, d\boldsymbol{Y}, d\boldsymbol{z})$ — by calculating the coefficient matrix $\boldsymbol{B} \in \mathcal{S}^{m'}_{++}$, and solving the Schur complement equation $\boldsymbol{B}d\boldsymbol{z} = \boldsymbol{s}$ — dashed line in Figure 7.1;

(ii) Time to compute the step length for the search direction — by computing the Cholesky factorization and the eigenvalues, and performing matrix operations for each small block matrix $(\mathcal{O}((\#C_r)^3))$— dotted line in Figure 7.1.



(a) 1-bordered diagonal SDP *(b) tridiagonal SDP*

FIG. 7.1. *Computational time for the 1-bordered diagonal SDP and the tridiagonal SDP (Total time; (i) Time to compute search directions; (ii) Time to compute step lengths).*

Figure 7.1(a) shows that we have to select a "good" chordal extension $G(V, F^{\circ}_{\mathrm{b}})$ in order to balance the time spent in (i), which mainly depends on the number of equality constraints m', and the time spent in (ii), which mainly depends on the dimensions of the small block matrices. This balance is crucial to reduce the total computational time to solve the SDP. For the 1-bordered diagonal SDP, a partition of the original problem into 32 small block matrices of 17×17 dimension each ($k = 5$) gives the "optimal" conversion, and it reduces the total computational time by a factor of approximately 150.

A similar discussion can be made for the tridiagonal SDP. In this case, given $k \in \{0, 1, \ldots, 9\}$, a chordal extension of the graph associated with the aggregate sparsity pattern $G(V, E^{\circ}_{\mathrm{t}})$ is chosen such that the maximal cliques for it are given by (3.3) with $\beta = 1$ and $\kappa = 2^{9-k}$. The sizes of each equivalent SDP to the tridiagonal SDP are given in Table 7.1. The computational time is shown in Figure 7.1(b). Notice the similarity between the computational time for these two examples with extremely sparse data matrices $\boldsymbol{A}_0, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_m$.

We observe two points from these numerical examples:

(a) the problem of detecting an "optimal" chordal extension of the aggregate sparsity pattern for an SDP is extremely important in order to balance the time spent in (i) and (ii), and therefore reduce the total computational time;

(b) the conversion to a multiple block matrices of smaller size (section 4) is extremely efficient when very sparse data matrices $\boldsymbol{A}_0, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_m$ have special sparsity patterns, and the number of added constraints $(m' - m)$ in the equivalent SDP is relatively small.

The last example comes from the topology optimization problem of truss structures [25], and we call it the *topology optimization SDP* here. The aggregate sparsity pattern for the data matrices $\boldsymbol{A}_p$ $(p = 0, 1, \ldots, 392)$ after diminishing the bandwidth by the reverse Cuthill-McKee ordering is shown in Figure 7.2. This matrix consists of two diagonal-blocks: a 327×327 block matrix with a small bandwidth, and a 392×392 diagonal matrix. Since SDPA can handle the latter diagonal matrix quite efficiently, we will only consider the block matrix with the small bandwidth. We define the chordal extension of the graph associated with the sparsity pattern of this block matrix as $G(V, F^{\circ}_{\text{top}})$, where $F_{\text{top}} = \{(i, j) \in V \times V : |i - j| \leq 45\}$ and $V = \{1, 2, \ldots, 327\}$. The maximal cliques corresponding to this chordal extension are given in (3.3).

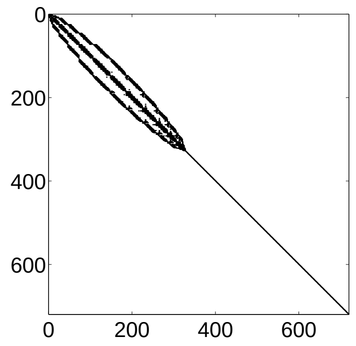


FIG. 7.2. *Aggregate sparsity pattern for the data matrices of the topology optimization SDP.*

The sizes of the SDPs resulting from the topology optimization SDP by the conversion method, and the computational time to solve them are shown in Table 7.2. The time to compute the search directions (i) grows drastically compared to the decrease in the time to compute the step lengths (ii) in this case, because we have to add $45 \cdot (45+1)/2$ new variables and equality constraints if we increase the number of block matrices by one. See (4.1). In this case, it is much better to solve the original SDP instead of converting it.

TABLE 7.2
Sizes of the equivalent SDPs to the topology optimization SDP and the computational time to solve them.

# block matrices	dimension of each block	# constraints m'	Total time	(i)	(ii)
1(+1)	327×327 $(+392 \times 392)$	392	237 s	69 s	164 s
2(+1)	186×186 $(+392 \times 392)$	$392 + 1035$	715 s	657 s	56 s
3(+1)	137×137 $(+392 \times 392)$	$392 + 2070$	3190 s	2550 s	36 s
4(+1)	115×115 $(+392 \times 392)$	$392 + 3105$	9032 s	8995 s	49 s

( ) indicates the block corresponding to the diagonal matrix

The last example shows a fundamental drawback of the conversion method:

(c) a large number of additional equality constraints are often required in the converted SDP.

Although we might be able to utilize more sophisticated ordering such as the nested dissection ordering to decrease the number of additional equality constraints, this drawback exhibits a certain limitation of the conversion method for practical use. In section 5, we have proposed a method to compute the search directions and the step length in the primal-dual interior-point method based on positive definite matrix completion which do not need the conversion method of section 4, and therefore avoids the above drawback (c). In Part II [22] of this article, we will continue discussing the technical details and implementation of this method, and we present its numerical results applied to larger classes of SDPs.

8. Concluding discussion. We have proposed two kinds of methods for a large scale sparse SDP exploiting the aggregate sparsity pattern E over its data matrices. One is a conversion of such an SDP into an SDP having multiple but smaller size positive semidefinite matrix variables. The other is a primal-dual interior-point method based on maximum-determinant positive definite matrix completion. Concerning practical implementation of these two methods, however, there remain lots of significant and interesting issues which we need to investigate further. Among others, we mention:

(A) How do we find an effective chordal extension $G(V, F^\circ)$ of $G(V, E^\circ)$? This issue is common to both of the methods. In Part II [22] of this article, we will study more extensively how we can utilize some of the existing ordering methods, such as the minimum degree ordering, the (nested) dissection ordering and the reverse Cuthill-McKee ordering, developed for the Cholesky factorization.

(B) The computation of the search direction, which we have discussed in section 5, for the latter method is also a very important issue. In Part II [22], we will explore more details on (i) how we efficiently construct the product form representation (5.3) of the maximum-determinant positive definite matrix completion $\hat{\boldsymbol{X}}$ of a partial symmetric matrix $\bar{\boldsymbol{X}} \in \mathcal{S}^n_{++}(F, ?)$ and (ii) how we compute the coefficients B_{pq} $(p = 1, 2, \dots, m,\ q = 1, 2, \dots, m)$ of the key linear equation $\boldsymbol{B}d\boldsymbol{z} = \boldsymbol{s}$ in (5.2) by utilizing the representation (5.3) effectively.

(C) Our methods still need to solve the Schur complement equation $\boldsymbol{B}d\boldsymbol{z} = \boldsymbol{s}$. As we have mentioned in the Introduction, the coefficient matrix $\boldsymbol{B}$ is fully dense in general, so that it becomes more difficult to apply direct methods to the equation as its size (= the number of equality constraints in the primal SDP (1.1)) becomes larger. To solve a large scale SDP having not only a large size matrix variable but also a large number of equality constraints, we can incorporate iterative methods [19, 23] to solve the Schur complement equation into our methods.

(D) The linear transformation in the primal and the dual spaces described in section 6 may be regarded as a preprocessing or preconditioning technique for SDPs. Since the transformation in the dual space does not affect the aggregate sparsity pattern of data matrices of a given SDP to be solved, without damaging the computational efficiency much, we may be able to use the transformation for numerical stability, which is another major purpose of preprocessing besides computational efficiency. In particular, if we apply the dual transformation using an $m \times m$ nonsingular matrix $\boldsymbol{D}$ and a $\boldsymbol{\zeta} \in \boldsymbol{R}^m$ to an SDP with data matrices $\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2, \dots, \boldsymbol{A}_m$, the coefficients B_{pq} $(p = 1, 2, \dots, m,\ q = 1, 2, \dots, m)$ of the key linear equation $\boldsymbol{B}d\boldsymbol{z} = \boldsymbol{s}$ in (5.2) turn out to be

$$B_{pq} = \text{Trace}\ \left(\sum_{k=1}^{m} \boldsymbol{A}_k D_{kp}\right) \hat{\boldsymbol{X}} \left(\sum_{k=1}^{m} \boldsymbol{A}_k D_{kq}\right) \bar{\boldsymbol{Y}}^{-1}\ (p = 1, 2, \dots, m,\ q = 1, 2, \dots, m).$$

Thus the transformation may work as a preconditioning for iterative methods such as the conjugate gradient method and the conjugate residual method (see

[19, 23]). It should be noted that the matrix $\boldsymbol{D}$ has enough parameters to control the eigenvalues of the matrix $\boldsymbol{B}$, although we need to investigate an effective choice of the matrix $\boldsymbol{D}$.

Theoretically it is an interesting issue to see whether we can design a polynomial-time and/or locally superlinearly convergent primal-dual path-following interior-point method based the maximum-determinant positive definite matrix completion.

Acknowledgment. The authors thank Dr. Akihisa Tamura of Kyoto University for helpful comments.

REFERENCES

[1] F. Alizadeh, J.-P. A. Haeberly and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results*, SIAM J. Optim., 8 (1998), pp. 746–768.

[2] S. J. Benson, Y. Ye and X. Zhang, *Solving large-scale sparse semidefinite programs for combinatorial optimization*, SIAM J. Optim., 10 (2000), pp. 443–461.

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

[4] B. Borchers, CSDP *2.3 user's guide*, Optim. Methods Softw., 11–12 (1999), pp. 597–611. Available at http://www.nmt.edu/~borchers/csdp.html.

[5] K. Fujisawa, M. Fukuda, M. Kojima and K. Nakata, *Numerical evaluation of* SDPA *(SemiDefinite Programming Algorithm)*, in High Performance Optimization, H. Frenk, K. Roos, T. Terlaky and S. Zhang, eds., Kluwer Academic Publishers, Dordrecht, 1999, pp. 267–301.

[6] K. Fujisawa, M. Kojima and K. Nakata, SDPA *(Semidefinite Programming Algorithm) — User's Manual —*, Technical Report B-308, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Japan, December 1995 (revised August 1996). Available at ftp://ftp.is.titech.ac.jp/pub/OpRes/software/SDPA.

[7] K. Fujisawa, M. Kojima and K. Nakata, *Exploiting sparsity in primal-dual interior-point methods for semidefinite programming*, Math. Program., 79 (1997), pp. 235–253.

[8] D. R. Fulkerson and O. A. Gross, *Incidence matrices and interval graphs*, Pacific J. Math., 15 (1965), pp. 835–855.

[9] A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, 1981.

[10] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[11] R. Gron, C. R. Johnson, E. M. Sá and H. Wolkowicz, *Positive definite completions of partial hermitian matrices*, Linear Algebra Appl., 58 (1984), pp. 109–124.

[12] C. Helmberg and F. Rendl, *Solving quadratic* $(0,1)$*-problems by semidefinite programming and cutting planes*, Math. Program., 82 (1998), pp. 291–315.

[13] C. Helmberg, F. Rendl, R. J. Vanderbei and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. Optim., 6 (1996), pp. 342–361.

[14] C. R. Johnson, *Matrix completion problems: a survey*, Proc. Sympos. Appl. Math., 40 (1990), pp. 171–198.

[15] M. Kojima, M. Shida and S. Shindoh, *Search directions in the* SDP *and the monotone* SDLCP*: generalization and inexact computation*, Math. Program., 85 (1999), pp. 51–80.

[16] M. Kojima, S. Shindoh and S. Hara, *Interior-point methods for the monotone semidefinite linear complementarity problems*, SIAM J. Optim., 7 (1997), pp. 86–125.

[17] M. Laurent, *Cuts, matrix completions and graph rigidity*, Math. Program., 79 (1997), pp. 255–283.

[18] J. G. Lewis, B. W. Peyton and A. Pothen, *A fast algorithm for reordering sparse matrices for parallel factorization*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 1146–1173.

[19] C.-J. Lin and R. Saigal, *Semidefinite programming and the quadratic assignment problem*, 16th International Symposium on Mathematical Programming, Lausanne, Switzerland, August 1997.

[20] R. D. C. Monteiro, *Primal-dual path-following algorithms for semidefinite programming*, SIAM J. Optim., 7 (1997), pp. 663–678.

[21] R. D. C. Monteiro and Y. Zhang, *A unified analysis for a class of long-step primal-dual path-following interior-point algorithms for semidefinite programming*, Math. Program., 81 (1998), pp. 281–299.

[22] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima and K. Murota, *Exploiting sparsity in semidefinite programming via matrix completion II: implementation and numerical results*, in preparation.

[23] K. Nakata, K. Fujisawa and M. Kojima, *Using the conjugate gradient method in interior-point methods for semidefinite programs* (in Japanese), Proc. Inst. Statist. Math., 46 (1998), pp. 297–316.

[24] Yu. E. Nesterov and M. J. Todd, *Primal-dual interior-point methods for self-scaled cones*, SIAM J. Optim., 8 (1998), pp. 324–364.

[25] M. Ohsaki, K. Fujisawa, N. Katoh and Y. Kanno, *Semi-definite programming for topology optimization of trusses under multiple eigenvalue constraints*, Comput. Methods Appl. Mech. Engrg., 180 (1999), pp. 203–217.

[26] D. J. Rose, R. E. Tarjan and G. S. Lueker, *Algorithmic aspects of vertex elmination on graphs*, SIAM J. Comput., 5 (1976), pp. 266-283.

[27] M. J. Todd, K. C. Toh and R. H. Tütüncü, *On the Nesterov-Todd direction in semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 769–796.

[28] K.C. Toh, M. J. Todd and R. H. Tütüncü, SDPT3 — *a* MATLAB *software package for semidefinite programming, Version 1.3*, Optim. Methods Softw., 11—12 (1999), pp. 545–581. Available at http://www.math.nus.edu.sg/~mattohkc/index.html.