Decomposition methods for sparse matrix nearness problems

Yifan Sun*  Lieven Vandenberghe*

Abstract

We discuss three types of sparse matrix nearness problems: given a sparse symmetric matrix, find the matrix with the same sparsity pattern that is closest to it in Frobenius norm and (1) is positive semidefinite, (2) has a positive semidefinite completion, or (3) has a Euclidean distance matrix completion. Several proximal splitting and decomposition algorithms for these problems are presented and their performance is compared on a set of test problems. A key feature of the methods is that they involve a series of projections on small dense positive semidefinite or Euclidean distance matrix cones, corresponding to the cliques in a triangulation of the sparsity graph. The methods discussed include the dual block coordinate ascent algorithm (or Dykstra’s method), the dual projected gradient and accelerated projected gradient algorithms, and a primal and a dual application of the Douglas-Rachford splitting algorithm.

1 Introduction

We discuss matrix optimization problems of the form

\[
\begin{align*}
\text{minimize} & \quad \|X - C\|_F^2, \\
\text{subject to} & \quad X \in S,
\end{align*}
\]

where the variable $X$ is a symmetric $p \times p$ matrix with a given sparsity pattern $E$, and $C$ is a given symmetric matrix with the same sparsity pattern as $X$. We focus on the following three choices for the constraint set $S$.

I. $S$ is the set of sparse positive semidefinite (PSD) matrices with sparsity pattern $E$:

\[ S = S^+_E \cap S^+_p. \]

Here $S^+_E$ denotes the set of sparse matrices of order $p$ with sparsity pattern $E$ and $S^+_p = \{ X \in S^p \mid X \succeq 0 \}$ is the set of positive semidefinite matrices of order $p$. In this problem we approximate a matrix $C$ by the nearest positive semidefinite matrix with sparsity pattern $E$.

II. $S$ is the set of sparse matrices with sparsity pattern $E$ that have a positive semidefinite completion:

\[ S = \Pi_E(S^+_p) = \{ \Pi_E(S) \mid S \in S^+_p \}. \]

Here, $\Pi_E(S)$ denotes Euclidean projection on $S^+_E$. In applications of this problem, the nonzeros in $C$ are observations of a subset of the entries of a positive semidefinite matrix. The

*Electrical Engineering Department, University of California, Los Angeles. Email: ysun01@ucla.edu, vandenbe@ucla.edu.
positive semidefinite matrices $S$ that match the values of $C$ in a least squares sense are the positive semidefinite completions of the solution $X$ of the problem (1). These completions $S$ are dense matrices in general and the sparse matrix $X$ is useful as a compact representation from which a full completion can be computed if needed.

III. $S$ is the set of sparse matrices with sparsity pattern $E$ that have a Euclidean distance matrix (EDM) completion:

$$S = \Pi_E(D^p).$$

Here, $D^p$ is the set of Euclidean distance matrices of order $p$, i.e., matrices with entries that can be expressed as squared pairwise Euclidean distances between pairs of points. The interpretation of this problem is similar to problem II, but with the Euclidean distance matrix property replacing positive semidefiniteness.

Note that the assumption that $C$ is sparse can be made without loss of generality. If $C$ is dense or not symmetric, it can be replaced by $(1/2)\Pi_E(C + C^T)$ in the objective function of (1).

We also discuss the following generalization of problems II and III:

$$\begin{align*}
\text{minimize} & \quad \|\Pi_E(X) - C\|_F^2 \\
\text{subject to} & \quad X \in S,
\end{align*}$$

with variable $X \in S^p_{E'}$, where $E'$ is an extension of $E$ (i.e., the nonzero positions in $E$ are a subset of the nonzeros in $E'$) and $S = \Pi_{E'}(S^p_{E'})$ or $S = \Pi_{E'}(D^p)$. In the extended problem (2) we approximate $C$ by a sparse matrix $\Pi_E(X)$ that has a positive semidefinite or EDM completion, and we compute some of the entries (indexed by $E' \setminus E$) of a completion of $\Pi_E(X)$.

The techniques presented in the paper are easily extended to similar problems with simple additional constraints on $X$ (for example, problems I and II with a constraint $\text{diag}(X) = 1$).

1.1 Matrix nearness problems

Problem (1) belongs to the general class of matrix nearness problems [36]. In a matrix nearness problem one projects a given matrix on the set of matrices that satisfy a certain property. Problems of this type arise in a wide range of applications, including statistics, machine learning, finance, signal processing, control, and computational biology, and can take a variety of different forms. The matrix $C$ may be symmetric or nonsymmetric, square or rectangular. The constraint set may be convex or not, and the objective function may be a Frobenius norm or another norm, or a nonmetric distance function. In this paper, we restrict our attention to problems involving symmetric matrix variables and convex constraints.

Matrix nearness problems with positive semidefinite and Euclidean distance matrix constraints are among the most widely studied types of matrix nearness problems. The nearest positive semidefinite matrix and nearest correlation matrix problems have applications in statistics, finance, and biology, and are discussed in [50, 37, 45, 44, 40, 2]. The nearest EDM problem is studied in multidimensional scaling techniques (see [64] for an overview), and is used in chemistry and molecular biology to compute molecular structure [62, 27, 57, 6], and for node localization in sensor networks [38, 3].

These applications all involve fairly simple convex sets and, in principle, they can be solved by general-purpose convex optimization algorithms. However, due to the large number of variables (order $p^2$ for dense matrix variables) the complexity of general-purpose methods grows rapidly.
with the matrix dimension $p$. Research on matrix nearness problems has therefore concentrated on
specialized first-order algorithms, or, when higher accuracy is required, quasi-Newton algorithms.
Examples of such first-order methods are alternating projection methods [26, 37, 27, 2], dual first-
order methods [16, 34], augmented Lagrangian methods [46], and alternating direction methods [15].
Algorithms based on quasi-Newton methods in combination with dual reformulations are described in [44, 40, 48].

In comparison, algorithms tailored to sparse matrix nearness problems, i.e., problems with an
additional sparsity constraint, have received less attention in the literature. Sparsity constraints
arise naturally in large matrix nearness problems and can have a different meaning depending on the
application and the interpretation of the data matrix $C$. In a first type of problem, we are interested
in recovering a sparse matrix with the properties represented as a set $S$, from a noisy measurement
or estimate of its nonzero elements. Most applications of problem I are of this type. A typical
example is the estimation of a sparse covariance or inverse covariance matrix from estimates of its
nonzero entries. In other situations, the sparse matrix $C$ represents a noisy estimate of a subset
of the entries of a dense matrix, which is known to have the structural properties represented by a
set $\hat{S}$. In this case the matrices in $\hat{S}$ that best fit the measurements are dense. When the matrix
dimensions are large it is of interest to avoid working with a dense matrix variable and represent
it implicitly as a sparse matrix, with the added constraint that the matrix has a completion with
the desired properties. The sparse partial matrix variable is therefore constrained to lie in the set
$S = \Pi_E(\hat{S})$. Problems II and III are motivated by applications of this kind, with $\hat{S}$ the set of
positive semidefinite matrices or Euclidean distance matrices, respectively. In some applications,
partial knowledge of the solution as a sparse completable matrix is sufficient. If the full dense
matrix is desired, it can be constructed from the partial solution by matrix completion algorithms
[29, 7].

Most algorithms for sparse matrix nearness problems in the literature are extensions of algo-
rithms for dense matrix nearness problems. In problem I one can use a dense matrix variable
and impose the sparsity constraint by adding linear equality constraints. Similarly, problems II
and III can be solved by using a dense matrix variable and masking irrelevant entries in the objective
[37, 46, 5, 47, 4]. Extensions of this type are still computationally expensive. In particular, they
involve eigenvalue decompositions of order $p$, used for projecting on the set of positive semidefinite
matrices or the set of Euclidean distance matrices. In contrast, the approach taken in this paper
is to avoid eigenvalue decompositions of order $p$ and only use eigenvalue decompositions of smaller
dense matrices. By applying decomposition techniques for cones of sparse matrices with chordal
structure, we write the three types of sparse matrix cones in terms of small dense positive semidef-
inite or Euclidean distance matrix cones [29, 28, 1, 7]. In combination with first-order methods,
these chordal decomposition techniques allow us to solve the matrix nearness problems without
using eigenvalue decompositions of order $p$.

1.2 Outline and notation

The rest of the paper is organized as follows. In section 2, we discuss some fundamental geometrical
properties of cones with partially separable structure. In section 3 we present decomposition results
for the three sparse matrix nearness problems. We formulate the problems as Euclidean projections
on convex cones with partially separable structure. Section 4 presents two types of dual algorithms
that apply when $E$ is a chordal sparsity pattern. In section 5 we discuss extensions that apply
when the pattern $E$ is not necessarily chordal. These extensions also handle the more general
problem (2). Section 6 gives numerical results with test problems that range from order \( p \sim 1000 \) to 100,000.

**Notation**  If \( \beta \) is an index set (ordered subset) of \( \{1, \ldots, p\} \), we define \( P_\beta \) as the \( |\beta| \times p \)-matrix

\[
(P_\beta)_{ij} = \begin{cases} 1 & j = \beta(i) \\ 0 & \text{otherwise} \end{cases}
\]

Multiplying a vector with \( P_\beta \) selects the subvector indexed by \( \beta \):

\[
P_\beta x = (x_{\beta(1)}, x_{\beta(2)}, \ldots, x_{\beta(r)})
\]

if \( \beta \) has \( r \) elements, denoted \( \beta(1), \beta(2), \ldots, \beta(r) \). Similarly, the matrix \( P_\beta \) can be used to select a principal submatrix of a \( p \times p \) matrix:

\[
(P_\beta X P_\beta^T)_{ij} = X_{\beta(i)\beta(j)}.
\]

The multiplication \( x = P_\beta^T y \) of a \( |\beta| \)-vector \( y \) with the transpose of \( P_\beta \) gives a \( p \)-vector \( x \) with \( P_\beta x = y \) and \( x_j = 0 \) for \( j \notin \beta \). The operation \( X = P_\beta^T Y P_\beta \) creates a \( p \times p \) matrix \( X \) from an \( r \times r \) matrix, with \( P_\beta X P_\beta^T = Y \) and \( X_{ij} = 0 \) for \( i \notin \beta \) or \( j \notin \beta \). For example, if \( p = 5 \) and \( \beta = \{1, 3, 4\} \) then

\[
P_\beta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad P_\beta^T y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 0 \end{bmatrix}, \quad P_\beta^T Y P_\beta = \begin{bmatrix} Y_{11} & 0 & Y_{12} & Y_{13} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ Y_{21} & 0 & Y_{22} & Y_{23} & 0 \\ Y_{31} & 0 & Y_{32} & Y_{33} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.
\]

We use the notation \( \delta_S(x) \) for the indicator function of a set \( S \):

\[
\delta_S(x) = \begin{cases} 0 & x \in S \\ +\infty & \text{otherwise} \end{cases}
\]

The notation \( \Pi_S \) is used for the Euclidean projection operator on a closed set \( S \).

### 2 Partially separable convex cones

In this section we discuss the standard convex optimization form to which the matrix nearness problems will be reduced and work out some basic results from convex duality.

#### 2.1 Projection on partially separable cone

The general problem we consider is

\[
\begin{align*}
\text{minimize} & \quad \|P_\gamma x - a\|^2 \\
\text{subject to} & \quad P_{\gamma_k} x \in C_k, \quad k = 1, \ldots, l
\end{align*}
\]
where $\| \cdot \|$ denotes the Euclidean norm. The optimization variable is an $n$-vector $x$. The sets $\gamma_1, \ldots, \gamma_l$ are index sets (ordered subsets of $\{1, 2, \ldots, n\}$) and $C_k$, $k = 1, \ldots, l$, are closed convex cones. We will use the notation

$$K = \{ x \in \mathbb{R}^n \mid P_{\gamma_k} x \in C_k, \ k = 1, \ldots, l \}$$

for the feasible set in (3). This is a closed convex cone and equal to the intersection $K = \bigcap_{k=1}^l K_k$ of the closed convex cones

$$K_k = \{ x \in \mathbb{R}^n \mid P_{\gamma_k} x \in C_k \}, \ \ k = 1, \ldots, l.$$  

The objective in (3) is the squared distance between the subvector $P_{\eta} x = x_\eta$ and a given $|\eta|$-vector $a$:

$$\| P_{\eta} x - a \|^2 = \sum_{k=1}^{l} (x_{\eta(k)} - a_k)^2.$$  

An important special case will be $\eta = \{1, 2, \ldots, n\}$. Then $P_\eta = I$ and the problem has a unique solution $x^*$, the projection of $a$ on the closed convex cone $K$.

When $\eta \neq \{1, 2, \ldots, n\}$ the problem is to project $a$ on $P_\eta K$, the projection of $K$ on the subspace of vectors with support $\eta$. In order to ensure that $P_\eta K$ is a closed set, we assume that $P_\eta x = 0$, $x \in K$ only hold for $x = 0$. This assumption is satisfied in the applications discussed in this paper (section 3). If the assumption holds, closedness of $P_\eta K$ follows from Theorem 8.1 in [51], which states that if $K$ is a closed convex cone and has a nontrivial intersection with the nullspace of $A$, then $AK$ is a closed convex cone. Since $P_\eta K$ is closed, problem (3) has a unique optimal solution for the subvector $P_\eta x^*$. However, the components of $x^*$ outside $\eta$ are not necessarily unique.

In the problems of interest the cones $C_k$ will be subspaces or closed convex cones with nonempty interior. We will assume that the problem (3) is strictly feasible, i.e., there exists an $\bar{x}$ with $P_{\gamma_k} \bar{x} \in \text{ri} C_k$ for $k = 1, \ldots, l$, where $\text{ri}$ stands for relative interior. This assumption is satisfied in the applications we discuss, and will be needed in section 2.2.

A key property of problem (3) is the partial separability of the cone $K$, i.e., the indicator function $\sum_{k=1}^l \delta_{C_k}(P_{\gamma_k} x)$ of $K$ is a partially separable function [28]. Partial separability with small index sets ($|\gamma_k| \ll n$) is a very useful type of structure for decomposition algorithms in conic optimization [56]. It is crucial to the efficiency of the algorithms presented in this paper and allows us to solve problems of the form (3) via a sequence of projections on the lower-dimensional cones $C_k$.

### 2.2 Conic duality

The Lagrange dual of problem (3) is

$$\begin{align*}
\text{maximize} \quad & -\| s + a \|^2 + \| a \|^2 \\
\text{subject to} \quad & P^T \eta s = \sum_{k=1}^l P^T_{\gamma_k} z_k \\
& z_k \in C_k^*, \quad k = 1, \ldots, l.
\end{align*}$$

(4)

The variables are $s \in \mathbb{R}^{|\eta|}$ and multipliers $z_k \in \mathbb{R}^{|\gamma_k|}$ for the $l$ constraints in (3). The sets $C_k^*$ are the dual cones $C_k^* = \{ z_k \mid s_k^T z_k \geq 0 \ \forall s_k \in C_k \}$ of $C_k$. If $C_k$ is a subspace, then $C_k^*$ is the
orthogonal complement of the subspace. Projections on $C^*_k$ can be computed via projections on $C_k$ and vice-versa, via the formulas

$$y = \Pi_{C_k}(y) + \Pi_{-C^*_k}(y) = \Pi_{C_k}(y) - \Pi_{C^*_k}(-y).$$

The variable $s$ in (4) can be eliminated by noting that $(P^T_\eta s)_i = 0$ for $i \notin \eta$, and therefore $\|s + a\| = \|P^T_\eta (s + a)\|$. This gives an equivalent form of the dual problem

$$\begin{align*}
\text{maximize} & \quad -\|\sum_{k=1}^l P^T_\gamma z_k + P^T_\eta a\|^2 + \|a\|^2 \\
\text{subject to} & \quad (\sum_{k=1}^l P^T_\gamma z_k)_i = 0, \quad i \notin \eta \\
& \quad z_k \in C^*_k, \quad k = 1, \ldots, l.
\end{align*}$$

(5)

The dual problem has a simple geometric interpretation. First assume that $\eta = \{1, 2, \ldots, n\}$ and $P_\eta = I$. Recall our assumption that there exists a point $\bar{x}$ in the intersection of the sets $r_iK_k$, $k = 1, \ldots, l$. The assumption implies that the dual cone of $K$ is $K^* = K^*_1 \pm \cdots \pm K^*_l$ [51, corollary 16.4.2]. Here $K^*_k = \{P^T_\gamma z_k \mid z_k \in C^*_k\}$, so

$$K^* = \left\{ \sum_{k=1}^l P^T_\gamma z_k \mid z_k \in C^*_k, \quad k = 1, \ldots, l \right\}.$$

The solution $s$ of problem (4) is the projection of $-a$ on $K^*$. (Equivalently, $-s$ is the projection of $a$ to the polar cone $-K^*$.) The solutions $x^*$ and $s^*$ of the two projection problems

$$\begin{align*}
\text{minimize} & \quad \|x - a\|^2 \\
\text{subject to} & \quad x \in K
\end{align*}$$

$$\begin{align*}
\text{maximize} & \quad -\|s + a\|^2 + \|a\|^2 \\
\text{subject to} & \quad s \in K^*
\end{align*}$$

are unique and related by the optimality conditions

$$a = x^* - s^*, \quad x^* \in K, \quad s^* \in K^*, \quad s^T x^* = 0.$$

This shows one can solve either the primal or the dual problem and compute the other solution from the relation $a = x^* - s^*$. The geometry is illustrated in figure 1. Note that although $x^*$ and $s^*$ are unique, the dual problem (4) is not strictly convex in the variables $z_1, \ldots, z_l$, so those optimal values are not necessarily unique.

When $\eta \neq \{1, 2, \ldots, n\}$, we can interpret the problem (4) in a similar way. Its solution $s^*$ is the projection of $-a$ on the dual cone of $P_\eta K$, which is given by $(P_\eta K)^* = \{s \mid P^T_\eta s \in K^*\}$. The optimality conditions that relate the two projections are

$$a = P_\eta x^* - s^*, \quad x^* \in K, \quad P^T_\eta s^* \in K^*, \quad s^T P_\eta x^* = 0.$$

Note that here the primal and dual approaches are not completely equivalent. From the solution $x^*$ of the primal problem (13) one obtains the dual solution $s^* = P_\eta x^* - a$. However from the dual solution $s^*$ one only finds a partial primal solution $P_\eta x^*$ and not the values $x^*_i$ for $i \notin \eta$.
3 Decomposition of sparse matrix nearness problems

In sections 3.2–3.4 we review three fundamental decomposition results for classes of matrices with chordal sparsity patterns and apply them to reformulate the matrix nearness problems listed in the introduction in the standard forms (3) and (4). We first give some background on chordal sparsity.

3.1 Chordal sparsity patterns

A symmetric sparsity pattern $E$ of order $p$ is defined as a set of index pairs $E \subseteq \{(i, j) \mid i, j \in \{1, \ldots, p\}, i \neq j\}$.

A symmetric $p \times p$ matrix $X$ is said to have sparsity pattern $E$ if $X_{ij} = 0$ for all $(i, j) \not\in E$ and $i \neq j$. The diagonal entries $X_{ii}$ and the off-diagonal entries $X_{ij}$ for $(i, j) \in E$ may or may not be zero. The set of symmetric matrices of order $p$ with sparsity pattern $E$ is denoted $S_E^p$.

With every sparsity pattern $E$ we can associate an undirected graph $G_E$ with vertex set $\{1, \ldots, p\}$ and edge set $E$. A clique (also called maximal clique) of a graph $G_E$ is a maximal subset of pairwise adjacent vertices. A clique $\beta$ in $G_E$ corresponds to a dense principal submatrix $X_{\beta\beta}$ of a matrix $X \in S_E^p$. (Here, ‘dense’ means that all entries may be nonzero.)

The sparsity pattern $E$ is chordal if the graph $G_E$ is chordal, i.e., all cycles in $G_E$ of length greater than three have a chord (an edge between non-consecutive vertices in the cycle). Chordal graphs are important in linear algebra because every positive definite matrix $X$ with a chordal sparsity pattern has a zero-fill Cholesky factorization, i.e., a factorization $P_\sigma X P_\sigma^T = LL^T$, where $P_\sigma$ is a permutation matrix, $L$ is lower triangular, and $P_\sigma^T (L + L^T) P_\sigma \in S_E^p$ [52]. The reordering by the permutation matrix $P_\sigma$ is called a perfect elimination ordering for $E$. A chordal graph with $p$ vertices has at most $p$ cliques and they are easily identified from a perfect elimination ordering; see [14] for a survey on chordal graphs.

A sparsity pattern $E'$ with $E \subset E'$ is called an extension of the sparsity pattern $E$. The graph $G_{E'}$ is the graph $G_E$ with additional edges $E' \setminus E$. Clearly, $S_E^p \subset S_{E'}^p$. The extension is a chordal extension if $G_{E'}$ is a chordal graph. A chordal extension is also called a triangulation. The problem of computing the minimum chordal extension (i.e., with minimum $|E'| - |E|$) is NP-complete [63]. The problem of computing a minimal chordal extension, i.e., a chordal extension
with the property that there exists no chordal extension \( \hat{E} \subset E' \), is much more tractable and several algorithms for it have been developed; see [33] for a survey. Chordal extensions can also be computed from a fill-reducing ordering, such as the (approximate) minimum degree reordering [22]. Although such extensions are not guaranteed to be minimal, they are often smaller than extensions computed by minimal ordering algorithms.

It will be convenient to represent sparse or dense symmetric matrices as vectors. For a dense matrix \( A \in S^p \), we define \( \text{vec}(A) \) as the \( \frac{p(p+1)}{2} \)-vector containing the elements of the lower-triangular part of \( A \) and with off-diagonal entries scaled by \( \sqrt{2} \), so that \( \text{tr}(AB) = \text{vec}(A)^T \text{vec}(B) \) for all \( A, B \in S^p \). Similarly, for a sparse matrix \( A \in S^p_E \), we define \( \text{vec}_E(A) \) as an \( (|E|+p) \)-vector containing the diagonal elements and the lower-triangular elements \( A_{ij} \) for \( \{i, j\} \in E \), again with the off-diagonal elements scaled by \( \sqrt{2} \), so that \( \text{tr}(AB) = \text{vec}_E(A)^T \text{vec}_E(B) \) for all \( A, B \in S^p_E \).

### 3.2 Positive semidefinite matrices

We now consider the first problem from the introduction: the projection of a sparse matrix \( A \in S^p_E \) on the set

\[ S = S^p_E \cap S^p_+ \]

of positive semidefinite matrices in \( S^p_E \). This is a closed convex cone (for any \( E \)). The following theorem from [28, 1] provides a characterization when the sparsity pattern is chordal.

**Theorem 1** Let \( E' \) be a chordal sparsity pattern of order \( p \), with cliques \( \beta_k, k = 1, \ldots, m \). A matrix \( S \in S^p_{E'} \) is positive semidefinite if and only if it can be expressed as

\[ S = \sum_{k=1}^m P_{\beta_k}^T Z_k P_{\beta_k} \]

with \( Z_k \succeq 0 \) for \( k = 1, \ldots, m \).

When \( E' \) is not chordal, the condition in the theorem is sufficient for positive semidefiniteness of \( S \), but not necessary.

Using theorem 1, we can formulate the problem of projecting a matrix \( A \in S^p_E \) on \( S^p_E \cap S^p_+ \), for a general sparsity pattern \( E \), as

\[
\begin{aligned}
\text{minimize} & \quad \left\| \sum_{k=1}^m P_{\beta_k}^T Z_k P_{\beta_k} - A \right\|_F^2 \\
\text{subject to} & \quad \left( \sum_{k=1}^m P_{\beta_k}^T Z_k P_{\beta_k} \right)_{ij} = 0, \quad \{i, j\} \in E' \setminus E \\
& \quad Z_k \succeq 0, \quad k = 1, \ldots, m.
\end{aligned}
\]

(6)

Here \( E' \) is a chordal extension of \( E \) and \( \beta_1, \ldots, \beta_m \) are the cliques of \( G_{E'} \). The variables \( Z_k \in S^{\beta_k} \) are used to parameterize a general matrix \( S = \sum_k P_{\beta_k}^T Z_k P_{\beta_k} \) in \( S^p_{E'} \cap S^p_+ \), according to theorem 1. The equality constraints force the off-diagonal elements \( S_{ij} \) with \( \{i, j\} \in E' \setminus E \) to be zero. If \( E \) is already chordal, we take \( E' = E \) and omit the equality constraints. The size of the optimization problem (6) depends on the efficiency of the chordal extension: the sizes of the cliques of \( G_{E'} \) determine the dimensions of the variables \( Z_k \) and the amount of fill \( E' \setminus E \) determines the number of equality constraints. Note that the problem (6) is strictly feasible, since \( Z_k = I, k = 1, \ldots, m \), is strictly feasible.
Problem (6) can be written in vector notation in the form (5). We define the variables $z_k = \text{vec}(Z_k)$, $k = 1, \ldots, l$. We take $l = m$, $n = p + |E'|$, and $a = -\text{vec}_E(A)$. The index sets $\eta$ and $\gamma_k$ are defined by imposing that the identities

$$P_{\eta}\text{vec}_{E'}(X) = \text{vec}_E(X), \quad P_{\gamma_k}\text{vec}_{E'}(X) = \text{vec}(P_{\beta_k}XP_{\beta_k}^T), \quad k = 1, \ldots, l, \quad (7)$$

hold for all $X \in S_{E'}^p$. The cone $C_k^*$ is the cone of dense positive semidefinite matrices of order $|\beta_k|$ converted to vectors:

$$C_k^* = \{\text{vec}(U) \mid U \in S_+^{|\beta_k|}, \quad k = 1, \ldots, m\}.$$

For later purposes we note that the Euclidean projection on $C_k^*$ can be computed by an eigenvalue decomposition: if $U \in S^{[\beta_k]}$ has eigenvalue decomposition $U = \sum_i \lambda_i q_iq_i^T$ then $\Pi_{C_k^*}(\text{vec}(U)) = \sum_i \max\{\lambda_i, 0\}\text{vec}(q_iq_i^T)$.

### 3.3 Matrices with positive semidefinite completions

Next we consider the projection of a sparse matrix on the cone of sparse symmetric matrices that have a positive semidefinite completion. This set is the projection of the cone $S_{E'}^p$ on $S_{E'}^p$, 

$$\Pi_{E'}(S_{E'}^p) = \{\Pi_E(X) \mid X \in S_{E'}^p\}.$$ 

The projection $\Pi_{E'}(S_{E'}^p)$ is a closed convex cone, for the following reason. If $\Pi_E(X) = 0$ then the diagonal of $X$ is zero. Therefore $\Pi_E(X) = 0$ and $X \in S_{E'}^p$ imply $X = 0$. This is a sufficient condition for the projection $\Pi_{E'}(S_{E'}^p)$ to be closed [51, theorem 9.1].

The following theorem is the dual counterpart of theorem 1 and gives a characterization of $\Pi_{E'}(S_{E'}^p)$ when the sparsity pattern $E'$ is chordal [29].

**Theorem 2** Let $E'$ be a chordal sparsity pattern of order $p$, with cliques $\beta_k$, $k = 1, \ldots, m$. A matrix $X \in S_{E'}^p$ is in $\Pi_{E'}(S_{E'}^p)$ if and only if

$$P_{\beta_k}XP_{\beta_k}^T \succeq 0, \quad k = 1, \ldots, m.$$ 

If $E'$ is not chordal, the condition in the theorem is necessary for $X \in \Pi_{E'}(S_{E'}^p)$ but not sufficient.

Using theorem 2, the problem of projecting a matrix $B \in S_{E'}^p$ on $\Pi_{E'}(S_{E'}^p)$ can be written as

$$\begin{align*}
\text{minimize} & \quad \|\Pi_E(X) - B\|_F^2 \\
\text{subject to} & \quad P_{\beta_k}XP_{\beta_k}^T \succeq 0, \quad k = 1, \ldots, m. \quad (8)
\end{align*}$$

The variable is $X \in S_{E'}^p$, where $E'$ is a chordal extension of $E$, and $\beta_1, \ldots, \beta_m$ are the cliques of $E'$. When $E' = E$, the projection $\Pi_E$ is the identity operator and we replace $\Pi_E(X)$ by $X$ in the objective. We note that problem (8) is strictly feasible, since $X = I$ is strictly feasible.

Problem (8) can be put in the form (3) with variable $x = \text{vec}_{E'}(X)$. The problem dimensions $n = p + |E'|$ and $l = m$, the vector $a = \text{vec}_{E'}(B)$, and the index sets $\eta, \gamma_1, \ldots, \gamma_l$ are defined as in the previous section. Since the dense positive semidefinite cone is self-dual, the cones $C_k$ are equal to $C_k^*$.

If $B = -A$, problems (6) and (8) are duals, and special cases of the primal and dual pair (3) and (4). The optimal solutions $S^* = \sum_k P_{\beta_k}Z_kP_{\beta_k}$ of (6) and $X^*$ of (8) are related as

$$S^* = \Pi_E(X^*) + A.$$ 

This shows that if $E' = E$, the two problems are equivalent (with $B = -A$), since $S^* = X^* + A$. When $E \subset E'$, one can compute from the solution $X^*$ the solution $S^*$. However, from $S^*$, one only obtains a partial solution $\Pi_E(X^*)$ of (8).
3.4 Matrices with a Euclidean distance matrix completion

A matrix $X \in \mathbf{S}^p$ is a Euclidean distance matrix (EDM) if its elements can be expressed as $X_{ij} = \|u_i - u_j\|^2$ for some set of points $u_k$, $k = 1, \ldots, p$. Schoenberg’s condition [53] states that $X$ is an EDM if and only if

$$\text{diag}(X) = 0 \quad \text{and} \quad c^T X c \leq 0 \text{ for all } c \text{ with } 1^T c = 0,$$

(9)

where $1$ represents the $p$-vector with all elements equal to one. From this it is clear that the set of Euclidean distance matrices is the intersection of closed cones, and therefore a closed convex cone. The notation $\mathbf{D}^p$ will be used for the set of Euclidean distance matrices matrices of order $p$, and $\mathbf{D}_0^p$ for the matrices that satisfy only the second condition in (9):

$$\mathbf{D}_0^p = \{ X \in \mathbf{S}^p | c^T X c \leq 0 \text{ for all } c \text{ with } 1^T c = 0 \} = \{ X \in \mathbf{S}^p | V^T XV \preceq 0 \}$$

where $V$ is any $p \times (p-1)$ matrix whose columns span the orthogonal complement of the vector $1$ in $\mathbb{R}^p$. The Euclidean distance matrices are the matrices in $\mathbf{D}_0^p$ with zero diagonal. The cones $\mathbf{D}_0^p$ and $\mathbf{D}^p$ are closed, convex cones, with two noticeable differences. The cone $\mathbf{D}^p$ has empty interior in $\mathbf{S}^p$, while the interior of $\mathbf{D}_0^p$ is nonempty (it contains the matrix $X = -I$). Additionally, $\mathbf{D}^p$ is pointed (since it contains only nonnegative matrices), while $\mathbf{D}_0^p$ is not, as it contains the subspace \{ $a1^T + 1a^T | a \in \mathbb{R}^p$ \}.

A sparse matrix $X \in \mathbf{S}_E^p$ has a Euclidean distance matrix completion if it is the projection $X = \Pi_E(D)$ of some Euclidean distance matrix $D$. The cone

$$\Pi_E(\mathbf{D}^p) = \{ \Pi_E(X) | X \in \mathbf{D}^p \}$$

is a closed convex cone, as can be seen as follows. First suppose the sparsity graph $G_E$ is connected. If $\Pi_E(X) = 0$ for some $X \in \mathbf{D}^p$, with $X_{ij} = \|u_i - u_j\|^2$ for $i, j = 1, \ldots, p$, then $u_i = u_j$ for all $i, j \in E$. If the graph is connected, this implies that the position vectors $u_i$ are all equal, i.e., $X = 0$. Hence $\Pi_E(X) = 0$, $X \in \mathbf{D}^p$ only holds if $X = 0$. It then follows from [51, theorem 9.1] that $\Pi_E(\mathbf{D}^p)$ is closed. Next, assume $G_E$ has $d$ connected components, with vertex sets $\alpha_1, \ldots, \alpha_d \subseteq \{1, 2, \ldots, p\}$. For $k = 1, \ldots, d$, let $E_k = \{ \{i, j\} | \{ \alpha_k(i), \alpha_k(j) \} \in E \}$ be the edge sets of the connected components of $G_E$. Since for each $k$, the graph with vertex set $\{1, 2, \ldots, |\alpha_k|\}$ and edge set $E_k$ is connected, the sets $\Pi_{E_k}(\mathbf{D}^{|\alpha_k|})$ are all closed convex cones. Additionally, $X \in \Pi_E(\mathbf{D}^p)$ if and only if $X \in \mathbf{S}_E^p$ and $X_{\alpha_k \alpha_k} \in \Pi_{E_k}(\mathbf{D}^{|\alpha_k|})$ for $k = 1, \ldots, d$. Hence $\Pi_E(\mathbf{D}^p)$ is the intersection of closed convex sets.

Bakonyi and Johnson have formulated a clique decomposition theorem for Euclidean distance matrix completion analogous to theorem 2 for positive semidefinite completion [7].

**Theorem 3** Let $E'$ be a chordal sparsity pattern of order $p$, with cliques $\beta_k$, $k = 1, \ldots, m$. A matrix $X \in \mathbf{S}_{E'}^p$ is in $\Pi_{E'}(\mathbf{D}^p)$ if and only if

$$X_{\beta_k \beta_k} \in \mathbf{D}^{|\beta_k|}, \quad k = 1, \ldots, m.$$ 

If $E'$ is not chordal, the condition in the theorem is necessary for $X$ to be in $\Pi_{E'}(\mathbf{D}^p)$ but not sufficient.
In combination with Schoenberg’s condition, this theorem allows us to formulate the problem of projecting a matrix $B \in S^p_E$ on $\Pi_E(D^p)$:

$$\begin{align*}
\text{minimize} & \quad \| \Pi_E(X) - B \|_F^2 \\
\text{subject to} & \quad P_{\beta_k} X P_{\beta_k}^T \in D^{[\beta_k]}, \quad k = 1, \ldots, m
\end{align*}$$

with variable $X \in S^p_{E'}$, where $E'$ is a chordal extension of $E$, and $\beta_1, \ldots, \beta_m$ are the cliques of $E'$. Equivalently, we can write the problem as

$$\begin{align*}
\text{minimize} & \quad \| \Pi_E(X) - B \|_F^2 \\
\text{subject to} & \quad P_{\beta_k} X P_{\beta_k}^T \in D^{[\beta_k]}, \quad k = 1, \ldots, m \\
& \quad \text{diag}(X) = 0.
\end{align*} \tag{10}$$

Problem (10) is strictly feasible. (For example, the matrix $X = 11^T - I$ is strictly feasible.) We will use (10) because projections on $D^{[\beta_k]}$ are simpler than projections on $D^{[\beta_k]}$, as will be discussed shortly. Problem (10) can be written in vector form as (3), with variable $x = \text{vec}_{E'}(X)$. Here, $n = p + |E'|$, $a = \text{vec}_{E}(B)$, and $l = m + 1$. The index sets $\eta, \gamma_1, \ldots, \gamma_m$ are defined by the identity (7) for all $X \in S^p_{E'}$. The index set $\gamma_{m+1}$ is defined by

$$P_{\gamma_{m+1}} \text{vec}_{E'}(X) = \text{diag}(X) \quad \forall X \in S^p_{E'}.$$ 

The cones $C_k$ are defined as $C_k = \{ \text{vec}(X) \mid X \in D_0^{[\beta_k]} \}, \quad k = 1, \ldots, m$, and $C_{m+1} = \{ 0 \} \subset \mathbb{R}^p$.

Formulas for projecting on $D_0^{p}$ can be found in [32, 26, 27]. Let $V$ be a $p \times (p-1)$ matrix with orthonormal columns that span the orthogonal complement of $1$. An example is the matrix

$$V = \frac{1}{p + \sqrt{p}} \left[ (1 + \sqrt{p})1^T - (p + \sqrt{p})I \right].$$

Define $e = (1/\sqrt{p})1 \in \mathbb{R}^p$ and $Q = [ V \quad e ]$. In [32, 27] the projection of a matrix $D \in S^p$ on $D_0^{p}$ is computed directly, as the solution of

$$\begin{align*}
\text{minimize} & \quad \| X - D \|_F^2 \\
\text{subject to} & \quad V^T X V \preceq 0.
\end{align*} \tag{11}$$

Since $Q$ is orthogonal, the problem is equivalent to

$$\begin{align*}
\text{minimize} & \quad \| Q^T X Q - Q^T D Q \|_F^2 = \left\| \begin{bmatrix} V^T X V - V^T D V & V^T X e - V^T D e \\ e^T X V - e^T D V & e^T X e - e^T D e \end{bmatrix} \right\|_F^2 \\
\text{subject to} & \quad V^T X V \preceq 0.
\end{align*}$$

From this it is clear that at the optimum the $1,1$ block of $Q^T X Q$ is the projection of $V^T D V$ on the negative semidefinite cone and the other blocks are equal to the corresponding blocks of $Q^T D Q$. The solution of (11) is therefore

$$\Pi_{D_0^{p}}(D) = Q \left[ \begin{bmatrix} -\Pi_{S^p}(-V^T D V) & V^T D e \\ e^T D v & e^T D e \end{bmatrix} Q^T, e^T D e \right].$$
where $-\Pi_{S_p^+}(-V^T DV)$ is the projection of $V^T DV$ on the negative semidefinite cone, obtained by replacing the positive eigenvalues of $V^T DV$ by zero.

The method in [26] computes the projection indirectly, via the projection on the dual cone of $D_0^p$ and the formula

$$\Pi_{D_0^p}(D) = D - \Pi_{-(D_0^p)^*}(D).$$

(12)

The dual cone is $(D_0^p)^* = \{VZV^T \mid Z \preceq 0\}$, so the projection of $D$ on $-(D_0^p)^*$ is the solution of

$$\begin{align*}
\text{minimize} & \quad \|VZV^T - D\|_F^2 \\
\text{subject to} & \quad Z \succeq 0.
\end{align*}$$

Equivalently, since $Q$ is orthogonal, we solve

$$\begin{align*}
\text{minimize} & \quad \|Q^T(VZV^T - D)Q\|_F^2 = \left\|\begin{bmatrix}
Z - V^T DV & -V^T De \\
-e^T DV & -e^T De
\end{bmatrix}\right\|_F^2 \\
\text{subject to} & \quad Z \succeq 0.
\end{align*}$$

The solution is $Z = \Pi_{S_{p-1}^+}(V^T DV)$ and substituting in (12) gives

$$\Pi_{D_0^p}(D) = D - V\Pi_{S_{p-1}^+}(V^T DV)V^T.$$  

4 Projection via dual decomposition

In this section we present algorithms for the primal and dual problems (3) and (4) when $\eta = \{1, 2, \ldots, n\}$ and $P_\eta = I$. The primal and dual problems are

$$\begin{align*}
\text{minimize} & \quad \|x - a\|^2 \\
\text{subject to} & \quad P_{\gamma_k} x \in C_k, \quad k = 1, \ldots, l
\end{align*}$$

(13)

and

$$\begin{align*}
\text{maximize} & \quad -\|\sum_{k=1}^l P_{\gamma_k}^T z_k + a\|^2 + \|a\|^2 \\
\text{subject to} & \quad z_k \in C_k^*, \quad k = 1, \ldots, l.
\end{align*}$$

(14)

As explained in the previous section, the three sparse matrix nearness problems can be written in this form when $E' = E$ and $E$ is a chordal sparsity pattern.

There is a rich literature on decomposition methods for computing the projection on an intersection of closed convex sets via a sequence of projections on each of the sets separately; see, for example, [24, 31, 58, 59] and the books [12, 18]. We will discuss two approaches based on duality. In the first approach (section 4.2) the gradient projection method [43, §7.2.1] or accelerated gradient projection method [41, 42, 9] are applied to the dual problem. These algorithms can be viewed as applications of Tseng’s alternating minimization method for minimizing a strongly convex function over the intersection of convex sets [58, 59, 10]. The second approach (section 4.3) is the dual block coordinate ascent method [60]. This method can be interpreted as a generalization of Dykstra’s cyclic projection algorithm [24, 17] or Han’s successive projection algorithm [30], and also as a dual block coordinate gradient projection method [11]. In the next section we first review some duality properties that underlie the two approaches.
4.1 Dual decomposition

For notational convenience, we introduce the notation

\[ P = [P_{\gamma_1}^T \ P_{\gamma_2}^T \ \cdots \ P_{\gamma_l}^T]^T, \quad C = C_1 \times C_2 \times \cdots \times C_l, \quad C^* = C_1^* \times C_2^* \times \cdots \times C_l^*. \]

With this notation we can write (13) and (14) as

\[
\begin{align*}
\text{minimize} & \quad \|x - a\|^2 \\
\text{subject to} & \quad Px \in C
\end{align*}
\]

\[
\begin{align*}
\text{maximize} & \quad -\|P^Tz + a\|^2 + \|a\|^2 \\
\text{subject to} & \quad z \in C^*
\end{align*}
\]

where \( x = (x_1, x_2, \ldots, x_l) \) and \( z = (z_1, z_2, \ldots, z_l) \). Note that the matrix \( P^TP \) is diagonal and that the \( j \)th diagonal entry is the number of index sets \( \gamma_k \) that contain the index \( j \):

\[(P^TP)_{jj} = |\{k \mid j \in \gamma_k\}|, \quad j = 1, \ldots, n. \quad (15)\]

The dual decomposition methods we discuss in the next two sections are descent methods for minimizing

\[ f(z) = \frac{1}{2}\|P^Tz + a\|^2 = \frac{1}{2}\|\sum_{k=1}^{l} P_{\gamma_k}^Tz_k + a\|^2 \quad (16) \]

over the product cone \( C^* = C_1^* \times C_2^* \times \cdots \times C_l^* \). The methods generate a sequence of dual feasible suboptimal points \( z \). From a dual feasible \( z \in C^* \), approximate primal and dual projections \( x \) and \( s \) are computed as \( x = P^Tz + a \) and \( s = P^Tz \). The distances to optimality \( \|x - x^*\| \) and \( \|s - s^*\| \) can be bounded in terms of the dual suboptimality \( f(z) - f(z^*) \). To see this, we note that for any dual optimal solution \( z^* \) and any \( z \in C^* \), we have

\[
\begin{align*}
\|P^Tz - P^Tz^*\|^2 & = \|P^Tz + a\|^2 - \|P^Tz^* + a\|^2 - 2(Pz^* + a)^TP^T(z - z^*) \\
& \leq \|P^Tz + a\|^2 - \|P^Tz^* + a\|^2.
\end{align*}
\]

The inequality holds because

\[
\nabla f(z^*)^T(z - z^*) = (P^Tz^* + a)^TP^T(z - z^*) \geq 0
\]

for all \( z \in C^* \) if \( z^* \) is optimal. Hence, if \( z \) is dual feasible and we define \( s = P^Tz, \ x = P^Tz + a, \) then

\[
\begin{align*}
\frac{1}{2}\|x - x^*\|^2 & = \frac{1}{2}\|s - s^*\|^2 = \frac{1}{2}\|P^T(z - z^*)\|^2 \\
& \leq \frac{1}{2}\|P^Tz + a\|^2 - \frac{1}{2}\|P^Tz^* + a\|^2 \\
& = f(z) - f(z^*).
\end{align*}
\]

This simple inequality allows us to translate convergence results for the dual objective function to convergence rates for the primal and dual distances to optimality.
4.2 Gradient projection method

The gradient projection method for minimizing the function (16) over $C^*$ uses the iteration

$$
\begin{align*}
z^i &= \Pi_{C^*} (z^{i-1} - t \nabla f(z^{i-1})) \\
     &= \Pi_{C^*} (z^{i-1} - tP(P^T z^{i-1} + a)),
\end{align*}
$$

(17)

where $t$ is a positive step size. In terms of the block vectors,

$$
\begin{align*}
z^i_k &= \Pi_{C^*_k} (z^{i-1}_k - tP \sum_{j=1}^{l} \gamma_j z^i_{j} + a) \\
     &= \Pi_{C^*_k} ((1 - t)z^{i-1}_k - tP \sum_{j \neq k} \gamma_j z^i_{j} + a),
\end{align*}
$$

On line 2 we use the fact that $P \gamma_k P^T = I$. The projections on $C^*_k$ can also be expressed in terms of projections on $C_k$ via the identity $u = \Pi_{C_k}(u) - \Pi_{C^*_k}(-u)$.

A standard convergence result states that a fixed step size $t = 1/L$ can be used, where $L$ is the largest eigenvalue of the Hessian $\nabla^2 f(z) = PP^T$:

$$
L = \lambda_{\text{max}}(PP^T) = \lambda_{\text{max}}(P^T P) = \max_{j=1, \ldots, n} |\{k \mid j \in \gamma_k\}|.
$$

(18)

(Recall that $P^T P$ is diagonal; see (15).) With this step size the sequence $z^i$ converges to a minimizer of $f$ over $C^*$, even when the minimizers are not unique [43, page 207]. Moreover the dual optimality gap decreases as

$$
f(z^i) - f(z^*) \leq \frac{L}{2t} \|z^0 - z^*\|^2,
$$

where $z^*$ is any optimal solution [9, theorem 3.1]. From (17) it then follows that the sequences $x^i = P^T z^i + a$ and $s^i = P^T z^i$ converge to the projections $x^*$ and $s^*$ at a rate $O(1/\sqrt{i})$:

$$
\|x^i - x^*\| = \|s^i - s^*\| \leq \sqrt{\frac{L}{i}} \|z^0 - z^*\|.
$$

The gradient projection algorithm is summarized in Algorithm 1. It is important to keep in mind that the projection on line 4 reduces to $l$ projections on $C^*_k$ or $C_k$, and can be computed in parallel:

$$
z^i_k = \Pi_{C^*_k}(y^i_k) = y^i_k + \Pi_{C_k}(-y^i_k), \quad k = 1, \ldots, l.
$$

The stopping condition of the algorithm is left open for now and will be discussed at the end of this section.

An important advantage of the gradient projection method is the availability of Nesterov-type accelerations [41, 42, 61, 9]. We will use Nesterov’s first accelerated gradient projection method [41], which was later generalized to an accelerated proximal gradient method in [9] and is widely known under the acronym FISTA used in [9]; see also [61]. FISTA applies a gradient projection update after an extrapolation step:

$$
z^i = \Pi_{C^*}(v^i - t \nabla f(v^i)) \quad \text{where} \quad v^i = z^{i-1} + \frac{i - 2}{i + 1} (z^{i-1} - z^{i-2})
$$
Algorithm 1 Gradient projection method for problems (13) and (14).

1. Initialize: Set \( t = 1/L \) with \( L \) defined in (18). Choose an initial \( z^0 = (z_0^0, \ldots, z_l^0) \), and take \( s^0 = P^T z^0 \) and \( x^0 = s^0 + a \).

2. for \( i = 1, 2, \ldots \) until convergence do
   3. Gradient step. Compute \( y^i = z^{i-1} - tPx^{i-1} \).
   4. Projection step. Compute \( z^i = \Pi_{C^*}(y^i) \), \( s^i = P^T z^i \), and \( x^i = s^i + a \).
   5. end for

(with the assumption \( z^{-1} = z^0 \), so the first iteration is the standard gradient projection update). The same step size \( 1/L \) is used as in the gradient projection method. The accelerated gradient projection algorithm has the same complexity per iteration as the basic algorithm.

As for the gradient projection method the iterates \( z^i \) can be shown to converge, even when the optimal solutions are not unique. This is discussed in the recent paper [19]. The dual optimality gap decreases as \( 1/i^2 \):

\[
 f(z^i) - f(z^*) \leq \frac{2L}{(i+1)^2} \|z^0 - z^*\|^2,
\]

where \( z^* \) is any optimal solution [9, theorem 4.4]. By the same argument as used for the gradient projection method, this leads to the following convergence result for the sequences \( x^i = P^T z^i + a \) and \( s^i = P^T z^i \):

\[
 \|x^i - x^*\| = \|s^i - s^*\| \leq \frac{2\sqrt{L}}{i+1} \|z^0 - z^*\|.
\]

The \( O(1/i) \) convergence rate is an improvement over the \( O(1/\sqrt{i}) \) rate of the gradient projection method. The method is summarized in Algorithm 2.

Algorithm 2 Fast gradient projection method for problems (13) and (14).

1. Initialize: Set \( t = 1/L \) with \( L \) defined in (18). Choose an initial \( z^{-1} = z^0 = (z_0^0, \ldots, z_l^0) \), and take \( s^0 = P^T z^0 \) and \( x^0 = s^0 + a \).

2. for \( i = 1, 2, \ldots \) until convergence do
   3. Extrapolation step. Compute \( v^i = z^{i-1} + \frac{i-2}{i+1} (z^{i-1} - z^{i-2}) \).
   4. Gradient step. Compute \( y^i = v^i - tP(P^Tv^i + a) \).
   5. Projection step. Compute \( z^i = \Pi_{C^*}(y^i) \), \( s^i = P^T z^i \), and \( x^i = s^i + a \).
   6. end for

Various stopping criteria can be used for Algorithms 1 and 2. For example, one can bound the error with which the iterates satisfy the optimality condition \( z = \Pi_{C^*}(z - t\nabla f(z)) \) for problem (16), where \( t \) is any positive number. In the gradient projection algorithm (Algorithm 1) we have

\[
 z^{i-1} = \Pi_{C^*}(z^{i-1} - t\nabla f(z^{i-1})) + r^i
\]

with \( r^i = z^{i-1} - z^i \). In the fast gradient projection algorithm (Algorithm 2) we have

\[
 v^i = \Pi_{C^*}(v^i - t\nabla f(v^i)) + r^i
\]
with $r^i = v^i - z^i$. This suggests using stopping conditions
\[
\frac{\|z^i - z^{i-1}\|}{\max\{\|z^i\|, 1\}} \leq \epsilon, \quad \frac{\|v^i - z^i\|}{\max\{\|z^i\|, 1\}} \leq \epsilon
\]
for the proximal gradient method and the fast proximal gradient method, respectively.

### 4.3 Dual block coordinate ascent

Dykstra’s algorithm [24, 17], also known as the successive projection algorithm [30], is a method for computing the Euclidean projection of a point on the intersection of convex sets, via a cyclic sequence of projections on each of the sets separately. It can be interpreted as a dual block-coordinate ascent algorithm [30, 26, 60].

Dykstra’s method has been applied to several types of dense matrix nearness problems in the literature. The problem of finding the nearest EDM matrix to a given matrix is discussed in [27] and [26, section 5.3]. The approach taken in these papers is to formulate the projection on $D_n$ and $\{X \in S^n \mid \text{diag}(X) = 0\}$. The nearest correlation matrix problem, i.e., projection on the intersection of $S^n_+$ and $\{X \in S^n \mid \text{diag}(X) = 1\}$, is discussed in [37, 40, 35].

Following the dual interpretation of Dykstra’s method we apply block-coordinate ascent to the dual problem (14). If we fix $z_j$ for $j \neq k$ and minimize $f(z)$ over $z_k$ only, the problem reduces to

\[
\text{minimize} \quad \|z_k + P_{\gamma_k}(\sum_{j \neq k} P_{\gamma_j}^T z_j + a)\|^2 \\
\text{subject to} \quad z_k \in C_k^*.
\]

This is a Euclidean projection of the point $w = -P_{\gamma_k}(\sum_{j \neq k} P_{\gamma_j}^T z_j + a)$ on $C_k^*$. The unique solution can be expressed in two equivalent forms:

\[
z_k = \Pi_{C_k^*}(w) = \Pi_{C_k^*}(-w) + w. \tag{19}
\]

Alternatively, we can view this as a block coordinate gradient projection update. The gradient of $f(z)$ with respect to $z_k$ is

\[
\nabla_{z_k} f(z) = P_{\gamma_k}(\sum_{j=1}^l P_{\gamma_j}^T z_j + a) = z_k + P_{\gamma_k}(\sum_{j \neq k} P_{\gamma_j}^T z_j + a),
\]

because $P_{\gamma_k} P_{\gamma_k}^T = I$. The vector $w$ is therefore equal to $w = z_k - \nabla_{z_k} f(z)$ and $\Pi_{C_k^*}(w)$ is a block coordinate gradient projection step with step size one.

In Algorithm 3 we minimize over $z_1, \ldots, z_l$ cyclically, using the second expression in (19). We also maintain a primal variable $x^i = \sum_{j=1}^l P_{\gamma_j}^T z_j + a$. The simplest initialization is to take $z^0 = 0$ and $x^0 = a$. On line 4 we project the point

\[
P_{\gamma_k}(\sum_{j \neq k} P_{\gamma_j}^T z_j^{i-1} + a) = P_{\gamma_k} x^{i-1} - z_k^{i-1}
\]
on $C_k$. The update on line 5 is easier to describe in words: $x^i$ is equal $x^{i-1}$ with the subvector $x_{\gamma_k}^{i-1}$ replaced by $v$. Line 6 can also be written as $z_k^i = z_k^{i-1} + v - P_{\gamma_k} x^{i-1}$. Hence

\[
z_k^i = z_k^{i-1} + v - P_{\gamma_k}(\sum_{j=1}^l P_{\gamma_j}^T z_j^{i-1} + a) = v - P_{\gamma_k}(\sum_{j \neq k} P_{\gamma_j}^T z_j^{i-1} + a).
\]
Algorithm 3 Dual block coordinate ascent for problems (13) and (14).

1. **Initialize**: Choose an initial \((z_1^0, \ldots, z_l^0) \in \mathcal{C}\) and set \(x^0 = P^T z^0 + a\).

2. **for** \(i = 1, 2, \ldots \) until convergence **do**
   
   3. **Select the next index**: \(k = (i - 1) \mod l + 1\).
   
   4. **Projection step**: Compute \(v = \Pi_{C_k}(P_{\gamma_k} x^{i-1} - z_{k}^{i-1})\).
   
   5. **Update the primal variable**: Compute \(x^i = P_{\gamma_k} v + (I - P_{\gamma_k} P_{\gamma_k}) x^{i-1}\).
   
   6. **Update dual variables**: Compute \(z_k^i = z_k^{i-1} + P_{\gamma_k} (x^i - x^{i-1})\) and define \(z_j^i = z_j^{i-1}\) for \(j \neq k\).

7. **end for**

When \(P_{\gamma_k} = I\) for \(k = 1, \ldots, l\), and with the initialization \(x^0 = a, z^0 = 0\), this is Dykstra’s algorithm for computing the projection on \(\bigcap_{k=1}^l K_k\) [24]. Algorithm 3 is a special case of Tseng’s dual block coordinate ascent algorithm [60, §3] and convergence follows from [60, theorem 3.1]. We can also give a convergence rate by applying a recent result by Beck and Tetruashvili on block coordinate gradient projection algorithms [11, theorem 6.3]: we have \(f(z^i) - f(z^*) \leq c / i\), where \(c\) is a constant. It then follows from (17) \(x^i\) and \(s^i = P^T z^i\) satisfy \(\|x^i - x^*\| = \|z^i - x^*\| = O(1/\sqrt{i})\).

In general, deciding when to terminate the block coordinate ascent method can be difficult, since the iterates may remain constant for several successive iterations. The stopping condition proposed in [49, eq. (13)] is based on measuring the residual \(r^j = z^j - z^{(j-1)l}\), i.e., the difference between the values of \(z\) at the end of two successive cycles. We will use a similar stopping condition of the form

\[
\frac{\|z^j - z^{(j-1)l}\|}{\max\{\|z^j\|, 1\}} \leq \epsilon.
\]

It can be shown that if \(r^j = 0\) then the iterates \(x^j\) have remained constant during cycle \(j\). This can be seen from the expression

\[
\|r^j\|^2 = \sum_{k=1}^l \|z_k^{(j-1)l + k} - z_k^{(j-1)l + k-1}\|^2 = \sum_{k=1}^l \|x^{(j-1)l + k} - x^{(j-1)l + k-1}\|^2.
\]

The first step follows because \(z_k\) changes only in iteration \(k\) of cycle \(j\); the second step follows from line 6 in the algorithm. This observation implies that \(x^j\) and \(z^j\) are optimal if \(r^j = 0\). By construction, \(x^i - a = P^T z^i\) after each step of the algorithm. Additionally, if \(k = (i - 1) \mod l + 1\) then the two relations \(z_k^i = \Pi_{C_k^j}(w)\) and \(P_{\gamma_k} x^i = -\Pi_{C_k}(w)\) hold, where \(w = -P_{\gamma_k} (\sum_{j \neq k} P_{\gamma_j} z_j^i + a)\).

This shows that \(z^i\) is always dual feasible. Moreover if \(x^i\) remains unchanged for an entire cycle then it is primal feasible and \((z_k^i)^T P_{\gamma_k} x^i = 0\) for \(k = 1, \ldots, l\).

5 Projection via Douglas-Rachford splitting

We now turn to the general problem (3) for arbitrary index sets \(\eta\). The assumption that \(\eta = \{1, 2, \ldots, n\}\) and \(P_\eta = I\) was crucial in the dual decomposition approaches of the previous section 4. When \(\eta \neq \{1, 2, \ldots, n\}\) the dual problem (5) includes a coupling equality constraint and the separable structure exploited by the dual gradient projection and block-coordinate ascent methods...
is no longer available. In this section, we present decomposition methods based on the Douglas-Rachford splitting method [39, 25, 8].

5.1 Douglas-Rachford algorithm

The Douglas-Rachford method is a popular method for minimizing a sum \( f(\tilde{x}) = g(\tilde{x}) + h(\tilde{x}) \) of two closed, convex, and possibly nonsmooth functions \( g \) and \( h \). In our applications, the second function will be the indicator function of a subspace \( V \), i.e., we solve

\[
\begin{align*}
\text{minimize} & \quad g(\tilde{x}) \\
\text{subject to} & \quad \tilde{x} \in V.
\end{align*}
\]

(20)

The Douglas-Rachford method specialized to this problem is also known as the method of partial inverses [54, 55, 25]. The algorithm starts at an arbitrary \( \tilde{z}_0 \) and repeats the following iteration:

\[
\begin{align*}
\tilde{x}^{i+1} &= \text{prox}_{tg}(\tilde{z}^i) \\
\tilde{y}^{i+1} &= \Pi_V(2\tilde{x}^{i+1} - \tilde{z}^i) \\
\tilde{z}^{i+1} &= \tilde{z}^i + \rho(\tilde{y}^{i+1} - \tilde{x}^{i+1}).
\end{align*}
\]

(21)

There are two algorithm parameters: a stepsize \( t > 0 \) and a relaxation parameter \( \rho \in (0, 2) \). The function \( \text{prox}_{tg} \) in the first step is the proximal operator of \( tg(x) \), defined as

\[
\text{prox}_{tg}(z) = \arg\min_u \left( g(u) + \frac{1}{2t}\|u - z\|^2 \right).
\]

In this paper, only two types of proximal operators will be encountered. If \( g(\tilde{x}) = \delta_S(\tilde{x}) \) is the indicator function of a closed convex set \( S \), then \( \text{prox}_{tg}(\tilde{x}) = \Pi_S(\tilde{x}) \), the Euclidean projection on \( S \). If \( g \) is a convex quadratic function \( g(\tilde{x}) = (1/2)\tilde{x}^TQ\tilde{x} + q^T\tilde{x} \), then

\[
\text{prox}_{tg}(\tilde{x}) = (I + tQ)^{-1}(\tilde{x} - tq).
\]

The Douglas-Rachford method (21) is of interest when the projection on \( V \) and the proximal operator of \( g \) are inexpensive.

The convergence of the Douglas-Rachford algorithm has been studied extensively. It can be shown that if \( g \) is a closed convex function, with \( V \cap \text{ri dom } g \neq \emptyset \), and the problem (20) has a solution, then the sequences \( \tilde{x}^i \) and \( \tilde{y}^i \) converge to a solution [8, corollary 27.2]. Convergence rates are usually expressed in terms of fixed point residuals. Recent results by Davis and Yin [20, theorem 7] also give a convergence rate for the objective value: it is shown that \( |f(\tilde{x}^i) - f(\tilde{x}^*)| = o(1/\sqrt{i}) \). The bound can be improved if \( g \) satisfies additional properties, e.g., if its gradient is Lipschitz continuous [21].

We will use the simple stopping condition described in [56]:

\[
\frac{\|r_p^i\|}{\max\{\|\tilde{x}^i\|, 1\}} \leq \epsilon_p, \quad \frac{\|r_d^i\|}{\max\{\|\tilde{s}^i\|, 1\}} \leq \epsilon_d,
\]

for some primal and dual tolerances \( \epsilon_p \) and \( \epsilon_d \), where \( \tilde{s} = t^{-1}(\tilde{z}^{i-1} - \tilde{x}^i) \) and

\[
r_p^i = \Pi_V(\tilde{x}^i) - \tilde{x}^i, \quad r_d^i = -\Pi_V(\tilde{s})
\]

are primal and dual residuals.
5.2 Primal splitting

To apply the Douglas-Rachford algorithm to the primal problem (3), we first reformulate it as

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|P_\eta x - a\|^2 + \sum_{k=1}^l \delta_{C_k}(y_k) \\
\text{subject to} & \quad P_\gamma x = y_k, \quad k = 1, \ldots, l.
\end{align*}
\]

(22)

The variables are \(x \in \mathbb{R}^n\) and an additional splitting variable \(y = (y_1, y_2, \ldots, y_l)\). This problem has the form (20) if we take \(\tilde{x} = (x, y)\),

\[
g(x, y) = \frac{1}{2} \|P_\eta x - a\|^2 + \sum_{k=1}^l \delta_{C_k}(y_k), \quad V = \{(x, y) \mid y = Px\}.
\]

The function \(g\) is separable with proximal operator

\[
\text{prox}_{tg}(x, y) = \begin{bmatrix} \frac{(I + t P_\eta^T P_\eta)^{-1}(x + tP_\eta a)}{\Pi_C(y)} \end{bmatrix}.
\]

(23)

Note that the inverse in the first block is the inverse of a strictly positive diagonal matrix, since \(P_\eta^T P_\eta\) is diagonal with \((P_\eta^T P_\eta)_{ii} = 1\) if \(i \in \eta\) and \((P_\eta^T P_\eta)_{ii} = 0\) otherwise. The projection on \(C\) in the second block reduces to \(l\) independent projections \(\Pi_C(y) = (\Pi_{C_1}(y_1), \ldots, \Pi_{C_l}(y_l))\). The projection on the subspace \(V\) in (5.2) is

\[
\Pi_V(x, y) = \begin{bmatrix} I \\ P \end{bmatrix} (I + P^T P)^{-1}(x + P^T y),
\]

(24)

which is also simple to compute since \(P^T P\) is diagonal; see (15).

A summary of the Douglas-Rachford method is given in Algorithm 4.

**Algorithm 4** Douglas-Rachford method for primal problem (3).

1. **Initialize:** Choose parameters \(t > 0, \rho \in (0, 2)\), initial \(u^0, v_1^0, \ldots, v_l^0\).
2. **for** \(i = 1, 2, \ldots\) until convergence **do**
   3. Compute \((x^i, y^i) = \text{prox}_{tg}(u^{i-1}, v^{i-1})\) using (23).
   4. Compute \((w^i, z^i) = \Pi_V(2x^i - u^{i-1}, 2y^i - v^{i-1})\) using (24).
   5. Update \(u^i = u^{i-1} + \rho(w^i - x^i)\) and \(v_k^i = v_k^{i-1} + \rho(z_k^i - y_k^i), \quad k = 1, \ldots, l\).
3. **end for**

The algorithm is easily extended to generalizations of problems (22) with simple equality constraints \(Ax = b\) added. An example is the constraint \(\text{diag}(X) = 1\) in (6) to compute the projection on the matrices with a correlation matrix completion. The additional constraint can be handled by adding the indicator function \(\delta_{\{0\}}(Ax - b)\) to the objective function of (22). The proximal operator of

\[
\frac{1}{2} \|P_\eta x - a\|^2 + \delta_{\{0\}}(Ax - b)
\]

is a weighted projection on an affine set and is inexpensive if \(A\) has few rows.
5.3 Dual splitting

To apply the Douglas-Rachford method to the dual problem (4) we write it as

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|s + a\|^2 + \sum_{k=1}^l \delta C^*_k(z_k) \\
\text{subject to} & \quad \sum_{k=1}^l P^T_{\eta k} z_k = P^T_{\eta} s.
\end{align*}
\] (25)

This is in the form (20) with \( \tilde{x} = (s, z) \),

\[
g(s, z) = \frac{1}{2} \|s + a\|^2 + \sum_{k=1}^l \delta C^*_k(z_k), \quad \mathcal{V} = \{(s, z) \mid P^T_{\eta} s = P^T_{\eta} z\}.
\]

The proximal operator of \( g \) is

\[
\text{prox}_{t g}(s, z) = \begin{bmatrix} (1 + t)^{-1}(s - ta) \\ \Pi_{C^*}(z) \end{bmatrix}.
\] (26)

The projection on \( \mathcal{V} \) can be written as

\[
\Pi_{\mathcal{V}}(s, z) = \begin{bmatrix} s \\ z \end{bmatrix} + \begin{bmatrix} -P_{\eta} \\ P \end{bmatrix} (P^T_{\eta} P + P^T P)^{-1} (P^T_{\eta} s - P^T_{\eta} z).
\] (27)

**Algorithm 5** Douglas-Rachford method for dual problem (4).

1. Initialize: Choose parameters \( t > 0, \rho \in (0, 2), \) initial \( u^0, v^0_1, \ldots, v^0_l \).
2. for \( i = 1, 2, \ldots \) until convergence do
   3. Compute \((s^i, z^i) = \text{prox}_{t g}(u^{i-1}, v^{i-1}) \) using (26).
   4. Compute \((w^i, y^i) = \Pi_{\mathcal{V}}(2s^i - u^{i-1}, 2z^i - v^{i-1}) \) using (27).
   5. Update \( u^i = u^{i-1} + \rho(w^i - s^i) \) and \( v^i_k = v^{i-1}_k + \rho(y^i_k - z_k) \), \( k = 1, \ldots, l \).
3. end for

6 Numerical results

In this section we test the proposed algorithms on problems with sizes ranging from 1000 to 100,000. The problems in the first set of experiments are constructed from thirteen symmetric sparsity patterns in the University of Florida sparse matrix collection [23]. In the second set of experiments, we consider a family of randomly generated sensor network node localization problems [13, 38, 3].

The experiments are performed on an Intel Xeon CPU E31225 processor with 32 GB RAM, running Ubuntu 14.04 (Trusty) and using MATLAB version 8.3.0 (2014a). In the dual gradient projection and Douglas-Rachford algorithms the main step per iteration is the projection on \( l \) dense PSD or EDM cones, and these projections can be done in parallel. In the dual block coordinate ascent method, some of the projections can be computed in parallel, if they are scheduled using a topological ordering on a clique tree [14]. Since the projections are the most expensive part of the algorithms, exploiting parallelism would result in a significant speedup. This possibility was not exploited in the code used for the experiments, which computes the projections sequentially.
Table 1: Thirteen symmetric sparsity patterns from the University of Florida sparse matrix collection. For each pattern we give the matrix order $p$ and the density, defined as $(p + 2|E|)/p^2$.

<table>
<thead>
<tr>
<th>pattern</th>
<th>$p$</th>
<th>density</th>
</tr>
</thead>
<tbody>
<tr>
<td>ex4</td>
<td>1601</td>
<td>1.24e-2</td>
</tr>
<tr>
<td>c-26</td>
<td>4307</td>
<td>1.86e-3</td>
</tr>
<tr>
<td>g3rm3m3</td>
<td>5357</td>
<td>7.24e-3</td>
</tr>
<tr>
<td>barth4</td>
<td>6019</td>
<td>1.13e-3</td>
</tr>
<tr>
<td>c-37</td>
<td>8294</td>
<td>1.11e-3</td>
</tr>
<tr>
<td>tuna2</td>
<td>12992</td>
<td>2.92e-4</td>
</tr>
<tr>
<td>crack_dual</td>
<td>20141</td>
<td>1.98e-4</td>
</tr>
<tr>
<td>biplane-9</td>
<td>21701</td>
<td>1.79e-4</td>
</tr>
<tr>
<td>mario001</td>
<td>38434</td>
<td>1.39e-4</td>
</tr>
<tr>
<td>c-60</td>
<td>43640</td>
<td>1.57e-4</td>
</tr>
<tr>
<td>c-67</td>
<td>57975</td>
<td>1.58e-4</td>
</tr>
<tr>
<td>rail_79841</td>
<td>79841</td>
<td>8.69e-5</td>
</tr>
<tr>
<td>luxembourg_osm</td>
<td>114599</td>
<td>1.82e-5</td>
</tr>
</tbody>
</table>

6.1 Chordal sparse matrix cones

We consider the three matrix nearness problems in sections 3.2–3.4 with chordal patterns $E$. The patterns are chordal extensions of thirteen nonchordal sparsity patterns from the University of Florida sparse matrix collection [23]. Table 1 gives some statistics for the patterns before the extension. The sparsity graphs of three of the larger patterns (mario001, c-60, c-67) are not connected, but since the largest connected component contains almost all the vertices (as shown in the table), we did not remove the smaller connected components.

The chordal extensions are computed in two steps. We first use graph elimination (or symbolic Cholesky factorization) using a fill-reducing reordering (the MATLAB amd reordering) to generate a first chordal extension. We then merge some of the smaller cliques of this extension according to heuristics discussed in [56, §6.2]. Table 2 gives the statistics of the two chordal extensions, before and after clique merging.

For each pattern $E$, we generate five instances $B$ with lower triangular nonzeros generated from independent normalized Gaussian distributions. The numerical results in the following tables are averages over the five instances.

We first consider projections on the PSD and PSD completable cones. When $E$ is a chordal pattern, the formulation (8) of the projection of $B$ on $\Pi_E(S_p^E)$ simplifies to

\[
\begin{align*}
\text{minimize} & \quad \|X - B\|_F^2 \\
\text{subject to} & \quad P_{\beta_k} X P_{\beta_k}^T \succeq 0, \quad k = 1, \ldots, m,
\end{align*}
\]

where $\beta_1, \ldots, \beta_m$ are the cliques of the sparsity pattern. The dual of this problem is

\[
\begin{align*}
\text{maximize} & \quad -\left\| \sum_{k=1}^m P_{\beta_k}^T Z_k P_{\beta_k} + B \right\|_F^2 + \|B\|_F^2 \\
\text{subject to} & \quad Z_k \succeq 0, \quad k = 1, \ldots, m,
\end{align*}
\]

and is equivalent to the projection of $-B$ on $S^E_p \cap S^p_+$, i.e., problem (6) for $A = -B$. We apply the fast projected gradient and block coordinate ascent methods to the dual problem (29), and the
Table 2: Chordal extensions of the patterns of table 1. The table shows the density, the number of cliques \( m \), and the average and maximum clique size, after a chordal extension and after a further clique merging step.

<table>
<thead>
<tr>
<th>( p )</th>
<th>density</th>
<th>( m )</th>
<th>avg. clique size</th>
<th>max. clique size</th>
<th>after extension</th>
<th>density</th>
<th>( m )</th>
<th>avg. clique size</th>
<th>max. clique size</th>
<th>after clique merging</th>
</tr>
</thead>
<tbody>
<tr>
<td>1601</td>
<td>3.24e-2</td>
<td>598</td>
<td>18.0</td>
<td>74</td>
<td>4.94e-2</td>
<td>94</td>
<td>46.3</td>
<td>74</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4307</td>
<td>3.69e-3</td>
<td>3740</td>
<td>6.1</td>
<td>66</td>
<td>1.07e-2</td>
<td>556</td>
<td>17.4</td>
<td>66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5357</td>
<td>2.97e-2</td>
<td>577</td>
<td>52.9</td>
<td>261</td>
<td>3.27e-2</td>
<td>267</td>
<td>80.8</td>
<td>261</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6019</td>
<td>6.12e-3</td>
<td>3637</td>
<td>11.1</td>
<td>89</td>
<td>1.11e-2</td>
<td>317</td>
<td>42.2</td>
<td>89</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8204</td>
<td>4.05e-3</td>
<td>7158</td>
<td>7.1</td>
<td>257</td>
<td>9.54e-3</td>
<td>1121</td>
<td>21.5</td>
<td>257</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12992</td>
<td>2.90e-3</td>
<td>11051</td>
<td>5.7</td>
<td>241</td>
<td>5.22e-3</td>
<td>704</td>
<td>37.2</td>
<td>241</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20141</td>
<td>1.21e-3</td>
<td>17053</td>
<td>6.5</td>
<td>168</td>
<td>2.80e-3</td>
<td>1098</td>
<td>35.7</td>
<td>168</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21701</td>
<td>1.41e-3</td>
<td>16755</td>
<td>8.0</td>
<td>147</td>
<td>2.99e-3</td>
<td>1099</td>
<td>40.7</td>
<td>147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38434</td>
<td>5.36e-4</td>
<td>30917</td>
<td>6.0</td>
<td>188</td>
<td>1.25e-3</td>
<td>2365</td>
<td>28.1</td>
<td>188</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43640</td>
<td>1.37e-3</td>
<td>39468</td>
<td>6.2</td>
<td>954</td>
<td>2.56e-3</td>
<td>6175</td>
<td>19.5</td>
<td>954</td>
<td></td>
<td></td>
</tr>
<tr>
<td>57975</td>
<td>2.45e-4</td>
<td>52404</td>
<td>5.5</td>
<td>132</td>
<td>9.04e-4</td>
<td>8875</td>
<td>14.9</td>
<td>132</td>
<td></td>
<td></td>
</tr>
<tr>
<td>79841</td>
<td>5.31e-4</td>
<td>61059</td>
<td>8.7</td>
<td>337</td>
<td>9.71e-4</td>
<td>4247</td>
<td>44.4</td>
<td>337</td>
<td></td>
<td></td>
</tr>
<tr>
<td>114599</td>
<td>4.34e-5</td>
<td>113361</td>
<td>2.9</td>
<td>45</td>
<td>2.02e-4</td>
<td>7035</td>
<td>18.9</td>
<td>58</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Douglas-Rachford method to the primal problem (written in the form (22)) and the dual problem (in the form (25)).

The step size in the fast projected gradient method is \( t = 1/L \). In the Douglas-Rachford methods we used \( t = 1, \rho = 1.75 \). (A different choice of \( t \) and \( \rho \) may accelerate the Douglas-Rachford methods.) A tolerance \( \epsilon = 10^{-3} \) was used in the the various stopping conditions given in section 2. The algorithms were terminated when the CPU time exceeded a maximum four hours. The results are given in table 3. The runtimes are averages over five instances; we did not observe a significant variance in runtime or number of iterations for the different instances. To reduce computational overhead, the stopping conditions are tested every 25 iterations.

Table 4 shows the results of a similar experiment for the projection on the EDM completable cone (problem III) by solving (10), via the formulation

\[
\begin{align*}
\text{minimize} \quad & \| X - B \|^2 \\
\text{subject to} \quad & P_{\beta_k} XP_{\beta_k}^T \in D_{0}^{[\beta_k]}, \quad k = 1, \ldots, m \\
& \text{diag}(X) = 0
\end{align*}
\]  

(30)

where \( \beta_1, \ldots, \beta_m \) are the cliques in the chordal pattern \( E \). The strictly lower-triangular nonzero values of \( B \) are assigned according to a uniform distribution in \([0, 1]\). The diagonal of \( B \) is set to zero.

In all cases when the method converged, the final objective values for the different algorithms are equal to two or three significant digits. (The unaccelerated projected gradient method in general took much longer than all other methods to converge, and the results are not included.) In general, the fast projected gradient method converged more slowly than the other methods. In all but four instances, the dual fast projected gradient method took between 200 and 1000 iterations and in two instances exceeded the time limit. In comparison, the dual block coordinate ascent and Douglas-Rachford algorithms took between 25 and 150 iterations to converge. (For the block coordinate
<table>
<thead>
<tr>
<th>p</th>
<th>total runtime</th>
<th>time/iteration</th>
<th>F-PG</th>
<th>BCD</th>
<th>P-DR</th>
<th>D-DR</th>
<th>F-PG</th>
<th>BCD</th>
<th>P-DR</th>
<th>D-DR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1601</td>
<td>2.7e1</td>
<td>3.5</td>
<td>4.4</td>
<td>5.0</td>
<td>1.2e-1</td>
<td>1.4e-1</td>
<td>1.8e-1</td>
<td>2.0e-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4307</td>
<td>5.5e2</td>
<td>6.8</td>
<td>3.3e1</td>
<td>3.6e1</td>
<td>2.0e-1</td>
<td>2.7e-1</td>
<td>3.3e-1</td>
<td>3.6e-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5357</td>
<td>4.8e2</td>
<td>8.4e1</td>
<td>4.6e1</td>
<td>5.2e1</td>
<td>1.3</td>
<td>1.7</td>
<td>1.8</td>
<td>2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6019</td>
<td>1.2e2</td>
<td>9.6</td>
<td>1.3e1</td>
<td>1.6e1</td>
<td>3.4e-1</td>
<td>3.8e-1</td>
<td>5.1e-1</td>
<td>6.2e-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8204</td>
<td>2.6e3</td>
<td>7.1e1</td>
<td>1.1e2</td>
<td>1.1e2</td>
<td>9.5e-1</td>
<td>1.4</td>
<td>1.4</td>
<td>1.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12992</td>
<td>2.4e2</td>
<td>6.2e1</td>
<td>3.6e1</td>
<td>4.2e1</td>
<td>9.5e-1</td>
<td>1.2</td>
<td>1.4</td>
<td>1.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20141</td>
<td>2.5e2</td>
<td>3.9e1</td>
<td>3.8e1</td>
<td>4.6e1</td>
<td>9.9e-1</td>
<td>1.6</td>
<td>1.5</td>
<td>1.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21701</td>
<td>3.3e2</td>
<td>3.4e1</td>
<td>4.6e1</td>
<td>5.8e1</td>
<td>1.2</td>
<td>1.4</td>
<td>1.8</td>
<td>2.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38434</td>
<td>4.7e2</td>
<td>4.7e1</td>
<td>6.2e1</td>
<td>7.8e1</td>
<td>2.1</td>
<td>1.9</td>
<td>2.5</td>
<td>3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43640</td>
<td>&gt; 4hr</td>
<td>1.9e3</td>
<td>1.6e3</td>
<td>1.5e3</td>
<td>1.0e1</td>
<td>1.9e1</td>
<td>1.6e1</td>
<td>1.5e1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>57975</td>
<td>&gt; 4hr</td>
<td>1.4e2</td>
<td>1.1e3</td>
<td>1.1e3</td>
<td>3.5</td>
<td>5.7</td>
<td>6.4</td>
<td>6.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>79841</td>
<td>2.4e3</td>
<td>3.0e2</td>
<td>2.4e2</td>
<td>3.0e2</td>
<td>6.3</td>
<td>7.6</td>
<td>9.7</td>
<td>1.2e1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>114599</td>
<td>5.3e2</td>
<td>5.5e1</td>
<td>1.0e2</td>
<td>1.2e2</td>
<td>2.6</td>
<td>2.2</td>
<td>4.0</td>
<td>4.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Projection on chordal PSD completable matrices. CPU times (in seconds) for the projection on $\Pi_E(S^n_+^p)$. The total runtimes and time per iterations are given. The algorithms are: dual fast projected gradient method (F-PG), dual block coordinate ascent (BCD), primal Douglas-Rachford method (P-DR), and dual Douglas-Rachford method (D-DR).

<table>
<thead>
<tr>
<th>p</th>
<th>total runtime</th>
<th>time/iteration</th>
<th>F-PG</th>
<th>BCD</th>
<th>P-DR</th>
<th>D-DR</th>
<th>F-PG</th>
<th>BCD</th>
<th>P-DR</th>
<th>D-DR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1601</td>
<td>7.6e1</td>
<td>1.2e1</td>
<td>5.7</td>
<td>5.2</td>
<td>1.5e-1</td>
<td>1.6e-1</td>
<td>2.3e-1</td>
<td>2.1e-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4307</td>
<td>1.9e3</td>
<td>3.2e1</td>
<td>2.1e1</td>
<td>2.0e1</td>
<td>2.9e-1</td>
<td>4.1e-1</td>
<td>4.2e-1</td>
<td>3.9e-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5357</td>
<td>1.1e3</td>
<td>2.3e2</td>
<td>5.6e1</td>
<td>5.3e1</td>
<td>1.5</td>
<td>1.9</td>
<td>2.2</td>
<td>2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6019</td>
<td>3.9e2</td>
<td>2.7e1</td>
<td>1.7e1</td>
<td>1.6e1</td>
<td>4.4e-1</td>
<td>5.0e-1</td>
<td>6.9e-1</td>
<td>6.2e-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8204</td>
<td>8.6e3</td>
<td>2.7e2</td>
<td>8.5e1</td>
<td>8.3e1</td>
<td>1.2</td>
<td>2.7</td>
<td>1.7</td>
<td>1.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12992</td>
<td>6.8e2</td>
<td>1.8e2</td>
<td>4.5e1</td>
<td>4.1e1</td>
<td>1.2</td>
<td>1.5</td>
<td>1.8</td>
<td>1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20141</td>
<td>9.1e2</td>
<td>1.3e2</td>
<td>5.0e1</td>
<td>4.4e1</td>
<td>1.3</td>
<td>1.7</td>
<td>2.0</td>
<td>1.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21701</td>
<td>1.1e3</td>
<td>1.6e2</td>
<td>6.2e1</td>
<td>5.4e1</td>
<td>1.6</td>
<td>2.1</td>
<td>2.5</td>
<td>2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38434</td>
<td>1.4e3</td>
<td>8.6e2</td>
<td>8.3e1</td>
<td>7.2e1</td>
<td>2.1</td>
<td>6.1</td>
<td>3.3</td>
<td>2.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43640</td>
<td>&gt; 4hr</td>
<td>1.1e4</td>
<td>9.1e2</td>
<td>9.3e2</td>
<td>1.4e1</td>
<td>7.1e1</td>
<td>1.8e1</td>
<td>1.9e1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>57975</td>
<td>&gt; 4hr</td>
<td>4.6e3</td>
<td>5.5e2</td>
<td>5.6e2</td>
<td>4.8</td>
<td>3.7e1</td>
<td>7.3</td>
<td>7.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>79841</td>
<td>8.8e3</td>
<td>9.8e2</td>
<td>3.2e2</td>
<td>2.8e2</td>
<td>9.5</td>
<td>9.8</td>
<td>1.3e1</td>
<td>1.1e1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>114599</td>
<td>2.5e3</td>
<td>1.3e2</td>
<td>1.3e2</td>
<td>1.1e2</td>
<td>3.3</td>
<td>5.4</td>
<td>5.1</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Projection on chordal EDM completable matrices. CPU times (in seconds) for the projection onto $\Pi_E(D_p^n)$. The total runtimes and time per iterations are given. The algorithms are: dual fast projected gradient method (F-PG), dual block coordinate ascent (BCD), primal Douglas-Rachford method (P-DR), and dual Douglas-Rachford method (D-DR).
\[
\begin{array}{|c|c|c|c|c|c|}
\hline
p & \text{total runtime} & \text{time/iteration} \\
& \text{P-DR} & \text{D-DR} & \text{P-DR} & \text{D-DR} \\
\hline
1601 & 3.5e1 & 2.8e1 & 1.6e-1 & 1.6e-1 \\
4307 & 5.8e1 & 4.8e1 & 2.9e-1 & 2.9e-1 \\
5357 & 4.4e2 & 3.8e2 & 1.6 & 1.7 \\
6019 & 1.3e2 & 1.1e2 & 4.5e-1 & 4.8e-1 \\
8204 & 4.1e2 & 3.6e2 & 1.3 & 1.3 \\
12992 & 1.9e2 & 1.3e2 & 1.2 & 1.3 \\
20141 & 2.6e2 & 2.1e2 & 1.3 & 1.4 \\
21701 & 5.2e2 & 4.3e2 & 1.6 & 1.7 \\
38434 & 1.3e2 & 1.1e2 & 4.5e-1 & 4.8e-1 \\
43640 & 4.1e2 & 3.6e2 & 1.3e1 & 1.3e1 \\
57975 & 1.4e3 & 1.2e3 & 5.6 & 5.3 \\
79841 & 2.1e3 & 1.8e3 & 8.6 & 9.1 \\
114599 & 7.1e2 & 4.5e2 & 3.7 & 3.6 \\
\hline
\end{array}
\]

Table 5: Projection on nonchordal PSD completable matrices. CPU times (in seconds) for the projection on \(\Pi_E(S^P_E)\). The Douglas-Rachford method is applied to the primal and dual problem form (P-DR and D-DR). The total runtimes and time per iterations are given.

Ascent algorithm, we count one cycle through all \(l\) cones as one iteration.)

6.2 Nonchordal sparse matrix cones

In the next two sets of experiments we consider problems with nonchordal sparsity patterns. We first consider the projection on the PSD completable cone (problem (8)) and its dual

\[
\begin{align*}
\text{maximize} & \quad -\|S + B\|_F^2 + \|B\|_F^2 \\
\text{subject to} & \quad S = \sum_{k=1}^m P_{\beta_k}^T Z_k P_{\beta_k} \\
& \quad S_{ij} = 0, \quad \forall \{i,j\} \in E' \setminus E \\
& \quad Z_k \succeq 0, \quad k = 1, \ldots, m.
\end{align*}
\]

This also computes the projection on the PSD cone (problem (6)) of \(A = -B\). We use the patterns listed in table 1 as \(E\) and the chordal extensions listed in table 2 as \(E'\). For each sparsity pattern \(E\), we consider five randomly generated matrices \(B \in S_E^p\), with lower-diagonal nonzero values chosen from a normal Gaussian distribution. The results for the primal and dual Douglas-Rachford methods (Algorithms 4 and 5) are given in table 5.

Compared to the chordal problems (28), solving the nonchordal problems in general took more iterations (between 75 to 300 iterations). The final objective values for the primal and dual Douglas-Rachford methods are equal to around 2 or 3 significant digits.

In the next experiment we consider projections on EDM completable cones for a family of randomly generated sparsity graphs. A Euclidean distance matrix \(D \in \mathbb{D}^p\) is constructed by choosing \(p\) points uniformly in a three-dimensional cube. The sparsity pattern \(E\) is defined as \(E = \{\{i,j\} \mid i \neq j, \quad D_{ij} \leq R\}\), where \(R\) is a positive parameter. For some problems with \(p \geq 10,000\), this constructed in graphs that were not connected. However the largest connected component contained over 95% of the vertices, so we did not remove these instances.
Next we add noise to the nonzero elements of $D$ to construct a matrix $B \in \mathbb{S}^p_E$:

$$B_{ij} = \begin{cases} 
D_{ij} + N_{ij} & \{i, j\} \in E \\
0 & \text{otherwise}
\end{cases}$$

where for each $i, j$, the noise $N_{ij} = N_{ji}$ is drawn from a Gaussian distribution with mean zero and standard deviation 0.1. This type of problem is similar to sensor network node localization problems studied in [13, 38, 3]. We solve problem (30) using the Douglas-Rachford method for the primal formulation (Algorithm 4).

Table 6 gives the problem statistics and runtime results, averaged over five instances. In general, each problem converged after 200 to 400 iterations.

<table>
<thead>
<tr>
<th>$p$</th>
<th>range ($\bar{R}$)</th>
<th>density (extens.)</th>
<th>density (extens.)</th>
<th>avr. # cliques</th>
<th>avg. clique size</th>
<th>total runtime</th>
<th>time/iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>5.00e-2</td>
<td>3.67e-2</td>
<td>2.56e-1</td>
<td>46.2</td>
<td>122.8</td>
<td>1.5e2</td>
<td>6.9e-1</td>
</tr>
<tr>
<td>1000</td>
<td>1.00e-1</td>
<td>9.15e-2</td>
<td>4.39e-1</td>
<td>36.6</td>
<td>206.3</td>
<td>4.1e2</td>
<td>1.6</td>
</tr>
<tr>
<td>5000</td>
<td>1.00e-2</td>
<td>3.93e-3</td>
<td>6.89e-2</td>
<td>228.0</td>
<td>121.6</td>
<td>1.6e3</td>
<td>5.3</td>
</tr>
<tr>
<td>5000</td>
<td>2.00e-2</td>
<td>1.03e-2</td>
<td>1.58e-1</td>
<td>191.4</td>
<td>240.7</td>
<td>5.0e3</td>
<td>1.9e1</td>
</tr>
<tr>
<td>10000</td>
<td>5.00e-3</td>
<td>1.47e-3</td>
<td>3.20e-2</td>
<td>474.4</td>
<td>102.1</td>
<td>3.6e3</td>
<td>1.0e1</td>
</tr>
<tr>
<td>25000</td>
<td>2.00e-3</td>
<td>3.96e-4</td>
<td>9.22e-3</td>
<td>1277.4</td>
<td>71.3</td>
<td>6.4e3</td>
<td>1.8e1</td>
</tr>
<tr>
<td>50000</td>
<td>1.00e-3</td>
<td>1.48e-4</td>
<td>2.70e-3</td>
<td>2821.4</td>
<td>46.3</td>
<td>5.4e3</td>
<td>1.8e1</td>
</tr>
<tr>
<td>75000</td>
<td>6.67e-4</td>
<td>8.34e-5</td>
<td>1.17e-3</td>
<td>4779.4</td>
<td>32.9</td>
<td>3.7e3</td>
<td>1.6e1</td>
</tr>
<tr>
<td>100000</td>
<td>5.00e-4</td>
<td>5.56e-5</td>
<td>6.04e-4</td>
<td>7214.4</td>
<td>24.4</td>
<td>2.6e3</td>
<td>1.3e1</td>
</tr>
</tbody>
</table>

Table 6: Projection on EDM completable cone, with randomly generated nonchordal patterns. Problem statistics and CPU runtimes (total time and time per iteration, in seconds) for the primal Douglas-Rachford method are given.

7 Conclusion

We have presented decomposition methods for projections on sparse PSD, PSD completable, and EDM completable cones. By combining clique decomposition theorems for chordal sparsity patterns and first order convex optimization algorithms, we are able to solve large problems with sizes ranging from 1000 to 100,000. As mentioned briefly in section 5, it is straightforward to extend the algorithms, for example, to problems with a small set of linear equality constraints. This only requires an additional projection step on the affine set defined by the constraint equations.

The key feature of the algorithms is that they involve only small dense eigenvalue decompositions, corresponding to the cliques in the chordal extension of the sparsity pattern. To underscore the importance of this property, we briefly outline some alternative first-order methods that do not use the clique decomposition. The first problem (a projection on the sparse PSD cone) can be viewed as projection on the intersection $\mathbb{S}^p_E \cap \mathbb{S}^+_{E}$ of two sets. One can apply Dykstra’s algorithm and solve the problem by alternating projection on the set of sparse matrices $\mathbb{S}^p_E$ and the dense PSD cone $\mathbb{S}^+_{E}$. Similarly, the projections on the PSD completable and EDM completable cones (problems II and III) can be viewed as minimizing a sum $f(X) + g(X)$, where

$$f(X) = \sum_{\{i,j\} \in E} (X_{ij} - A_{ij})^2, \quad g(X) = \delta_C(X)$$
where \( C = S_p^+ \) or \( D^p \). The Douglas-Rachford method for minimizing the sum requires projections on \( C \). Hence, at every iteration of these algorithms, a single eigenvalue decomposition of order \( p \) is needed. However, a full eigenvalue decomposition quickly becomes impractical for \( p \) greater than 10,000. On the machine used for our experiments, a single dense eigenvalue decomposition of order \( p = 20,000 \) takes 15 minutes to compute, and exceeds memory for \( p \geq 50,000 \). Sparse eigenvalue decomposition methods also pose difficulties. Even when the initial and final matrix variables in the algorithms are sparse, the intermediate variables (and in particular, the arguments to the projections on \( S_p^+ \) and \( D^p \)) are dense and, unless the method has almost converged, have close to \( p/2 \) positive and negative eigenvalues. On the same machine, a single full matrix projection using a sparse eigenvalue decomposition took more than an hour for a problem of size \( p = 8204 \) and more than eight hours for a problem of size \( p = 12,992 \). In comparison, the runtimes of the decomposition methods discussed in the paper depend less strongly on \( p \) and more on the sparsity pattern and density; a test problem with order \( p = 21,701 \) converged in 3 minutes, using about 2 seconds per iteration. In none of the test problems were we close to running out of memory.

There are some interesting differences among the decomposition methods. The dual block-coordinate ascent method (Dykstra’s method) and the Douglas-Rachford method seem to converge in fewer iterations than the accelerated dual projection method (despite the stronger theoretical convergence results for the projected gradient method). In addition, the Douglas-Rachford methods are more general and can be applied to the problems with nonchordal sparsity patterns. However they converged more slowly on the test problems with non-chordal patterns. We did not observe a difference in efficiency between the primal and dual Douglas-Rachford methods. A general difficulty when applying the Douglas-Rachford algorithm is the sensitivity to the choice of the problem parameters \( t \) and \( \rho \). We used the same fixed values for all the experiments, and it is possible that the performance can be further improved by tuning the parameters.

References


