THE CP-MATRIX APPROXIMATION PROBLEM

JINYAN FAN AND ANWA ZHOU

ABSTRACT. A symmetric matrix $A$ is completely positive (CP) if there exists an entrywise nonnegative matrix $V$ such that $A = VV^T$. In this paper, we study the CP-matrix approximation problem: for a given symmetric matrix $C$, find a CP matrix $X$ such that $X$ is close to $C$ as much as possible, under some linear constraints. We formulate the problem as a linear optimization problem with the norm cone and the cone of moments, then construct a hierarchy of semidefinite relaxations for solving it.

1. Introduction

A real $n \times n$ symmetric matrix $A$ is completely positive (CP) if there exist nonnegative vectors $v_1, \cdots, v_r \in \mathbb{R}_+^n$ such that

$$A = v_1 v_1^T + \cdots + v_r v_r^T. \tag{1.1}$$

The number $r$ is called the length of the decomposition (1.1). The smallest $r$ in the above is called the CP-rank of $A$. If $A$ is CP, we call (1.1) a CP decomposition of $A$. So, $A$ is CP if and only if $A = VV^T$ for an entrywise nonnegative matrix $V$. Clearly, a CP-matrix is doubly nonnegative (DNN), i.e., it is not only positive semidefinite but also nonnegative entrywise.

Let $S_n$ be the space of real $n \times n$ symmetric matrices, equipped with the standard matrix inner product $A \bullet B := \text{trace}(A^T B)$ for $A, B \in S_n$. For a cone $C \subseteq S_n$, recall that its dual cone is the set

$$C^* := \{B \in S_n : A \bullet B \geq 0 \text{ for all } A \in C\}.$$

Denote

$$\text{CP}_n = \{A \in S_n : A = VV^T \text{ with } V \geq 0\}, \text{ the completely positive cone,}$$

$$\text{COP}_n = \{B \in S_n : x^T B x \geq 0 \text{ for all } x \geq 0\}, \text{ the copositive cone.}$$

Both $\text{CP}_n$ and $\text{COP}_n$ are proper cones (i.e. closed, convex, pointed and full-dimensional). Moreover, they are dual to each other [3]. It is well-known that checking the membership in $\text{CP}_n$ is NP-hard, while checking the membership in $\text{COP}_n$ is co-NP-hard [9, 21].

Completely positive matrices have wide applications in combinatorics and statistics. For example, the block designs, maximin efficiency-robust tests and Markovian model of DNA evolution need to check the complete positivity of a given matrix [3]. A lot of problems in


Key words and phrases. completely positive matrices, CP-matrix approximation, linear optimization with moments, semidefinite algorithm.

The first author is partially supported by NSFC 11171217 and 11571234.
applications can be formulated as variations of the standard quadratic optimization problem

\[
\min_{x \in \mathbb{R}^n} \quad x^T B x \\
\text{s.t.} \quad x_1^2 + \cdots + x_n^2 = 1, \\
\quad x \geq 0,
\]

where \( B \in S_n \) is a given matrix. It can be proved that (1.2) is equivalent to the linear conic optimization problem with the CP matrix cone:

\[
\begin{array}{ll}
\min_{X} & B \cdot X \\
\text{s.t.} & I_n \cdot X = 1, \\
& X \in \mathcal{CP}_n,
\end{array}
\]

where \( I_n \) denotes the \( n \times n \) identity matrix (cf. [4]). Interested readers are referred to [1, 7, 10, 11, 27] for more examples. Since it is difficult to treat \( \mathcal{CP}_n \) directly, the linear conic optimization problem (1.3) is still hard to solve. If we approximate \( \mathcal{CP}_n \) by a more computationally tractable cone, say, the cone of doubly nonnegative matrices, then we solve the relaxed optimization and can get an approximate solution \( \hat{X} \). In applications, the \( \hat{X} \) may not be a true optimizer of (1.3). For such cases, people often want to get a CP matrix, whose trace is one, which is close to \( \hat{X} \) as much as possible. Mathematically, this is a CP-approximation problem.

CP-approximation also has applications in statistics. For instance, in some applications, the random vectors are often nonnegative. Their covariance matrices can be expressed by CP matrices. For a nonnegative random vector \( x \) with expectation \( b \) and covariance matrix \( V \), the sum \( A := V + b b^T \) is a CP matrix (cf. [31]). In applications, \( V \) often has noises, which often makes \( A \) not completely positive. Thus, one often wants to approximate \( A \) by a CP matrix, which can be further used to estimate the true covariance matrix \( V \).

Though the relaxation techniques are useful in solving CP optimization problems, solving the relaxation problems can only provide an upper or lower bound on the original problem and the optimal solution is generally not in the CP cone, that is, we can only get a “relaxed” solution. In practice, people would ideally want to know how far the relaxed solution is away from the feasible set of the original problem and what is the “nearest” feasible CP matrix to the relaxed solution. This motivates us to consider the following general CP-matrix approximation problem:

\[
\begin{array}{ll}
\min_{X} & \|X - C\|_p \\
\text{s.t.} & A_i \cdot X = b_i, \quad i = 1, \ldots, m_e, \\
& A_i \cdot X \geq b_i, \quad i = m_e + 1, \ldots, m, \\
& X \in \mathcal{CP}_n,
\end{array}
\]

where \( C, A_i \in S_n, b_i \in \mathbb{R}(i = 1, \ldots, m) \), and \( \| \cdot \|_p \) is the \( p \)-norm (\( p = 1, 2, \infty \) or \( F \)). The problem is to find a matrix in the intersection of a set of linear constraints and the cone of CP matrices such that it is close to a given matrix as much as possible, or approximate a given matrix by a CP matrix under linear constraints.

Specially, if \( C = 0 \), then (1.4) becomes the feasibility problem of finding a CP matrix that satisfies the linear constraints and has the minimum \( p \)-norm.

If there are no linear constraints, (1.4) reduces to the CP projection problem:

\[
\begin{array}{ll}
\min_{X} & \|X - C\|_p \\
\text{s.t.} & X \in \mathcal{CP}_n,
\end{array}
\]
i.e., finding a CP matrix that is closest to $C$, or finding a best CP approximation of $C$. Note that $C$ is CP if and only if the minimum of (1.5) is zero. So, solving (1.5) also provides a way to check whether a matrix is CP. In [29], the polyhedral approximations of $\text{COP}_n$ are used to compute the projection of a matrix onto $\text{COP}_n$, then the projection onto $\text{CP}_n$ can be obtained by a dual approach. In [17], a Newton-like algorithm ($\text{SymNMF}$) is presented to solve the problem:

\begin{equation}
\begin{cases}
\min_{V \in \mathbb{R}^{n \times l}} \|C - VV^T\|_F, \\
\text{s.t. } V \succeq 0,
\end{cases}
\end{equation}

where $C \in \mathcal{S}_n$ and $l \in \mathbb{N}$ are given. Indeed, the problem (1.6) can be regarded as approximating a given matrix by a CP matrix whose CP decomposition length is prescribed in advance. However, the projection matrix of $C$ onto the CP cone may have no CP decompositions with length $l$. So, the CP approximation matrix computed by the $\text{SymNMF}$ may not be the best CP approximation (i.e., the projection matrix). This often occurs in our numerical experiments.

In this paper, we formulate the CP-matrix approximation problem (1.4) as a linear optimization problem with the cone of moments and the $p$-norm cone, then construct a hierarchy of semidefinite relaxations for solving it. If it is infeasible, we can get a certificate for that. If it is feasible, we can get a best CP approximation. Moreover, a CP decomposition of the approximation matrix can also be obtained.

The paper is organized as follows. In section 2, we review the norm cone and its dual cone, and characterize CP matrices as moment sequences. In section 3, we show how to formulate (1.4) as a linear optimization problem with the norm cone and the cone of moments; its dual problem is also given. In section 4, we present a hierarchy of semidefinite relaxations based algorithm for (1.4) and study the convergence properties. We also discuss how to solve the subproblem for the norm cone with different $p$. Some computational results are given in section 5. Finally, we conclude the paper in section 6.

2. Preliminaries

In this section, we first give the dual norm of the $p$-norm on $\mathcal{S}_n$ for $p = 1, 2, \infty$ and $F$ respectively; the $p$-norm cone and its dual cone are also given. Then we characterize CP matrices as truncated moment sequences, and review some basics about moments and localizing matrices, as well as the semidefinite relaxations of the CP cone (cf. Lasserre [18, 19] and Nie [22]).

2.1. $p$-norm cone and its dual. For $A \in \mathbb{R}^{n \times n}$, the $p$-norms of $A$ ($p = 1, 2, \infty, F$) are defined by:

\begin{align*}
\|A\|_1 &= \max_j \sum_{i=1}^n |A_{ij}|, \quad \text{the maximum absolute column sum norm or 1-norm}, \\
\|A\|_2 &= (\lambda_{\max}(A^TA))^{1/2}, \quad \text{the spectral norm or 2-norm}, \\
\|A\|_\infty &= \max_i \sum_{j=1}^n |A_{ij}|, \quad \text{the maximum absolute row sum norm or } \infty\text{-norm}, \\
\|A\|_F &= (\text{trace}(A^TA))^{1/2}, \quad \text{the Frobenius norm or } F\text{-norm}.
\end{align*}

Note that, when $A \in \mathcal{S}_n$, the 1-norm is the same as $\infty$-norm.
Let \( \| \cdot \| \) be a norm on \( S_n \). The associated dual norm, denoted by \( \| \cdot \|_* \), is defined by
\[
\|A\|_* = \sup \{ A \cdot X : \|X\| \leq 1 \},
\]
(2.1)
(cf. [5, Section A.1.6]). It can be proved that the dual norm of the \( p \)-norm for \( p = 1, 2, \infty \) and \( F \) are:
\[
\|A\|_{1*} = \sum_{j=1}^{n} \max_{i} |A_{ij}|, \\
\|A\|_{2*} = \text{trace}((A^T A)^{1/2}), \\
\|A\|_{\infty*} = \max_{i,j} |A_{ij}|, \\
\|A\|_{F*} = \|A\|_F.
\]
(2.2)

For \( \| \cdot \| \) on \( S_n \), the norm cone is defined by
\[
\mathcal{K} = \{(X, s) \in S_n \times \mathbb{R}_+ : \|X\| \leq s\},
\]
where \( S_n \times \mathbb{R}_+ \) is the Cartesian product of \( S_n \) and \( \mathbb{R}_+ \). The dual cone of \( \mathcal{K} \) is defined by
\[
\mathcal{K}^* = \{(Y, t) \in S_n \times \mathbb{R}_+ : X \cdot Y + st \geq 0 \text{ for all } (X, s) \in \mathcal{K}\}.
\]
For the \( p \)-norm cone \( (p = 1, 2, \infty \) and \( F \)),
\[
\mathcal{K}_p = \{(X, s) \in S_n \times \mathbb{R}_+ : \|X\|_p \leq s\},
\]
we can prove that the dual cone of \( \mathcal{K}_p \) is
\[
\mathcal{K}_p^* = \{(Y, t) \in S_n \times \mathbb{R}_+ : \|Y\|_{p*} \leq t\}.
\]

2.2. Characterization as moments. A symmetric matrix \( A \in S_n \) can be identified by a vector consisting of its upper triangular entries, i.e.
\[
\text{vech}(A) = (A_{11}, A_{12}, \ldots, A_{1n}, A_{22}, \ldots, A_{2n}, A_{33}, \ldots, A_{nm})^T.
\]
Let \( \mathbb{N} \) be the set of nonnegative integers. For \( \alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n \), denote \( |\alpha| \) := \( \alpha_1 + \cdots + \alpha_n \). Let
\[
E := \{ \alpha \in \mathbb{N}^n : \alpha = \epsilon_i + \epsilon_j, 1 \leq i \leq j \leq n \},
\]
where \( \epsilon_i \) is the \( i \)-th unit vector in \( \mathbb{R}^n \). Then, \( A \) can also be identified as
\[
a = (a_\alpha)_{\alpha \in E} \in \mathbb{R}^E, \quad a_\alpha = A_{ij} \text{ if } \alpha = \epsilon_i + \epsilon_j \text{ with } i \leq j,
\]
where \( \mathbb{R}^E \) denotes the space of real vectors indexed by \( \alpha \in E \). We call \( a \) an \( E \)-truncated moment sequence (E-tms). Clearly, \( A \mapsto a \) is a linear isomorphism between \( S_n \) and \( \mathbb{R}^E \).

Let
\[
\Delta = \{ x \in \mathbb{R}^n : x_1^2 + \cdots + x_n^2 - 1 = 0, x_1 \geq 0, \cdots, x_n \geq 0 \}
\]
be the nonnegative part of the unit sphere. Every nonnegative vector is a multiple of a vector in \( \Delta \). So, by (1.1), \( A \in CP_n \) if and only if there exist \( \rho_1, \cdots, \rho_r > 0 \) and \( u_1, \cdots, u_r \in \Delta \) such that
\[
A = \rho_1 u_1^T + \cdots + \rho_r u_r^T.
\]
(2.3)
We say the \( E \)-tms \( a \) admits a representing measure supported in \( \Delta \) if there exists a nonnegative Borel measure \( \mu \) supported in \( \Delta \) such that
\[
a_\alpha = \int_{\Delta} x^\alpha d\mu, \quad \forall \alpha \in E,
\]
(2.4)
where $x^a := x_1^{a_1} \cdots x_n^{a_n}$. A measure $\mu$ satisfying the above is called a $\Delta$-representing measure for $a$. A measure is called finitely atomic if its support is a finite set, and is called $r$-atomic if its support consists of at most $r$ distinct points. We refer to [14, 18, 22] for representing measures of truncated moments sequences.

Hence, by (2.4), a symmetric matrix $A$, with the identifying vector $a \in \mathbb{R}^E$, is completely positive if and only if $a$ admits an $r$-atomic $\Delta$-measure (cf. Nie [22]), i.e.,

$$a = \rho_1[u_1]_E + \cdots + \rho_r[u_r]_E,$$

where each $\rho_i > 0$, $u_i \in \Delta$ and

$$[u_i]_E := (u^a_i)_{a \in E}, \; \; i = 1, \ldots, r.$$

Denote

$$\mathcal{R} = \{a \in \mathbb{R}^E : a \text{ admits a } \Delta\text{-measure}\}.$$

Then, $\mathcal{R}$ is the CP cone (cf. [23]). So,

$$\Delta \subset \mathcal{C}_P \text{ if and only if } a \in \mathcal{R}.$$

### 2.3. Localizing matrices and flatness

Let $E$ and $\Delta$ be given in (2.2) and (2.3), respectively. Denote by

$$\mathbb{R}[x]_E := \text{span}\{x^a : a \in E\}$$

the space of real homogeneous polynomials of degree two. We say $\mathbb{R}[x]_E$ is $\Delta$-full if there exists a polynomial $p \in \mathbb{R}[x]_E$ such that $p|_\Delta > 0$ (i.e. $p(u) > 0$ for all $u \in \Delta$). As shown by Nie in [23], the dual cone of $\mathcal{R}$ is

$$\mathcal{P} = \{p \in \mathbb{R}[x]_E : p(x) \geq 0, \forall x \in \Delta\}.$$

For an $E$-tms $a \in \mathbb{R}^E$, define a Riesz functional $F_a$ acting on $\mathbb{R}[x]_E$ as

$$F_a(p) := \sum_{a \in E} p_a a_a, \; \; \text{for all } p = \sum p_a x^a.$$

For convenience, we also denote the inner product $\langle p, a \rangle := F_a(p)$.

Let

$$\mathbb{N}^n_d := \{a \in \mathbb{N}^n : |a| \leq d\} \quad \text{and} \quad \mathbb{R}[x]_d := \text{span}\{x^a : a \in \mathbb{N}^n_d\}.$$

For $s \in \mathbb{R}^d_{\geq 0}$ and $q \in \mathbb{R}[x]_{2k}$, the $k$-th localizing matrix of $q$ generated by $s$ is the symmetric matrix $L_q^{(k)}(s)$ satisfying

$$F_s(qp^2) = \text{vec}(p)^T(L_q^{(k)}(s))\text{vec}(p), \; \; \forall p \in \mathbb{R}[x]_{k-[\deg(q)/2]}.$$

In the above, $\text{vec}(p)$ denotes the coefficient vector of polynomial $p$ in the graded lexicographical ordering, and $|t|$ denotes the smallest integer that is not smaller than $t$. In particular, when $q = 1$, $L_1^{(k)}(s)$ is called a $k$-th order moment matrix and is denoted as $M_k(s)$. Note that we have

$$L_q^{(k)}(s) = \sum_{a} q_a s^{a+\beta+\gamma} \beta,\gamma \in \mathbb{N}^n_{\geq 0} - [\deg(q)/2]$$

and

$$M_k(s) = L_1^{(k)}(s) = (s^{\beta+\gamma})_{\beta,\gamma \in \mathbb{N}^n_{\geq 0}}.$$

We refer to [12, 14, 22] for more details about localizing and moment matrices.

Denote the polynomials:

$$h(x) := x_1^2 + \cdots + x_d^2 - 1, g_0(x) := 1, g_1(x) := x_1, \cdots, g_n(x) := x_n.$$
Note that $\Delta$ given in (2.3) is nonempty and compact. It can also be described equivalently as

$$\Delta = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \geq 0\},$$

where $g(x) = (g_0(x), g_1(x), \ldots, g_n(x))$. If $n = 2$ and $k = 2$, the $k$th localizing matrices of polynomials $h$ and $g$ generated by $s$ can be computed as follows:

$$L_{x_1^2 + x_2^2 - 1}^{(2)}(s) = \begin{pmatrix} s_{(2,0)} + s_{(0,2)} - s_{(0,0)} & s_{(3,0)} + s_{(1,2)} - s_{(1,0)} & s_{(2,1)} + s_{(0,3)} - s_{(0,1)} \\ s_{(3,0)} + s_{(1,2)} - s_{(1,0)} & s_{(4,0)} + s_{(2,2)} - s_{(2,0)} & s_{(3,1)} + s_{(1,3)} - s_{(1,1)} \\ s_{(2,1)} + s_{(0,3)} - s_{(0,1)} & s_{(3,1)} + s_{(1,3)} - s_{(1,1)} & s_{(2,2)} + s_{(0,4)} - s_{(0,2)} \end{pmatrix},$$

$$M_2(s) := L_1^{(2)}(s) = \begin{pmatrix} s_{(0,0)} & s_{(1,0)} & s_{(2,0)} & s_{(3,0)} & s_{(4,0)} \\ s_{(1,0)} & s_{(2,0)} & s_{(3,0)} & s_{(4,0)} & s_{(5,0)} \\ s_{(2,0)} & s_{(3,0)} & s_{(4,0)} & s_{(5,0)} & s_{(6,0)} \\ s_{(1,1)} & s_{(2,1)} & s_{(3,1)} & s_{(4,1)} & s_{(5,1)} \\ s_{(2,1)} & s_{(3,1)} & s_{(4,1)} & s_{(5,1)} & s_{(6,1)} \\ s_{(1,2)} & s_{(2,2)} & s_{(3,2)} & s_{(4,2)} & s_{(5,2)} \\ s_{(2,2)} & s_{(3,2)} & s_{(4,2)} & s_{(5,2)} & s_{(6,2)} \end{pmatrix},$$

$$L_{x_1^2}^{(2)}(s) = \begin{pmatrix} s_{(1,0)} & s_{(2,0)} & s_{(3,0)} & s_{(4,0)} \\ s_{(2,0)} & s_{(3,0)} & s_{(4,0)} & s_{(5,0)} \\ s_{(1,1)} & s_{(2,1)} & s_{(3,1)} & s_{(4,1)} \\ s_{(2,1)} & s_{(3,1)} & s_{(4,1)} & s_{(5,1)} \end{pmatrix}, \quad L_{x_1^2}^{(2)}(s) = \begin{pmatrix} s_{(0,1)} & s_{(1,1)} & s_{(2,1)} & s_{(3,1)} \\ s_{(1,1)} & s_{(2,1)} & s_{(3,1)} & s_{(4,1)} \\ s_{(2,1)} & s_{(3,1)} & s_{(4,1)} & s_{(5,1)} \\ s_{(1,2)} & s_{(2,2)} & s_{(3,2)} & s_{(4,2)} \end{pmatrix}.$$
exists \( R > 0 \) such that \( R - \|x\|^2 \in I(h) + Q(g) \). Clearly, if \( f \in I(h) + Q(g) \), then \( f|_{\Delta} \geq 0 \). Conversely, if \( f|_{\Delta} > 0 \) and \( I(h) + Q(g) \) is archimedean, then \( f \in I(h) + Q(g) \). This is due to Putinar’s Positivstellensatz (cf. [28]).

For each \( k \in \mathbb{N} \), denote

\[
\Psi_k = \{ p \in \mathbb{R}[x]_E : p \in I_{2k}(h) + Q_k(g) \}.
\]

Note that \( E \) is finite, \( \mathbb{R}[x]_E \) is \( \Delta \)-full because \( p = \sum_{i=1}^{n} x_i^2 |_{\Delta} > 0 \), and \( I(h) + Q(g) \) is archimedean because \( 1 - \|x\|^2 = -h(x) \in I(h) + Q(g) \). By Nie [23, Propositions 3.5], we have

\[
\Psi_1 \subseteq \cdots \subseteq \Psi_k \subseteq \Psi_{k+1} \subseteq \cdots \subseteq \mathcal{P}.
\]

Moreover,

\[
\text{int}(\mathcal{P}) \subseteq \bigcup_{k=1}^{\infty} \Psi_k \subseteq \mathcal{P}.
\]

Correspondingly, for each \( k \in \mathbb{N} \), denote

\[
\Gamma_k = \left\{ s \in \mathbb{R}^{\mathbb{N}_E} : L_k^{(k)}(s) = 0, L_k^{(j)}(s) \geq 0, j = 0, 1, \ldots, n \right\},
\]

and

\[
\Upsilon_k = \{ s|_{E} : s \in \Gamma_k \},
\]

(If \( k < \text{deg}(E)/2 \), \( \Upsilon_k \) is defined to be \( \mathbb{R}^E \), by default). Since \( E \) is finite, \( \mathbb{R}[x]_E \) is \( \Delta \)-full and \( I(h) + Q(g) \) is archimedean, by [23, Proposition 3.3], we have

\[
\Upsilon_1 \supseteq \cdots \supseteq \Upsilon_k \supseteq \Upsilon_{k+1} \supseteq \cdots \supseteq \mathcal{R},
\]

and

\[
\bigcap_{k=1}^{\infty} \Upsilon_k = \mathcal{R}.
\]

Moreover, \( \Psi_k \) and \( \Upsilon_k \) are dual to each other (cf. [19, 23]).

As shown above, the hierarchy of \( \Upsilon_k \) provides outer approximations of \( \mathcal{R} \) and converges monotonically and asymptotically to \( \mathcal{R} \). So, \( \Upsilon_k \) can approximate the completely positive cone \( \mathcal{R} \) arbitrarily well.

### 3. Linear optimization problem with CP matrices

In this section, we formulate the CP-matrix approximation problem (1.4) as a linear optimization problem with the cone of moments and the \( p \)-norm cone. The duality is also discussed.

Introducing a variable \( \gamma \in \mathbb{R}_+ \), we transform (1.4) to the problem

\[
\begin{cases}
\min_{X, \gamma} & \gamma \\
\text{s.t.} & \|X - C\|_p \leq \gamma, \\
& A_i \bullet X = b_i, \ i = 1, \ldots, m_e, \\
& A_i \bullet X \geq b_i, \ i = m_e + 1, \ldots, m, \\
& X \in \mathcal{CP}_n.
\end{cases}
\]

(3.1)
Let \( Y = X - C \). Then (3.1) can be equivalently written as

\[
\begin{aligned}
& \min_{X,Y} \gamma \\
\text{s.t.} & \quad A_i \bullet X = b_i, \ i = 1, \ldots, m_e, \\
& \quad A_i \bullet X \geq b_i, \ i = m_e + 1, \ldots, m, \\
& \quad X - Y = C, \\
& \quad X \in CP_n, \\
& \quad (Y,Y) \in \mathcal{K}_p.
\end{aligned}
\]  
(3.2)

The Lagrange function of (3.2) is:

\[
L(X,Y,\gamma,\lambda, P, S, Z, \xi) = \gamma - \sum_{i=1}^{m_e} \lambda_i (A_i \bullet X - b_i) - \sum_{i=m_e+1}^{m} \lambda_i (A_i \bullet X - b_i) \\
- (X - Y - C) \bullet P - X \bullet S - Y \bullet Z - \gamma \xi.
\]  
(3.3)

Denote by \( \mathcal{F}(3.2) \) the feasible set of (3.2). The Lagrange dual problem of (3.2) is

\[
\begin{aligned}
& \max_{\lambda, P, S, Z, \xi} \inf_{(X,Y,\gamma,\lambda, P, S, Z, \xi) \in \mathcal{F}(3.2)} \ L(X,Y,\gamma,\lambda, P, S, Z, \xi) \\
\text{s.t.} & \quad \lambda_i \geq 0, \ i = m_e + 1, \ldots, m, \\
& \quad P \in S_n, \\
& \quad S \in COP_n, \\
& \quad (Z, \xi) \in \mathcal{K}_p^n.
\end{aligned}
\]  
(3.4)

Let \( b = (b_1, \ldots, b_m)^T \). Then (3.4) can be simplified as:

\[
\begin{aligned}
& \max_{\lambda, S, Z} b^T \lambda + C \bullet Z \\
\text{s.t.} & \quad \sum_{i=1}^{m} \lambda_i A_i + S + Z = 0, \\
& \quad \lambda_i \geq 0, \ i = m_e + 1, \ldots, m, \\
& \quad (S, (Z, 1)) \in COP_n \times \mathcal{K}_p^n.
\end{aligned}
\]  
(3.5)

Denote

\[
x = \text{vech}(X) \in \mathbb{R}^\tilde{n}, \\
a_i = \text{vech}(2E_n - I_n) \circ \text{vech}(A_i) \in \mathbb{R}^\tilde{n}, \ i = 1, \ldots, m,
\]
where \( \tilde{n} = n(n + 1)/2 \), \( \circ \) denotes the Hadamard product, and \( E_n \) is the \( n \times n \) all-ones matrix. Then, (3.2) can be formulated as the following linear optimization problem:

\[
(\mathcal{P}) : \begin{aligned}
& \min_{x,Y} \gamma \\
\text{s.t.} & \quad a_i^T x = b_i, \ i = 1, \ldots, m_e, \\
& \quad a_i^T x \geq b_i, \ i = m_e + 1, \ldots, m, \\
& \quad x - \text{vech}(Y) = \text{vech}(C), \\
& \quad (x, (Y, \gamma)) \in \mathcal{R} \times \mathcal{K}_p,
\end{aligned}
\]

where \( \mathcal{R} \) is given by (2.6). The dual problem of \( \mathcal{P} \) is

\[
(\mathcal{D}) : \begin{aligned}
& \max_{\lambda, S, Z} b^T \lambda + C \bullet Z \\
\text{s.t.} & \quad \sum_{i=1}^{m} \lambda_i a_i + s + \text{vech}(2E_n - I_n) \circ \text{vech}(Z) = 0, \\
& \quad \lambda_i \geq 0, \ i = m_e + 1, \ldots, m, \\
& \quad (s, (Z, 1)) \in \mathcal{P} \times \mathcal{K}_p^n,
\end{aligned}
\]

where \( \mathcal{P} \) is given by (2.8).
By weak duality, for all feasible points \((x, Y, \gamma)\) in \((P)\) and \((\lambda, s, Z)\) in \((D)\), we have
\[
\vartheta_P \geq \vartheta_D.
\]
If there exists \((x, Y, \gamma)\) such that \((x, Y, \gamma) \in \mathcal{T}(P)\), \(a_i \cdot x > b_i (i = m_e + 1, \ldots, m)\) and \((x, (Y, \gamma)) \in \text{int}(\mathcal{R} \times \mathcal{K}_p)\), or, if there exists \((\lambda, s, Z) \in \mathcal{T}(D)\) such that \(\lambda_i > 0 (i = m_e + 1, \ldots, m)\) and \((s, (Z, 1)) \in \text{int}(\mathcal{P} \times \mathcal{K}^*_p)\), then the strong duality holds, i.e. the equality holds for (3.6) (cf. [2]).

For convenience, we say \((D)\) has a relative interior, if there exists \((\lambda, s, Z) \in \mathcal{T}(D)\) such that \(\lambda_i > 0 (i = m_e + 1, \ldots, m)\) and \((s, (Z, 1)) \in \text{int}(\mathcal{P} \times \mathcal{K}^*_p)\).

4. A semidefinite algorithm for CP approximation

In this section, we construct a hierarchy of semidefinite relaxations for solving the CP-matrix approximation problem (1.4). An algorithm is presented and its convergence properties are studied. We also discuss how to solve the semidefinite relaxation subproblem for different \(p\)-norm.

4.1. A semidefinite relaxations based algorithm. As shown in (2.18) and (2.22), \(\mathcal{R}\) and \(\mathcal{P}\) have nice relaxations \(\mathcal{Y}_k\) and \(\mathcal{V}_k\), respectively. By (2.20) and (2.21), the hierarchy of semidefinite relaxations for solving \((P)\) is
\[
(P^k) : \begin{cases}
\vartheta^k_P = \min_{x, Y, \gamma} & \gamma \\
\text{s.t.} & a_i^T x = b_i, \ i = 1, \ldots, m_e, \\
 & a_i^T x \geq b_i, \ i = m_e + 1, \ldots, m, \\
 & x - \text{vech}(Y) = \text{vech}(C), \\
 & x = \tilde{x}|_{E}, \\
 & (\tilde{x}, (Y, \gamma)) \in \Gamma_k \times \mathcal{K}_p,
\end{cases}
\]
for the relaxation order \(k = 2, 3, \ldots\) The dual problem of \((P^k)\) is
\[
(D^k) : \begin{cases}
\vartheta^k_D = \max_{\lambda, s, Z} & b^T \lambda + C \bullet Z \\
\text{s.t.} & \sum_{i=1}^m \lambda_i a_i + s + \text{vech}(2E_n - I_n) \circ \text{vech}(Z) = 0, \\
& \lambda_i \geq 0, \ i = m_e + 1, \ldots, m, \\
& (s, (Z, 1)) \in \mathcal{V}_k \times \mathcal{K}^*_p.
\end{cases}
\]
Both \((P^k)\) and \((D^k)\) are SDPs, so they can be solved efficiently.

Clearly, \(\vartheta^k_P \leq \vartheta_P\) and \(\vartheta^k_D \leq \vartheta_D\) for all \(k\). Suppose \((x^{*,\hat{k}}, Y^{*,\hat{k}}, \gamma^{*,\hat{k}}, \tilde{z}^{*,\hat{k}})\) is a minimizer of \((P^k)\) and \((\lambda^{*,\hat{k}}, s^{*,\hat{k}}, Z^{*,\hat{k}})\) is a maximizer of \((D^k)\). If \(x^{*,\hat{k}} = \tilde{z}^{*,\hat{k}}|_{E} \in \mathcal{R}\), then \(\vartheta^k_P = \vartheta_P\) and \((x^{*,\hat{k}}, Y^{*,\hat{k}}, \gamma^{*,\hat{k}})\) is a minimizer of \((P)\), i.e., the relaxation \((P^k)\) is exact for solving \((P)\).

In such case, if \(\vartheta^k_P = \vartheta_P\), then \(\vartheta^k_D = \vartheta_D\) and \((\lambda^{*,\hat{k}}, s^{*,\hat{k}}, Z^{*,\hat{k}})\) is a maximizer of \((D)\). If the relaxation \((P^k)\) is infeasible, then \((P)\) is infeasible, i.e., (1.4) is infeasible.

Based on the analysis above, we present the algorithm as follows.

Algorithm 4.1 (A semidefinite relaxations based algorithm for solving (1.4)).

Step 0. Input \(C \in \mathcal{S}_n\) and \(\Delta\) as (2.11). Let \(k := 2\).

Step 1. Solve the primal-dual pair \((P^k)-(D^k)\). If \((P^k)\) is infeasible, stop and output that \((P)\) is infeasible; otherwise, compute an optimal solution \((x^{*,\hat{k}}, Y^{*,\hat{k}}, \gamma^{*,\hat{k}}, \tilde{z}^{*,\hat{k}})\) of \((P^k)\). Let \(t := 1\).

Step 2. Let \(\hat{x} := \tilde{x}^{*,\hat{k}}|_{E}\). If the rank condition (2.13) is not satisfied, go to Step 4.
Step 3. Compute \( \rho_i > 0 \) and \( u_i \in \Delta \). Output the CP decomposition of the approximation matrix

\[
X^* = \rho_1 u_1 u_1^T + \cdots + \rho_r u_r u_r^T,
\]

where \( r = \text{rank}(M_t(\xi)) \). Stop.

Step 4. If \( t < k \), set \( t := t + 1 \) and go to Step 2; otherwise, set \( k := k + 1 \) and go to Step 1.

Algorithm 4.1 can be implemented by the software GloptiPoly 3 [16], which calls the SDP solver SeDuMi [30]. If (1.4) is infeasible, Algorithm 4.1 can give a certificate for that (i.e., the relaxation problem \((P^k)\) is infeasible for some \( k \)). If (1.4) is feasible, Algorithm 4.1 can give a best CP approximation matrix. The CP decomposition of the approximation matrix can also be obtained. We use the method in Henrion and Lasserre [15] to compute \( \lambda^* \) and \( u_i \).

4.2. Convergence properties. We give the asymptotic convergence of Algorithm 4.1 as follows.

Theorem 4.2. Let \( E \) and \( \Delta \) be as in (2.2) and (2.11), respectively. Suppose \((P)\) is feasible and \((D)\) has a relative interior point. Algorithm 4.1 has the following properties:

(i) For all \( k \) sufficiently large, \((D^k)\) has a relative interior point and \((P^k)\) has a minimizer \((x^*, Y^*, \gamma^*, k^*)\).

(ii) The sequence \( \{(x^*, Y^*, \gamma^*)\} \) is bounded, and each of its accumulation points is a minimizer of \((P)\). The sequence \( \{\gamma^*\} \) converges to the minimum of (1.4).

Proof. (i) Let \((x^0, s^0, Z^0) \in \mathcal{F}(D)\) with \( \lambda^0 > 0(i = m_x + 1, \ldots, m) \), and \((s^0, (Z^0, 1)) \in \text{int}(\mathcal{P} \times \mathcal{K}^*_p)\). Then, \( s^0 |_\Delta > 0 \) (cf. [23, Lemma 3.1]). Note that since \( \Delta \) is compact, there exist \( \epsilon > 0 \) and \( \delta > 0 \) such that

\[
s|_\Delta - \epsilon > \epsilon, \quad \forall s \in B(s^0, \delta).
\]

By [25, Theorem 6], there exists \( N_0 > 0 \) such that

\[
s - \epsilon \in I_{2N_0}(h) + Q_{N_0}(g), \quad \forall s \in B(s^0, \delta).
\]

So \( (D^k) \) has a relative interior point for all \( k \geq N_0 \), thus the strong duality holds for \((P^k)\) and \((D^k)\). As \((P)\) is feasible, the relaxation problem \((P^k)\) is also feasible. So, \((P^k)\) has a minimizer \((x^*, Y^*, \gamma^*, k^*)\) (cf. [2, Theorem 2.4.1]).

(ii) We first show \( \{(x^*, Y^*, \gamma^*)\} \) is bounded. Let \((x^0, s^0, Z^0)\) and \( \epsilon \) be as in the proof of (i). The set \( I_{2N_0}(h) + Q_{N_0}(g) \) is dual to \( \Gamma_{N_0} \). For all \( k \geq N_0 \), we have \( \bar{x}^* \in \Gamma_{N_0} \) and

\[
0 \leq \langle \tilde{y}^0 - \epsilon, \bar{x}^* \rangle = \langle \tilde{y}^0, \bar{x}^* \rangle - \epsilon \langle 1, \bar{x}^* \rangle,
\]

\[
\langle (s^0, Z^0, 1), (x^*, Y^*, \gamma^*) \rangle = \gamma^* - \tilde{y}^0 - C \cdot Z^0.
\]

Since \( \gamma^* \leq \tilde{y} \) and \( \langle s^0, \bar{x}^* \rangle = \langle \tilde{y}^0, \bar{x}^* \rangle \leq \langle (s^0, Z^0, 1), (x^*, Y^*, \gamma^*) \rangle \), it holds that

\[
\langle s^0, \bar{x}^* \rangle \leq T_0 := \tilde{y} - \tilde{y}^0 - (C \cdot Z^0).
\]

We get

\[
0 \leq \langle s^0 - \epsilon, \bar{x}^* \rangle \leq T_0 - \epsilon (\bar{x}^*)_0,
\]

\[
(\bar{x}^*)_0 \leq T_1 := T_0 / \epsilon.
\]

Note that \( I(h) + Q(g) \) is archimedean, following the line of proof given in [23, Theorem 4.3 (ii)], we can obtain that the sequence \( \{x^*\} \) is bounded. Due to the relationships between the definitions of \( x, Y \) and \( \gamma \), we know \( \{(x^*, Y^*, \gamma^*)\} \) is bounded.
Suppose \((x^*, Y^*, \gamma^*)\) is an accumulation point of \(\{(x^{*,k}, Y^{*,k}, \gamma^{*,k})\}\). Without loss of generality, we assume
\[
(x^{*,k}, Y^{*,k}, \gamma^{*,k}) \to (x^*, Y^*, \gamma^*), \quad k \to \infty.
\]
Since \(x^{*,k} \in \mathcal{T}_k\), by (2.22) and (2.23), we have \(x^* \in \bigcap_{k=1}^{\infty} \mathcal{T}_k = \mathcal{R}\). Note that \((x^{*,k}, Y^{*,k}, \gamma^{*,k}) \in \mathcal{F}(P^k)\), we further obtain \((x^*, Y^*, \gamma^*) \in \mathcal{F}(P)\). Hence,
\[
(4.1) \quad \vartheta_P \leq \gamma^*.
\]
Since \((P^k)\) is a relaxation problem of \((P)\) and \((x^{*,k}, Y^{*,k}, \gamma^{*,k})\) is a minimizer of \((P^k)\), we have
\[
\vartheta_P \geq \gamma^{*,k}, \quad k = 1, 2, \ldots
\]
Taking \(k \to \infty\), we get
\[
(4.2) \quad \vartheta_P \geq \lim_{k \to \infty} \gamma^{*,k} = \gamma^*.
\]
which together with (4.1) implies that
\[
\vartheta_P = \gamma^*.
\]
So, \((x^*, Y^*, \gamma^*)\) is a minimizer of \((P)\), and the sequence \(\{\gamma^{*,k}\}\) converges to the minimum of \((P)\). □

Remark 4.3. If (1.4) is feasible, then, under some general conditions (cf. [24, 26]), we can get a flat extension \(\tilde{x}^{*,k}\) by solving the hierarchy of \((P^k)\), within finitely many steps (cf. [23, Section 4]).

4.3. Subproblem solving. We discuss how to solve the subproblem \((P^k)\) in Algorithm 4.1 for different \(p\)-norm (i.e., \(p = 1, 2, \infty, F\)).

1. 1-norm or \(\infty\)-norm cone. The 1-norm and \(\infty\)-norm are the same for symmetric matrices. Let \(Y = Y^+ - Y^-\), where \(Y^+, Y^- \in \mathcal{S}_n\) and \(Y^+, Y^- \geq 0\). Then \((P^k)\) can be transformed to the following problem:
\[
\begin{aligned}
\min_{x, y^+, y^-, \gamma} & \quad \gamma \\
\text{s.t.} & \quad a_i^T x = b_i, \quad i = 1, \ldots, m_e, \\
& \quad a_i^T x \geq b_i, \quad i = m_e + 1, \ldots, m, \\
& \quad x - \text{vech}(Y^+ - Y^-) = \text{vech}(C), \\
& \quad \tilde{E}_j \bullet (Y^+ + Y^-) \leq \gamma, \quad j = 1, \ldots, n, \\
& \quad Y^+, Y^- \geq 0, Y^+, Y^- \in \mathcal{S}_n, \\
& \quad x = \tilde{x}|\mathcal{E}_i, \tilde{x} \in \Gamma_k,
\end{aligned}
\]
where \(\tilde{E}_j\) is the matrix whose \(j\)-th column is of all ones and other entries are zeros. (4.3) is a linear optimization problem with linear matrix inequalities.

2. 2-norm cone. Note that \((Y, \gamma) \in \mathcal{K}_2\) if and only if \(\begin{pmatrix} \gamma I_n & Y \\ Y^T & \gamma I_n \end{pmatrix} \succeq 0\). Since \(Y = X - C\), we can transform \((P^k)\) to the problem
\[
\begin{aligned}
\min_{x, \gamma, \tilde{x}} & \quad \gamma \\
\text{s.t.} & \quad a_i^T x = b_i, \quad i = 1, \ldots, m_e, \\
& \quad a_i^T x \geq b_i, \quad i = m_e + 1, \ldots, m, \\
& \quad \gamma I_n = \text{vech}^{-1}(x) - C \\
& \quad \gamma I_n^T (\text{vech}^{-1}(x) - C) \succeq 0, \\
& \quad x = \tilde{x}|\mathcal{E}_i, \tilde{x} \in \Gamma_k,
\end{aligned}
\]
where $\text{vech}^{-1}(\cdot)$ denotes the inverse of the linear operator $\text{vech}(\cdot)$.

3. $F$-norm cone. Let

$$y = \text{vech}(\sqrt{2}E_n + (1 - \sqrt{2})I_n) \circ \text{vech}(Y).$$

Then $(Y, \gamma) \in \mathcal{K}_F$ if and only if $(y, \gamma) \in \mathcal{L}_{n+1}$, where

$$\mathcal{L}_{n+1} = \{(y, \gamma) \in \mathbb{R}^{n+1} : \|y\|_2 \leq \gamma\}$$

is the second-order cone (or Lorentz cone). Since $Y = X - C$, $(P^k)$ can be transformed to the following problem:

$$\min_{x, y, \gamma} \gamma$$

s.t. $a_i^T x = b_i$, $i = 1, \ldots, m_e$,

$a_i^T x \geq b_i$, $i = m_e + 1, \ldots, m$,

$y = \text{vech}(\sqrt{2}E_n + (1 - \sqrt{2})I_n) \circ (x - \text{vech}(C))$,

$x = \tilde{x}|_E$,

$(\tilde{x}, (y, \gamma)) \in \Gamma_k \times \mathcal{L}_{n+1}$.

(4.5) is a linear optimization problem with linear matrix inequalities and the second-order cone.

The problems (4.3), (4.4) and (4.5) can be solved by implementing the software SeDuMi [30].

5. Numerical experiments

In this section, we first present some numerical results for the CP projection problem (1.5), which has no linear constraints, then for the general CP-matrix approximation problem (1.4). The experiments are implemented on a laptop with an Intel Core i5-2520M CPU and 4GB of RAM, using Matlab R2012b. We only display 4 digits for each number.

5.1. CP projection problem in $F$-norm. We use Algorithm 4.1 to compute the CP projection of a given matrix, and compare it with the SymNMF in [17], which computes CP approximations of a given matrix.

Example 5.1. Consider the symmetric matrix given in [29]:

$$C = \begin{pmatrix} 2 & 1 & 1 & 1 & 2 \\ 1 & 2 & 2 & 1 & 1 \\ 1 & 2 & 6 & 5 & 1 \\ 1 & 1 & 5 & 6 & 2 \\ 2 & 1 & 1 & 2 & 3 \end{pmatrix}.$$

Algorithm 4.1 terminates at $k = 3$, with $\gamma^{*, k} = 0.0000$. The projection matrix of $C$ onto $CP_3$ is

$$X^* = \begin{pmatrix} 2.0000 & 1.0000 & 1.0000 & 1.0000 & 2.0000 \\ 1.0000 & 2.0000 & 2.0000 & 1.0000 & 1.0000 \\ 1.0000 & 2.0000 & 6.0000 & 5.0000 & 1.0000 \\ 1.0000 & 1.0000 & 5.0000 & 6.0000 & 2.0000 \\ 2.0000 & 1.0000 & 1.0000 & 2.0000 & 3.0000 \end{pmatrix}.$$
The CP decomposition of $X$ is $X^* = \sum_{i=1}^{6} \rho_i u_i u_i^T$, where the points and their weights are:

- $\rho_1 = 4.7348$, $u_1 = (0.0206, 0.5542, 0.7575, 0.3444, 0.0000)^T$,
- $\rho_2 = 0.7571$, $u_2 = (0.5584, 0.0000, 0.8295, 0.0000, 0.0000)^T$,
- $\rho_3 = 3.8267$, $u_3 = (0.6383, 0.3630, 0.0000, 0.0654, 0.6757)^T$,
- $\rho_4 = 0.2218$, $u_4 = (0.6196, 0.4327, 0.1272, 0.0583, 0.6397)^T$,
- $\rho_5 = 4.9097$, $u_5 = (0.1096, 0.0000, 0.4111, 0.7631, 0.4865)^T$,
- $\rho_6 = 4.5499$, $u_6 = (0.1137, 0.0000, 0.6511, 0.7505, 0.0000)^T$.

Since the minimum is zero, the CP projection of $C$ is itself. This implies that $C$ is CP. A CP decomposition of $C$ is also obtained. The length of the decomposition above is much smaller than that given in [29]. This shows an advantage of Algorithm 4.1.

Next, we use the SymNMF to compute CP approximation matrices of $C$. Let $l = 3$ in (1.6). We obtain the minimum of (1.6) $\eta^* = 0.5143$, as well as the minimizer

$$V^* = \begin{pmatrix} 1.2608 & 0.1543 & 0.3675 \\ 0.5799 & 0 & 1.2243 \\ 0.0066 & 1.7317 & 1.7127 \\ 0.3825 & 2.3408 & 0.5801 \\ 1.6093 & 0.5867 & 0 \end{pmatrix}.$$ 

The CP approximation of $C$ can be computed by $W^* = V^* V^*^T$. Since $\eta^* > 0$, $W^*$ is not the CP projection of $C$.

Let $l = 6$ in (1.6). The SymNMF gets $\eta^* = 0.0050$ and

$$V^* = \begin{pmatrix} 0 & 0.7192 & 1.1228 & 0.0491 & 0.4684 & 0.0000 \\ 0 & 1.0341 & 0 & 0 & 0.5475 & 0.7941 \\ 0.8307 & 0.0350 & 0.2283 & 1.0029 & 1.4290 & 1.4865 \\ 1.5484 & 0 & 0.5956 & 1.4714 & 0.5530 & 0.8816 \\ 0.8445 & 0.9673 & 1.1621 & 0 & 0 & 0 \end{pmatrix},$$

which implies that $W^* = V^* V^*^T$ is not the best CP approximation of $C$. We see that, even if $l$ in (1.6) is given the length of the CP decomposition of $C$, the SymNMF may not find the CP projection because it may get stuck in a local minimum.

**Example 5.2.** Consider the CP matrix given as:

$$C = \begin{pmatrix} 4 & 5 & 4 & 6 & 4 & 2 \\ 5 & 1 & 4 & 7 & 4 & 6 \\ 4 & 4 & 4 & 2 & 5 & 4 \\ 6 & 7 & 2 & 0 & 3 & 7 \\ 4 & 4 & 5 & 3 & 1 & 6 \\ 2 & 6 & 4 & 7 & 6 & 4 \end{pmatrix},$$

which is generated randomly in Matlab.
Algorithm 4.1 terminates at $k = 3$, with $\gamma^{*,k} = 9.7852$. So, $C$ is not CP. The projection matrix of $C$ onto $\text{CP}_3$ is

$$X^* = \begin{pmatrix}
5.3185 & 4.4216 & 3.6259 & 4.2906 & 3.4446 & 3.6738 \\
4.4216 & 4.8770 & 3.6321 & 4.7517 & 3.9731 & 5.1853 \\
3.6259 & 3.6321 & 4.7972 & 3.0761 & 3.9515 & 3.9240 \\
3.4446 & 3.9731 & 3.9515 & 3.6374 & 3.7867 & 4.5141 \\
\end{pmatrix}.$$  

The CP decomposition of $X^*$ is $X^* = \sum_{i=1}^{3} \rho_i u_i u_i^T$, where the points and their weights are:

- $\rho_1 = 13.8129$, $u_1 = (0.4826, 0.3631, 0.5439, 0.2982, 0.3980, 0.3035)^T$,
- $\rho_2 = 8.1253$, $u_2 = (0.5085, 0.4843, 0.0000, 0.5573, 0.1916, 0.3995)^T$,
- $\rho_3 = 7.9465$, $u_3 = (0.0000, 0.3805, 0.2991, 0.3516, 0.4046, 0.6917)^T$.

Let $l = 3$ in (1.6). The SymNMF obtains the minimum $\eta^* = 9.7852$ and the minimizer

$$V^* = \begin{pmatrix}
2.1814 & 0.6135 & 0.4280 \\
1.5222 & 1.5661 & 0.3269 \\
1.0741 & 0.9270 & 1.6682 \\
1.5324 & 1.5443 & 0 \\
1.0083 & 1.3552 & 0.9659 \\
0.9474 & 2.2921 & 0.4683
\end{pmatrix}.$$  

In this example, the SymNMF finds the CP projection of $C$.

Let $l = 6$ in (1.6). The SymNMF obtains the same minimum $\eta^* = 9.7852$, and the minimizer

$$V^* = \begin{pmatrix}
0.2375 & 1.1593 & 0.6315 & 1.5859 & 0.7213 & 0.6958 \\
1.0975 & 0.7625 & 0.9082 & 1.0511 & 0.5339 & 0.9360 \\
0.4323 & 0.8063 & 1.2452 & 0.3350 & 1.4891 & 0.2826 \\
1.1233 & 0.7096 & 0.7465 & 1.1546 & 0.2817 & 0.9997 \\
0.8862 & 0.6259 & 1.0807 & 0.4675 & 0.9370 & 0.5876 \\
1.7300 & 0.4629 & 1.2157 & 0.5232 & 0.5478 & 1.0545
\end{pmatrix}.$$  

It gives another CP decomposition of the projection matrix.

**Example 5.3.** Consider the DNN matrix given in [6]:

$$C = \begin{pmatrix}
1 & 1 & 0 & 0 & 1 \\
1 & 3 & 4 & 0 & 0 \\
0 & 4 & 16 & 4 & 0 \\
0 & 0 & 4 & 2 & 1 \\
1 & 0 & 0 & 1 & 2
\end{pmatrix}.$$  

Algorithm 4.1 terminates at $k = 5$, with $\gamma^{*,k} = 0.5305$. So, $C$ is not CP. The projection matrix of $C$ onto $\text{CP}_3$ is

$$X^* = \begin{pmatrix}
1.1620 & 0.8778 & 0.0674 & 0.1562 & 0.8742 \\
0.8778 & 3.0922 & 3.9492 & 0.1178 & 0.0949 \\
0.0674 & 3.9492 & 16.0280 & 3.9350 & 0.0523 \\
0.1562 & 0.1178 & 3.9350 & 2.1506 & 0.8788 \\
0.8742 & 0.0949 & 0.0523 & 0.8788 & 2.0976
\end{pmatrix}.$$
The CP decomposition of $X^*$ is $X^* = \sum_{i=1}^{8} \rho_i u_i u_i^T$, where the points and their weights are:

- $\rho_1 = 10.9370$, $u_1 = (0.0000, 0.0286, 0.9258, 0.3770, 0.0000)^T$,
- $\rho_2 = 0.8181$, $u_2 = (0.6776, 0.1615, 0.0000, 0.0000, 0.7176)^T$,
- $\rho_3 = 1.3630$, $u_3 = (0.4013, 0.0000, 0.0000, 0.2852, 0.8701)^T$,
- $\rho_4 = 8.2213$, $u_4 = (0.0000, 0.4826, 0.8754, 0.0000, 0.0000)^T$,
- $\rho_5 = 1.4922$, $u_5 = (0.6024, 0.7985, 0.0000, 0.0000, 0.0000)^T$,
- $\rho_6 = 0.2685$, $u_6 = (0.0000, 0.0000, 0.8131, 0.5353, 0.2302)^T$,
- $\rho_7 = 1.0388$, $u_7 = (0.0000, 0.0000, 0.0000, 0.6268, 0.7791)^T$,
- $\rho_8 = 0.3982$, $u_8 = (0.2514, 0.7009, 0.6666, 0.0000, 0.0000)^T$.

We also use the SymNMF to compute CP approximations of $C$. Let $l = 3$ in (1.6). The SymNMF obtains the minimum $\eta^* = 1.3140$, and the minimizer

$$V^* = \begin{pmatrix} 0.1914 & 0 & 0.7753 \\ 0 & 0.1733 & 1.7460 \\ 0 & 0 & 2.2819 \\ 0 & 0 & 0 & 3.2817 \\ 0 & 0 & 0 & 0 & 0.5571 \\ 0 & 0 & 0 & 0 & 0.0215 \\ 0 & 0 & 0 & 0 & 1.4112 \end{pmatrix},$$

which gives the CP approximation $W^* = V^* V^*^T$ of $C$. Since $\eta^*$ is larger than the minimum $\gamma^{*,k}$ given by Algorithm 4.1, $W^*$ is not the best CP approximation (i.e., the CP projection).

Let $l = 8$ in (1.6). The SymNMF obtains the minimum $\eta^* = 0.5921$ and the minimizer

$$V^* = \begin{pmatrix} 0 & 0 & 0 & 0.5041 & 0 & 0.0438 & 0.9195 & 0 \\ 0.9062 & 0.9703 & 0.0001 & 0 & 0 & 0.5777 & 0.9781 & 0 \\ 1.8648 & 1.7889 & 2.2363 & 0 & 1.1509 & 0.9437 & 0 & 1.4567 \\ 0.0732 & 0 & 0.9733 & 0.5766 & 0.6009 & 0 & 0 & 0.6589 \\ 0 & 0 & 0 & 1.4443 & 0.0561 & 0 & 0.1310 & 0 \end{pmatrix}.$$  

Since $\eta^* > \gamma^{*,k}$, the SymNMF fails to find the CP projection of $C$ as it gets stuck in a local minimum, though $l$ is given the length of the CP decomposition of the projection matrix.

**Example 5.4.** Consider the DNN matrix given in [13]:

$$C = \begin{pmatrix} 4 & 0 & 0 & 2 & 2 \\ 0 & 4 & 3 & 0 & 2 \\ 0 & 3 & 4 & 2 & 0 \\ 2 & 0 & 2 & 4 & 0 \\ 2 & 2 & 0 & 0 & 4 \end{pmatrix}.$$  

Algorithm 4.1 terminates at $k = 5$, with $\gamma^{*,k} = 0.4826$. So, $C$ is not CP. The projection matrix of $C$ onto $CP_5$ is

$$X^* = \begin{pmatrix} 4.0597 & 0.0903 & 0.0903 & 1.9331 & 1.9331 \\ 0.0903 & 4.1365 & 2.8635 & 0.1012 & 1.8988 \\ 0.0903 & 2.8635 & 4.1365 & 1.8988 & 0.1012 \\ 1.9331 & 0.1012 & 1.8988 & 4.0750 & 0.0750 \\ 1.9331 & 1.8988 & 0.1012 & 0.0750 & 4.0750 \end{pmatrix}.$$
The CP decomposition of $X^*$ is $X^* = \sum_{i=1}^{7} \rho_i u_i u_i^T$, where the points and their weights are:

- $\rho_1 = 3.6595, \ u_1 = (0.7611, 0.0000, 0.0000, 0.0316, 0.6479)^T$,
- $\rho_2 = 3.5490, \ u_2 = (0.0442, 0.5761, 0.0000, 0.0000, 0.8161)^T$,
- $\rho_3 = 1.0753, \ u_3 = (0.7459, 0.0000, 0.0000, 0.6660, 0.0000)^T$,
- $\rho_4 = 2.6323, \ u_4 = (0.7121, 0.0000, 0.0482, 0.7001, 0.0000)^T$,
- $\rho_5 = 0.5363, \ u_5 = (0.0000, 0.7517, 0.3291, 0.0000, 0.5714)^T$,
- $\rho_6 = 5.4659, \ u_6 = (0.0000, 0.6970, 0.7167, 0.0265, 0.0000)^T$,
- $\rho_7 = 3.5682, \ u_7 = (0.0000, 0.0000, 0.5955, 0.8029, 0.0000)^T$.

Let $l = 3$ in (1.6). The SymNMF gets the minimum $\eta^* = 2.3636$ and the minimizer

$$V^* = \begin{pmatrix}
0 & 1.2204 & 1.2210 \\
1.8174 & 0 & 0.6277 \\
1.8170 & 0.6285 & 0 \\
0.2141 & 1.9579 & 0 \\
0.2147 & 0 & 1.9577
\end{pmatrix}.$$

So, $W^* = V^* V^T$ is not the CP projection of $C$.

Let $l = 7$. The SymNMF gets the minimum $\eta^* = 0.4826$ and the minimizer

$$V^* = \begin{pmatrix}
0.1015 & 1.3842 & 0 & 0.0802 & 0 & 0 & 1.4584 \\
0.8601 & 0.0023 & 1.1805 & 0 & 0.8338 & 1.1436 & 0 \\
0 & 0 & 1.1852 & 1.1259 & 0.1030 & 1.2055 & 0 \\
0 & 0 & 0.0055 & 1.5924 & 0 & 0.0827 & 1.2379 \\
1.2505 & 1.2407 & 0 & 0 & 0.9840 & 0 & 0.0608
\end{pmatrix}.$$

So, the SymNMF finds the CP projection of $C$.

**Example 5.5.** Consider the elapsed time of computing the CP projection of a random symmetric matrix. For each $n = 2, 3, \ldots, 10$, we generate 50 random symmetric $n \times n$ matrices by the Matlab codes:

$$B = \text{rand}(n, n); \quad C = (B + B^T)/2.$$

Table 1 shows the average time (seconds) consumed by Algorithm 4.1 to compute the CP projection matrix.

<table>
<thead>
<tr>
<th>n</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>0.36</td>
<td>0.53</td>
<td>0.80</td>
<td>1.56</td>
<td>4.76</td>
<td>21.37</td>
<td>101.29</td>
<td>428.21</td>
<td>1732.21</td>
</tr>
</tbody>
</table>

**Table 1.** The average time for computing the CP projection matrix.

**Remark 5.6.** Suppose the length of the CP decomposition of the projection matrix is known in advance. A natural idea to compute the CP projection of a given matrix is to transform (1.6) to the unconstrained problem:

$$\min_{X \in \mathbb{R}^{n \times l}} \|C - (X^2)(X^2)^T\|_F^2,$$

where $C \in \mathbb{S}_n$ and $l \in \mathbb{N}$ are given, and the variable $X^2$ is component-wise product. (5.1) is a polynomial optimization problem. Theoretically, it can be solved for any given $C$ and $l$ by implementing the software GloptiPoly. However, even if $n$ and $l$ are small, the scale
of the semidefinite relaxation problems could be very large, which will be too expensive to solve. So, in general, we do not implement the GloptiPoly to solve (5.1) in practice.

5.2. CP-approximation in 1-norm or $\infty$-norm.

Example 5.7. Consider the symmetric matrix $C$ given as:

\[
C = \begin{pmatrix}
6 & 1 & 1 & 3 \\
1 & 2 & 1 & 0 \\
1 & 1 & 2 & 1 \\
3 & 0 & 1 & 2
\end{pmatrix}.
\]

It was shown in [3] that $C \in CP_{4}$ and its CP-rank is 4.

Case 1. Consider (1.4) with the CP cone and the linear constraints $A_{i} \bullet X = b_{i} (i = 1, 2)$, where

\[
A_{1} = I_{4}, \quad A_{2} = \begin{pmatrix}
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0
\end{pmatrix},
\]

\[
b_{1} = 12, \quad b_{2} = 12.
\]

Algorithm 4.1 terminates at $k = 3$, with $\gamma^{u,k} = 0.0000$ and $x^{u,k} \in R$. So, $X^{*} = C$. This implies that $C$ is not only CP but also satisfies the linear constraints. The CP decomposition of $C$ is $C = \sum_{i=1}^{4} \rho_{i} u_{i} u_{i}^{T}$, where the points and their weights are:

\[
\rho_{1} = 1.2672, \quad u_{1} = (0.7070, 0.7072, 0.0000, 0.0000)^{T},
\]

\[
\rho_{2} = 2.1964, \quad u_{2} = (0.2115, 0.7887, 0.5773, 0.0000)^{T},
\]

\[
\rho_{3} = 6.0574, \quad u_{3} = (0.8944, 0.0000, 0.0000, 0.4472)^{T},
\]

\[
\rho_{4} = 2.4790, \quad u_{4} = (0.4128, 0.0000, 0.7152, 0.5640)^{T}.
\]

We obtained a minimal CP decomposition of $C$. It is different from the minimal CP decomposition given in [3].

Case 2. Consider (1.4) with the CP cone and the linear constraints $A_{i} \bullet X = b_{i} (i = 1, 2)$, where $A_{1}, A_{2}, b_{2}$ are the same as in Case 1, but

\[
b_{1} = 10.
\]

Algorithm 4.1 terminates at $k = 3$ with $\gamma^{u,k} = 0.9056$ and $x^{u,k} \in R$. The optimal solution is

\[
X^{*} = \begin{pmatrix}
5.0944 & 1.0000 & 1.0000 & 3.0000 \\
1.0000 & 1.7185 & 1.0000 & 0.6241 \\
1.0000 & 1.0000 & 1.0944 & 1.0000 \\
3.0000 & 0.6241 & 1.0000 & 2.0927
\end{pmatrix}.
\]

The CP decomposition of $X^{*}$ is $X^{*} = \sum_{i=1}^{3} \rho_{i} u_{i} u_{i}^{T}$, where the points and their weights are:

\[
\rho_{1} = 1.5043, \quad u_{1} = (0.7354, 0.6709, 0.0000, 0.0954)^{T},
\]

\[
\rho_{2} = 6.4871, \quad u_{2} = (0.8123, 0.0489, 0.1898, 0.5493)^{T},
\]

\[
\rho_{3} = 2.0086, \quad u_{3} = (0.0000, 0.7147, 0.6546, 0.2463)^{T}.
\]

Case 3. Consider (1.4) with the CP cone and the linear constraints $A_{i} \bullet X = b_{i} (i = 1, 2)$, where $A_{1}, A_{2}, b_{1}$ are the same as in Case 2, but

\[
b_{2} = -5.
\]
Algorithm 4.1 terminates at \( k = 2 \) as \( (P^k) \) is infeasible. So, (1.4) is infeasible.

**Case 4.** Consider (1.4) with the CP cone and the linear constraints \( A_1 \cdot X = b_1 \) and \( A_2 \cdot X \geq b_2 \), where \( A_i, b_i (i = 1, 2) \) are the same as in Case 3.

Algorithm 4.1 terminates at \( k = 3 \) with \( \gamma^{k,k} = 0.8224 \). The optimal solution is

\[
X^* = \begin{pmatrix}
5.2994 & 1.0000 & 1.0000 & 2.8783 \\
1.0000 & 1.4671 & 1.0000 & 0.2895 \\
1.0000 & 1.0000 & 1.4111 & 0.7665 \\
2.8783 & 0.2895 & 0.7665 & 1.8224 \\
\end{pmatrix}.
\]

The CP decomposition of \( X^* \) is \( X^* = \sum_{i=1}^{3} \rho_i u_i u_i^T \), where the points and their weights are:

\[
\rho_1 = 5.8000, \quad u_1 = (0.8788, 0.0000, 0.0000, 0.4773)^T, \\
\rho_2 = 2.2370, \quad u_2 = (0.4209, 0.2734, 0.7239, 0.4733)^T, \\
\rho_3 = 1.9630, \quad u_3 = (0.4648, 0.8138, 0.3489, 0.0000)^T.
\]

### 5.3. CP-approximation in 2-norm.

**Example 5.8.** Consider the symmetric matrix \( C \) given as (cf. [3, page 188]):

\[
(5.3) \quad C = \begin{pmatrix}
6 & 1 & 0 & 1 & 3 \\
1 & 2 & 1 & 0 & 0 \\
0 & 1 & 2 & 1 & 0 \\
1 & 0 & 1 & 2 & 1 \\
3 & 0 & 0 & 1 & 2 \\
\end{pmatrix}.
\]

**Case 1.** Consider (1.4) with the CP cone and the linear constraints \( A_i \cdot X = b_i(i = 1, 2, 3) \), where

\[
A_1 = I_5, \quad A_2 = \begin{pmatrix}
1 & -1 & 1 & -1 & 1 \\
-1 & 2 & -2 & 2 & -2 \\
1 & -2 & 3 & -3 & 3 \\
-1 & 2 & -3 & 4 & -4 \\
1 & -2 & 3 & -4 & 5 \\
\end{pmatrix}, \quad A_3 = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
\end{pmatrix},
\]

\( b_1 = 14, \quad b_2 = 18, \quad b_3 = 10. \)

Algorithm 4.1 terminates at \( k = 3 \), with \( \gamma^{k,k} = 0.0000 \) and \( x^{k,k} \in \mathcal{R} \). So, \( X^* = C \). This implies that \( C \) is not only CP but also satisfies the linear constraints. The CP decomposition of \( C \) is \( C = \sum_{i=1}^{5} \rho_i u_i u_i^T \), where the points and their weights are:

\[
\rho_1 = 0.9997, \quad u_1 = (0.0000, 0.0000, 0.0000, 0.8944, 0.4474)^T, \\
\rho_2 = 1.9998, \quad u_2 = (0.7070, 0.7073, 0.0000, 0.0000, 0.0000)^T, \\
\rho_3 = 7.0002, \quad u_3 = (0.8452, 0.0000, 0.0000, 0.1690, 0.5071)^T, \\
\rho_4 = 2.0000, \quad u_4 = (0.0000, 0.0000, 0.7070, 0.7072, 0.0000)^T, \\
\rho_5 = 2.0000, \quad u_5 = (0.0000, 0.7070, 0.7073, 0.0000, 0.0000)^T.
\]

We obtained a minimal CP decomposition of \( C \). It is different from the minimal CP decomposition given in [3].

**Case 2.** Consider (1.4) with the CP cone and the linear constraints \( A_i \cdot X = b_i(i = 1, 2, 3) \), where \( A_1, A_2, A_3, b_1, b_3 \) are the same as in Case 1, and

\( b_2 = 20. \)
Algorithm 4.1 terminates at \( k = 3 \) with \( \gamma^{*,k} = 0.1665 \). The optimal solution is

\[
X^* = \begin{pmatrix}
5.9177 & 1.0038 & 0.0829 & 1.0484 & 3.0416 \\
1.0038 & 2.0281 & 0.9543 & 0.1262 & 0.0777 \\
0.0829 & 0.9543 & 1.9528 & 0.9706 & 0.0914 \\
1.0484 & 0.1262 & 0.9706 & 2.0235 & 0.9452 \\
3.0416 & 0.0777 & 0.0914 & 0.9452 & 2.0778
\end{pmatrix}.
\]

The CP decomposition of \( X^* \) is \( X^* = \sum_{i=1}^{7} \rho_i u_i u_i^T \), where the points and their weights are:

\[
\begin{align*}
\rho_1 &= 5.3308, & u_1 &= (0.8611, 0.0000, 0.0000, 0.0000, 0.5084)^T, \\
\rho_2 &= 2.0094, & u_2 &= (0.0000, 0.8101, 0.5863, 0.0000, 0.0000)^T, \\
\rho_3 &= 2.3891, & u_3 &= (0.4711, 0.0000, 0.0000, 0.7225, 0.5061)^T, \\
\rho_4 &= 1.7647, & u_4 &= (0.0414, 0.0000, 0.7929, 0.6080, 0.0000)^T, \\
\rho_5 &= 0.3057, & u_5 &= (0.1160, 0.0000, 0.7072, 0.5547, 0.4228)^T, \\
\rho_6 &= 0.6683, & u_6 &= (0.8244, 0.5199, 0.0000, 0.0000, 0.2236)^T, \\
\rho_7 &= 1.5320, & u_7 &= (0.7970, 0.5875, 0.0000, 0.1402, 0.0000)^T.
\end{align*}
\]

**Case 3.** Consider (1.4) with the CP cone and the linear constraints \( A_i \cdot X = b_i(i = 1, 2, 3) \), where \( A_1, A_2, A_3, b_1, b_3 \) are the same as in Case 1, but

\[ b_2 = -20. \]

Algorithm 4.1 terminates at \( k = 2 \) as \( (P^k) \) is infeasible. So, (1.4) is infeasible.

**Case 4.** Consider (1.4) with the CP cone and the linear constraints \( A_i \cdot X = b_i(i = 1, 2) \) and \( A_3 \cdot X \geq b_3 \), where \( A_i(i = 1, 2, 3) \) are the same as in Case 1, and

\[ b_1 = 5, \quad b_2 = 10, \quad b_3 = -5. \]

Algorithm 4.1 terminates at \( k = 3 \), with \( \gamma^{*,k} = 3.1791 \) and \( x^{*,k} \in \mathcal{R} \). The optimal solution is:

\[
(5.4) \quad X^* = \begin{pmatrix}
3.4760 & 0.4960 & 0.4772 & 0.7079 & 2.0791 \\
0.4960 & 0.0708 & 0.0681 & 0.1010 & 0.2967 \\
0.4772 & 0.0681 & 0.0655 & 0.0972 & 0.2854 \\
0.7079 & 0.1010 & 0.0972 & 0.1442 & 0.4234 \\
2.0791 & 0.2967 & 0.2854 & 0.4234 & 1.2435
\end{pmatrix}.
\]

The CP decomposition of \( X^* \) is \( X^* = \rho_1 u_1 u_1^T \), where the points and their weights are:

\[ \rho_1 = 5.0000, \quad u_1 = (0.8338, 0.1190, 0.1145, 0.1698, 0.4987)^T. \]

### 5.4. CP-approximation in \( f \)-norm.

**Example 5.9.** Consider the symmetric matrix \( C \) given in Example 5.8.

**Case 1.** Consider (1.4) with the CP cone and the linear constraints \( A_i \cdot X = b_i(i = 1, 2, 3) \), where \( A_i, b_i(i = 1, 2, 3) \) are the same as in Case 1 of Example 5.8.

Algorithm 4.1 terminates at \( k = 3 \), with \( \gamma^{*,k} = 0.0000 \) and \( x^{*,k} \in \mathcal{R} \). So, \( C \in \mathcal{CP}_3 \). We get the same CP decomposition of \( C \) as that in Case 1 of Example 5.8.

**Case 2.** Consider (1.4) with the CP cone and the linear constraints \( A_i \cdot X = b_i(i = 1, 2, 3) \), where \( A_i, b_i(i = 1, 2, 3) \) are the same as in Case 2 of Example 5.8.
Algorithm 4.1 terminates at \( k = 3 \) with \( \gamma^{*,k} = 0.2730 \). The optimal solution is

\[
X^* = \begin{pmatrix}
5.9255 & 1.0435 & 0.0373 & 1.0435 & 3.0373 \\
1.0435 & 1.9627 & 1.0062 & 0.0745 & 0.0062 \\
0.0373 & 1.0062 & 2.0000 & 0.9689 & 1.1118 \\
1.0435 & 0.0745 & 0.9689 & 2.0373 & 0.9317 \\
3.0373 & 0.0062 & 0.1118 & 0.9317 & 2.0745
\end{pmatrix}.
\]

The CP decomposition of \( X^* \) is \( X^* = \sum_{i=1}^{8} \rho_i u_i u_i^T \), where the points and their weights are:

\[
\begin{align*}
\rho_1 &= 0.5521, & u_1 &= (0.0677, 0.0000, 0.9971, 0.0000, 0.0338)^T, \\
\rho_2 &= 1.8849, & u_2 &= (0.0000, 0.7576, 0.6527, 0.0000, 0.0000)^T, \\
\rho_3 &= 0.2252, & u_3 &= (0.0000, 0.5663, 0.5812, 0.5844, 0.0000)^T, \\
\rho_4 &= 1.0358, & u_4 &= (0.0000, 0.0000, 0.4734, 0.8808, 0.0000)^T, \\
\rho_5 &= 2.1552, & u_5 &= (0.7904, 0.6126, 0.0000, 0.0000, 0.0047)^T, \\
\rho_6 &= 0.9895, & u_6 &= (0.0000, 0.0000, 0.5860, 0.7942, 0.1607)^T, \\
\rho_7 &= 3.3621, & u_7 &= (0.8678, 0.0000, 0.0000, 0.0000, 0.4969)^T, \\
\rho_8 &= 3.7953, & u_8 &= (0.7340, 0.0000, 0.0000, 0.3746, 0.5665)^T.
\end{align*}
\]

**Case 3.** Consider (1.4) with the CP cone and the linear constraints \( A_i \cdot X = b_i (i = 1, 2, 3) \), where \( A_i, b_i (i = 1, 2, 3) \) are the same as in Case 3 of Example 5.8.

Algorithm 4.1 terminates at \( k = 2 \) as \( P^k \) is infeasible. So, (1.4) is infeasible.

**Case 4.** Consider (1.4) with the CP cone and the linear constraints \( A_i \cdot X = b_i (i = 1, 2) \) and \( A_3 \cdot X \geq b_3 \), where \( A_i (i = 1, 2, 3) \) are the same as in Case 4 of Example 5.8.

Algorithm 4.1 terminates at \( k = 2 \), with \( \gamma^{*,k} = 5.0042 \) and \( x^{*,k} \in R \). The optimal solution is:

\[
(5.5) \quad X^* = \begin{pmatrix}
3.5222 & 0.4468 & 0.3998 & 0.6882 & 2.0909 \\
0.4468 & 0.0567 & 0.0507 & 0.0873 & 0.2652 \\
0.3998 & 0.0507 & 0.0454 & 0.0781 & 0.2374 \\
0.6882 & 0.0873 & 0.0781 & 0.1345 & 0.4085 \\
2.0909 & 0.2652 & 0.2374 & 0.4085 & 1.2412
\end{pmatrix}.
\]

The CP decomposition of \( X^* \) is \( X^* = \rho_1 u_1 u_1^T \), where the points and their weights are:

\[
\rho_1 = 5.0000, \quad u_1 = (0.8393, 0.1065, 0.0953, 0.1640, 0.4982)^T.
\]

**Example 5.10.** Consider the symmetric matrix \( C \) given in Example 5.2.

**Case 1.** Consider (1.4) without linear constraints. The results are given in Example 5.2.
Case 2. Consider (1.4) with the CP cone and the linear constraints \( A_i \cdot X = b_i (i = 1, 2) \), where \( A_i, b_i (i = 1, 2) \) are generated randomly:

\[
\begin{align*}
A_1 &= \begin{pmatrix}
12 & 0 & 7 & -5 & 4 & -2 \\
0 & 3 & 1 & -2 & -6 & -13 \\
7 & 1 & 4 & 1 & -9 & 6 \\
-5 & -2 & 1 & 7 & -9 & 10 \\
4 & -6 & -9 & -9 & -19 & 1 \\
-2 & -13 & 6 & 10 & 1 & 13
\end{pmatrix}, \\
A_2 &= \begin{pmatrix}
-4 & 3 & 11 & 11 & 2 & -5 \\
3 & 6 & 3 & -3 & 5 & -9 \\
11 & 3 & 5 & 0 & -3 & -9 \\
11 & -3 & 0 & 14 & -4 & -16 \\
2 & 5 & -3 & -4 & 7 & -14 \\
-5 & -9 & -9 & -16 & -14 & 3
\end{pmatrix}, \\
b_1 &= -17, \quad b_2 = 6.
\end{align*}
\]

Algorithm 4.1 terminates at \( k = 3 \), with \( \gamma^{*,k} = 11.4970 \). The optimal solution is

\[
X^* = \begin{pmatrix}
5.5277 & 4.9260 & 5.0372 & 4.5381 & 3.1904 & 2.7187 \\
5.0372 & 4.2671 & 6.0026 & 3.3516 & 3.4975 & 3.0055 \\
4.5381 & 4.8028 & 3.5516 & 5.5691 & 2.9252 & 4.7051 \\
3.1904 & 3.0364 & 3.4975 & 2.9252 & 2.3729 & 3.0340 \\
2.7187 & 3.5882 & 3.0055 & 4.7051 & 3.0340 & 6.9676
\end{pmatrix}.
\]

The CP decomposition of \( X^* \) is \( X^* = \sum_{i=1}^{3} \rho_i u_i u_i^T \), where the points and their weights are:

\[
\begin{align*}
\rho_1 &= 5.0609, \quad u_1 = (0.5069, 0.5243, 0.0000, 0.6726, 0.0996, 0.0766)^T, \\
\rho_2 &= 7.4131, \quad u_2 = (0.0000, 0.1967, 0.0000, 0.4359, 0.1960, 0.8561)^T, \\
\rho_3 &= 18.6772, \quad u_3 = (0.4757, 0.4030, 0.5669, 0.3165, 0.3303, 0.2839)^T.
\end{align*}
\]

Case 3. Consider (1.4) with the CP cone and the linear constraints \( A_i \cdot X = b_i (i = 1, 2) \), where \( A_i, b_i (i = 1, 2) \) are generated randomly:

\[
\begin{align*}
A_1 &= \begin{pmatrix}
8 & -2 & 5 & 6 & 5 & -4 \\
-2 & 10 & 8 & 12 & 17 & 4 \\
5 & 8 & 7 & 6 & -2 & -3 \\
6 & 12 & 6 & 4 & 12 & 7 \\
5 & 17 & -2 & 12 & 10 & -8 \\
-4 & 4 & -3 & 7 & -8 & 9
\end{pmatrix}, \\
A_2 &= \begin{pmatrix}
-2 & -16 & -12 & 4 & 1 & -5 \\
-16 & 3 & 8 & -3 & -10 & 0 \\
-12 & 8 & -13 & -1 & 11 & 3 \\
4 & -3 & -1 & -3 & 5 & 9 \\
1 & -10 & 11 & 5 & 10 & 3 \\
-5 & 0 & 3 & 9 & 3 & -15
\end{pmatrix}, \\
b_1 &= -6, \quad b_2 = 4.
\end{align*}
\]

Algorithm 4.1 terminates at \( k = 2 \) as \( (P^k) \) is infeasible. So, (1.4) is infeasible.
Case 4. Consider (1.4) with the CP cone and the linear constraints \( A_1 \bullet X = b_1, A_2 \bullet X \succeq b_2 \), where \( A_i, b_i (i = 1, 2) \) are generated randomly:

\[
A_1 = \begin{pmatrix}
5 & 7 & -4 & -9 & 4 & 9 \\
7 & -2 & 6 & -4 & 7 & -6 \\
-4 & 6 & -17 & -9 & -1 & 6 \\
-9 & -4 & -9 & 5 & -13 & 6 \\
4 & 7 & -1 & -13 & -3 & 1 \\
9 & -6 & 6 & 6 & 1 & -6
\end{pmatrix},
\]

\[
A_2 = \begin{pmatrix}
2 & -4 & 6 & 4 & 7 & 1 \\
-4 & -2 & 11 & 2 & 6 & 7 \\
6 & 11 & 12 & -9 & -2 & 7 \\
4 & 2 & -9 & -3 & 0 & 10 \\
7 & 6 & -2 & 0 & 4 & -11 \\
1 & 7 & 7 & 10 & -11 & 11
\end{pmatrix},
\]

\[
b_1 = 7, \quad b_2 = -10.
\]

Algorithm 4.1 terminates at \( k = 3 \), with \( \gamma^{*k} = 10.4410 \). The optimal solution is

\[
X^* = \begin{pmatrix}
5.5853 & 4.9676 & 3.4690 & 3.9792 & 3.9314 & 4.6082 \\
4.9676 & 5.0242 & 3.6848 & 4.2784 & 4.1044 & 5.3069 \\
3.4690 & 3.6848 & 3.6883 & 2.3311 & 3.7530 & 3.9202 \\
3.9314 & 4.1044 & 3.7530 & 2.8846 & 3.9128 & 4.3521 \\
\end{pmatrix}.
\]

The CP decomposition of \( X^* \) is

\[
X^* = \sum_{i=1}^{3} \rho_i u_i u_i^T,
\]

where the points and their weights are:

\[
\rho_1 = 3.4136, \quad u_1 = (0.7633, 0.3769, 0.3647, 0.0000, 0.3772, 0.0000)^T,
\]

\[
\rho_2 = 7.4149, \quad u_2 = (0.4695, 0.4255, 0.0000, 0.6215, 0.1124, 0.4467)^T,
\]

\[
\rho_3 = 18.1576, \quad u_3 = (0.3287, 0.4196, 0.4221, 0.3042, 0.4285, 0.5115)^T.
\]

6. Conclusions

We study the CP-matrix approximation problem of approximating a symmetric matrix by one in the intersection of a set of linear constraints and the CP cone. We formulate the problem as a linear optimization problem with the cone of moments and the \( p \)-norm cone (\( p = 1, 2, \infty, \) or \( F) \), and construct a hierarchy of semidefinite relaxations for solving it. An algorithm is presented and its convergence is also studied. If the problem is infeasible, we can get a certificate for that. If the problem is feasible, we can get an approximation matrix; moreover, a CP decomposition of the approximation matrix can also be obtained. Numerical results show that Algorithm 4.1 is efficient in solving the CP-matrix approximation problem.

Acknowledgments

The authors are grateful to the handling editor and the anonymous referees for their valuable comments and suggestions, which have greatly improved the presentation of this paper.
References


Department of Mathematics, and MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, P.R. China
E-mail address: jyfan@sjtu.edu.cn

Department of Mathematics, Shanghai Jiao Tong University, Shanghai 200240, P.R. China
E-mail address: congcongyan@sjtu.edu.cn