Noname manuscript No.
(will be inserted by the editor)

Computing the Completely Positive Factorization via Alternating Minimization

Roger Behling · Hugo Lara · Harry Oviedo

Received: date / Accepted: date

Abstract In this article, we propose a novel alternating minimization scheme for finding completely positive factorizations. In each iteration, our method splits the original factorization problem into two optimization subproblems, the first one being a orthogonal procrustes problem, which is taken over the orthogoal group, and the second one over the set of entrywise positive matrices. We present both a convergence analysis of the method and favorable numerical results.

Keywords Completely positive factorization · alternating minimization · matrix factorization · nonconvex optimization · orthogonal group · Stiefel manifold.

Mathematics Subject Classification (2010) 65K05 · 90C30 · 90C56 · 53C21.

1 Introduction.

Our aim is finding a factorization of a given completely positive matrix. A matrix $A \in \mathbb{R}^{n\times n}$ is called completely positive if for some positive integer r, there exists an entrywise nonnegative matrix $B \in \mathbb{R}^{n\times r}$ such that $A = BB^\top$. Throughout the paper, CP_n denotes the set of all completely positive matrices. Obviously, if $A \in CP_n$ then must be symmetric and have nonnegative entries. For the factorization of a matrix $A \in CP_n$ we face two tasks. First we have to find r such that the completely positive factorization is possible, and then we look for the factor B. In this paper, let us consider a given $A \in CP_n$ and a fixed r for which the following problem is feasible.

$$\text{Find } B \in \mathbb{R}^{n\times r} \quad \text{s.t} \quad A = BB^\top \text{ and } B \geq 0. \tag{1}$$

It is well known that CP_n is a proper convex cone, whose extreme rays are the rank-one matrices $xx^\top$ with $x \in \mathbb{R}^n_+$, that is

$$CP_n = conv\{xx^\top : x \in \mathbb{R}^n, x \geq 0\}.$$

Closely related to CP_n is the class of copositive matrices

$$COP_n := \{A \in \mathbb{S}^{n\times n} : x^\top Ax \geq 0,\ \forall x \in \mathbb{R}^n_+\},$$

where $\mathbb{S}^{n\times n}$ denotes the set of $n \times n$ symmetric matrices. In fact, CP_n is the dual cone of COP_n (see for instance [1]).

R. Behling
Escola de Matemática Aplicada, Fundação Getulio Vargas (FGV/EMAp), Rio de Janeiro, Brazil, and Universidade Federal de Santa Catarina, Blumenau, Brazil. E-mail: rogerbehling@gmail.com

Hugo Lara
Universidade Federal de Santa Catarina, Blumenau, Brazil. E-mail: hugo.lara.urdaneta@ufsc.br

H. Oviedo
Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez (UAI), Santiago de Chile, Chile. E-mail: harry.oviedo@uai.cl

One of the challenges when dealing with CP_n matrices is solving the factorization problem (1) efficiently ([1],[22], [14]). In this regard, we shall propose a novel alternating minimization scheme. To introduce our method, we are going to rewrite (1) as

$$\min_{B,Y} \frac{1}{2}||B - Y||_F^2, \quad \text{s.t} \quad A = BB^\top, \, Y \geq 0. \tag{2}$$

Since (1) has a solution, it must be equivalent to (2), which in turn, can be equivalently posed as

$$\min_{Y \in \mathbb{R}^{n\times r}, X \in \mathbb{R}^{r\times r}} \frac{1}{2}||LX - Y||_F^2, \quad \text{s.t} \quad X^\top X = I, \, Y \geq 0, \tag{3}$$

where $L \in \mathbb{R}^{n\times r}$ is any matrix such that $A = LL^\top$, for instance the one provided by Cholesky. If by chance the chosen L has nonnegative entries, then the original problem (1) is already solved. Otherwise, we look for a solution X, Y of (3), and the correspondent B in (2) is given by $B = LX$. We note that the equality constraints in (3) are known as orthogonality constraints and define the so called Stiefel manifold [2].

The method we propose, referred to as Splitting Alternating Minimization (SAM), seeks a solution X, Y of (3) by generating sequences X_k, Y_k coming from the following alternating minimization scheme

$$X_{k+1} = \arg \min_{X \in \mathbb{R}^{r\times r}} \frac{1}{2}||LX - Y_k||_F^2 \qquad \text{s.t} \quad X^\top X = I, \tag{4}$$

$$Y_{k+1} = \arg \min_{Y \in \mathbb{R}^{n\times r}} \frac{1}{2}||Y - LX_{k+1}||_F^2 \quad \text{s.t} \quad Y \geq 0. \tag{5}$$

We are going to derive a convergence analysis of this method, present favorable numerical experiments, and explain how it differs from other alternating minimization techniques. Moreover, we are going to consider a version of SAM, called SPAM, which employs a proximal point regularization term in the first subproblem. Despite of SAM being inspired by the recent progresses on completely positive factorization given in [14], its numerical performance is notably better. We note that problem (4) is not convex, which brings theoretical obstacles. Alternating minimization under convexity [3] or quasi–strong convexity [9] is well understood.

There is vast motivation for studying problem (1). For instance, copositive factorization appears in applications in combinatorics (block design [15]), also Markovian models of DNA evolution [10], project management [16], economic modeling [18], and more recently, relaxations of combinatorial optimization problems, and nonconvex quadratic optimization problems can be formulated as linear problems over CP_n or COP_n. In [12], it is shown that the tail dependence of a multivariate regularly-varying random vector can be summarized in a so-called tail pairwise dependence matrix, which is a CP_n matrix. Nonnegative factorizations of this matrix can be used to estimate probabilities of extreme events, or to simulate realizations. Further applications of nonnegative factorization of completely positive and copositive matrices can be found on data mining and clustering [20] and in automatic control [4,6].

This paper is organized as follows. Section 2 contains theoretical preliminaries. In section 3, we introduce our alternating minimization method based on suitable Karush–Kuhn–Tucker (KKT) conditions, and section 4 consists of its convergence analysis. Numerical experiments are carried out in section 5, and concluding remarks are presented in the last section.

2 Remarks on the problem

It is worth noting that if we have a factorization $A = BB^\top; B \in \mathbb{R}^{n\times r}, B \geq 0$, then for every $\hat{r} \geq r$ one can select suitable columns and build a matriz $\hat{B} \in \mathbb{R}^{n\times \hat{r}}$, such that $\hat{B} \geq 0$ and $A = \hat{B}\hat{B}^\top$. So, a completely positive factorization is not unique. An important related concept is the cp-rank of a matrix , which is defined as

$$cp(A) := \min_{r}\{r \in \mathbb{N} : A = BB^\top; B \in \mathbb{R}^{n\times r}, B \geq 0\}.$$

If $A \notin CP_n$, then $cp(A) = \infty$. Analogously, the cp-plus-rank of A is

$$cp^+(A) := \min_{r}\{r \in \mathbb{N} : A = BB^\top; B \in \mathbb{R}^{n\times r}, B > 0\}.$$

Therefore, for all $A \in \mathbb{S}^{n\times n}$,

$$rank(A) \leq cp(A) \leq cp^{+}(A). \tag{6}$$

Computing the cp-rank of any $A \in CP_n$ is still an open problem (see [5]), in [7] a tight upper bound was established:

$$cp(A) \leq cp_n := \begin{cases} n & for\ n \in \{2,3,4\} \\ \frac{n}{2}(n+1) - 4 & for \quad n \geq 5 \end{cases}$$

which provides an estimate for r such that A can be factorized by means of $n \times r$ matrix. This is discussed in [14] (see also [17]).

A key ingredient of Groetzner-Dür's approach [14] for solving problem (1) is the orthogonal group: A matrix $X \in \mathbb{R}^{r\times r}$ is called orthogonal if $XX^\top = X^\top X = I$. The set of all orthogonal $r \times r$ matrices, denoted by $\mathcal{O}(r)$, is called orthogonal group, also known as Stiefel manifold. Let us denote $\mathcal{O}^{+}(r) = \{X \in \mathcal{O}(r) : det(X) = 1\}$ and $\mathcal{O}^{-}(r) = \{X \in \mathcal{O}(r) : det(X) = -1\}$. Since $\mathcal{O}(r) = \mathcal{O}^{+}(r) \cup \mathcal{O}^{-}(r)$ we conclude that $\mathcal{O}(r)$ is disconnected. It is compact, but not convex. Also,

$$conv(\mathcal{O}(r)) = \{X \in \mathbb{R}^{n\times r} : XX^\top \preceq I\} = \{X \in \mathbb{R}^{n\times r} : \begin{pmatrix} I & X^\top \\ X & I \end{pmatrix} \succeq 0\},$$

is a convex relaxation for $\mathcal{O}(r)$. The crucial result involving the orthogonal group in [14] establishes that an arbitrary factorization of the target matrix A can be turned into a positive one, by means of orthogonal matrices. This is formally stated below.

Lemma 1 *Let A be a completely positive $n \times n$ matrix, $r \geq cp(A)$ and $L \in \mathbb{R}^{n\times r}$ satisfying $A = LL^\top$ (but possibly not nonnegative), then there exists $X \in \mathcal{O}(r)$ such that $LX \geq 0$ and $A = (LX)(LX)^\top$.*

This lemma justifies the reformulation (3) of problem (1), and its proof is presented, for instance, in [14]. The reformulation (3) is still hard to solve, because the special orthogonal group $\mathcal{O}^{+}(n)$ is not convex.

3 The method of alternating minimization applied to the completely positive factorization.

We recall that our alternating minimization method (4) – (5), aiming to solve (1) is formulated upon the KKT conditions of (3). The Lagrangian function associated to (3), with multipliers $\Lambda \in \mathbb{R}^{r\times r}$ so that $\Lambda^\top = \Lambda$ and $\Theta \in \mathbb{R}^{n\times r}$, is defined as

$$\mathcal{L}(X, Y, \Lambda, \Theta) = \frac{1}{2}||LX - Y||_F^2 - \frac{1}{2}\langle \Lambda, X^\top X - I\rangle - \langle \Theta, Y\rangle, \tag{7}$$

where the inner product between two matrices $Z = [z_{ij}] \in \mathbb{R}^{m\times n}$ and $W = [w_{ij}] \in \mathbb{R}^{m\times n}$ is defined as $\langle Z, W\rangle = tr(Z^\top W) = \sum_{ij} z_{ij}w_{ij}$. Based on the Lagrangian function (7), we get the KKT conditions for (3):

$$L^\top(LX - Y) - X\Lambda = 0 \tag{8a}$$
$$Y - LX - \Theta = 0 \tag{8b}$$
$$X^\top X - I = 0 \tag{8c}$$
$$Y \geq 0 \tag{8d}$$
$$\Theta \geq 0 \tag{8e}$$
$$\Theta_{ij}Y_{ij} = 0, \quad \forall (i,j) \in \mathcal{I}, \tag{8f}$$

where $\mathcal{I} = \{(i,j) \in \mathbb{N}^2 : 1 \leq i \leq n,\ 1 \leq j \leq r\}$.

The KKT system above is indeed a necessary optimality condition for (3). We observe that there are redundancies within the orthogonality constraints, however, this does not prevent a solution of (3) to be a KKT point of this problem. The redundancies occur because the strict upper triangular part of matrix $X^\top X - I$ coincides with its strict lower triangular part. If we rewrite (3) with only upper triangular constraints for the orthogonality, then this optimization problem fulfills the linear independence constraint qualification, guaranteeing the existence of Lagrange multipliers associated to an optimal solution (X^*, Y^*) of the upper triangular version of (3). If λ_{ij}^*, for $i < j$ is one of the Lagrange multipliers, then it is easy to check that

$\frac{1}{2}\lambda_{ij}^*$ and $\frac{1}{2}\lambda_{ji}^*$ are also Lagrange multipliers for the duplicated constraints $x_i^\top x_j = 0$ and $x_j^\top x_i = 0$. So, any solution of (3) has to be a KKT point of this problem.

The following technical result provides a reduced version of the KKT conditions.

Lemma 2 *Suppose that* $(X, Y) \in \mathbb{R}^{r\times r} \times \mathbb{R}^{n\times r}$ *is a local solution of* (3) *then the following conditions are satisfied at* (X, Y)

$$L^\top Y - XY^\top LX = 0 \tag{9a}$$
$$X^\top X - I = 0 \tag{9b}$$
$$Y \geq 0 \tag{9c}$$
$$Y - LX \geq 0 \tag{9d}$$
$$(Y - LX)_{ij} Y_{ij} = 0, \quad \forall (i, j) \in \mathcal{I}. \tag{9e}$$

Proof If (X, Y) is a local solution of (3) then this point must verify the KKTs conditions (8) for some $\Lambda \in \mathbb{R}^{r\times r}$ and $\Theta \in \mathbb{R}^{n\times r}$. Pre–multiplying both sided of (8a) by $X^\top$ and considering the condition (8c), we have the following expression for the Lagrangian multiplier Λ

$$\Lambda = X^\top L^\top (LX - Y). \tag{10}$$

Since Λ should be a symmetric matrix because the constraint $X^\top X - I = 0$ is also symmetric, from (10), we obtain $\Lambda = (LX - Y)^\top LX$. Substituting this result in (8a), we arrive at

$$L^\top (LX - Y) - X(LX - Y)^\top LX = 0.$$

Using (8c) in the above equation, we get

$$L^\top Y - XY^\top LX = 0. \tag{11}$$

And therefore the pair (X, Y) satisfies the equation (9a). In addition, the conditions (9b) and (9c) directly follows from the KKTs conditions (8c) and (8d).

On the other hand, merging the conditions (8b) and (8e) we obtain $0 \leq \Theta = Y - LX$, which proves the relation (9d). Finally, in view of (8b) we have $\Theta = Y - LX$. Therefore, substituting this value in the last KKT condition (8f) we prove (9e). This completes the proof. □

The conditions presented in Lemma 2 are very important since they provide rules to identify possible local minimizers of the optimization problem (3), and thus can be used as stopping criteria for iterative algorithms. Furthermore, it is easy to note that if we have a pair (X, Y) such that the conditions (9) are satisfied then taking $\Lambda = (LX - Y)^\top LX$ and $\Theta = Y - LX$, we recover the KKT conditions (8). Therefore, we conclude that the KKT conditions are equivalent to those presented in Lemma 2. However, the conditions established in Lemma 2 do not depend on dual variables (Λ, Θ), and hence are easier to monitor for primal algorithms (algorithms that only update primal variables).

3.1 The algorithmic framework.

In this subsection, we shall apply the alternating minimization strategy (4)–(5) with the aim of solving the optimization problem (3). Recall that the scheme reads as:

$$X_{k+1} = \arg \min_{X\in\mathbb{R}^{r\times r}} \frac{1}{2}||LX - Y_k||_F^2 \qquad \text{s.t} \quad X^\top X = I,$$
$$Y_{k+1} = \arg \min_{Y\in\mathbb{R}^{n\times r}} \frac{1}{2}||Y - LX_{k+1}||_F^2 \quad \text{s.t} \quad Y \geq 0,$$

where $X_0 \in \mathcal{O}(r)$ and $Y_0 = P_{\geq 0}[LX_0]$. Here, $P_{\geq 0}[M] = \max\{0, M\}$ denotes the orthogonal projection of a matrix $M \in \mathbb{R}^{m\times n}$ onto the set $\mathbb{R}_+^{m\times n}$, and the max function is understood element–wise.

In the following, we are going to discuss the characterization of the solutions of both subproblems above. The global solution of the first subproblem (4) can be obtained from the SVD decomposition $L^\top Y_k = U_k \Sigma_k V_k^\top$, and is given by $X_{k+1} = U_k V_k^\top$. This is proven in the lemma below. As for the second subproblem (5), it is solved by $Y_{k+1} = P_{\geq 0}[LX_{k+1}]$, which is straightforward to prove.

Lemma 3 *Consider the orthogonal constrained least–square problem*

$$\min_{X\in\mathbb{R}^{r\times r}} \frac{1}{2}||LX - Y||_F^2 \quad s.t. \quad X^\top X = I, \tag{12}$$

where $L \in \mathbb{R}^{n\times r}$ *is a given matrix and* $Y \in \mathbb{R}^{n\times r}$ *is fixed. Let* $\bar{X} = UV^\top$ *be the matrix obtained from the singular value decomposition of* $L^\top Y$ *such that* $L^\top Y = U\Sigma V^\top$*. Then,* $\bar{X}$ *is a global solution of* (12)*.*

Proof Problem (12) is known as *orthogonal procrustes problem* and has an analytical solution, which was proved in [23]. Here we give the proof in order to make this article self–contained. Firstly, observe that

$$\begin{aligned}\frac{1}{2}||LX - Y||_F^2 &= \frac{1}{2}||LX||_F^2 - tr(X^\top L^\top Y) + \frac{1}{2}||Y||_F^2\\ &= \frac{1}{2}||L||_F^2 - tr(X^\top L^\top Y) + \frac{1}{2}||Y||_F^2.\end{aligned}$$

The second equality is obtained because the Frobenius norm is invariant under orthogonal transformations. Thus, solving (12) is equivalent to maximizing $\mathcal{J}(X) = tr(X^\top L^\top Y)$ over $\mathcal{O}(r)$. To do so, let $U\Sigma V^\top = L^\top Y$ be a SVD decomposition, then using trace properties we have

$$tr(X^\top L^\top Y) = tr(X^\top U\Sigma V^\top) = tr(V^\top X^\top U\Sigma) = tr(\Sigma U^\top XV) = tr(\Sigma\Omega) = \sum_{i=1}^{n} \sigma_{ii}\omega_{ii}, \tag{13}$$

where $\Omega = U^\top XV$. This matrix Ω is an orthogonal matrix because it is a product of three orthogonal matrices. Since $\Omega \in O(r)$ then any entry of Ω verifies that $\omega_{ij} \le 1$. Hence, the function $F : \mathcal{O}(r) \to \mathbb{R}$ given by $F(\Omega) = tr(\Sigma\Omega)$ is maximized at $\bar{\Omega} = I$ and therefore the solution $\bar{X}$ of (12) is given by $\bar{X} = UIV^\top$, which completes the proof. □

Having stated closed formulas for subproblems (4) and (5), leads us to the algorithm next.

Algorithm 1 Splitting alternating minimization (SAM).

Require: $A = LL^\top \in \mathbb{R}^{n\times n}$, where $L \in \mathbb{R}^{n\times r}$. Initialize with $X_0 \in \mathcal{O}(r)$ and $Y_0 = P_{\ge 0}[LX_0]$, $k = 0$.
1: **while** not convergence **do**
2: Compute a SVD factorization $L^\top Y_k = U_k\Sigma_k V_k^\top$.
3: $X_{k+1} = U_k V_k^\top$.
4: $Y_{k+1} = P_{\ge 0}[LX_{k+1}]$.
5: $k \leftarrow k + 1$.
6: **end while**

Notice that if $\{(X_k, Y_k)\}$ is any infinite sequence generated by Algorithm 1 then the objective function $\mathcal{F} : \mathcal{O}(r)\times\mathbb{R}_+^{n\times r} \to \mathbb{R}$ given by $\mathcal{F}(X, Y) = 0.5||LX - Y||_F^2$ that appears in (3) satisfies the following inequality

$$\mathcal{F}(X_{k+1}, Y_{k+1}) < \mathcal{F}(X_{k+1}, Y_k) \le \mathcal{F}(X_k, Y_k), \tag{14}$$

where the strict inequality above is a consequence of the strict convexity of the second optimization subproblem (5). It follows from the fact that $\{\mathcal{F}(X_k, Y_k)\}$ is bounded below and the relation (14) that the sequence $\{\mathcal{F}(X_k, Y_k)\}$ is convergent. Therefore, our proposed procedure constructs a sequence of feasible points and promotes a monotone decrease in the residual $||LX - Y||_F^2$.

In order to design an alternating minimization method with improved convergence guarantees, we propose a variant of Algorithm 1 inspired by the recently developed proximal point iteration on the Stiefel Manifold [21]. Specifically, we update the iterates using the following iterative process

$$X_{k+1} = \arg\min_{X\in\mathbb{R}^{r\times r}} \frac{1}{2}||LX - Y_k||_F^2 + \frac{1}{2\alpha}||X - X_k||_F^2 \qquad \text{s.t} \quad X^\top X = I, \tag{15}$$

$$Y_{k+1} = \arg\min_{Y\in\mathbb{R}^{n\times r}} \frac{1}{2}||Y - LX_{k+1}||_F^2 \quad \text{s.t} \quad Y \ge 0, \tag{16}$$

where α is a positive real number.

As we can see, the recursive scheme (15)–(16) is closely related to the SAM method. In fact, if α approaches infinity, then the procedure (15)–(16) tends to the SAM method. However, as we will prove in the next section, this proximal version enjoys a stronger global convergence result than the one established for the SAM method, while preserving the numerical efficiency of Algorithm 1. It is straightforward to prove that the optimization subproblem (15) is equivalent to

$$X_{k+1} = \arg \min_{X \in \mathbb{R}^{r\times r}} \left\| \begin{bmatrix} L \\ (1/\sqrt{\alpha})I \end{bmatrix} X - \begin{bmatrix} Y_k \\ (1/\sqrt{\alpha})X_k \end{bmatrix} \right\|_F^2 \quad \text{s.t.} \quad X^\top X = I. \tag{17}$$

Thus, (15) reduces to an orthogonal procrustes problem, and therefore according to Lemma 3, we have that (15) has a closed–form solution given by $X_{k+1} = \tilde{U}_k \tilde{V}_k^\top$, where

$$[L^\top \;\; (1/\sqrt{\alpha})I] \begin{bmatrix} Y_k \\ (1/\sqrt{\alpha})X_k \end{bmatrix} = \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^\top,$$

is a singular value decomposition of the matrix $L^\top Y_k + \alpha^{-1} X_k$.

Algorithm 2 Splitting proximal alternating minimization (SPAM).

Require: $A = LL^\top \in \mathbb{R}^{n\times n}$, where $L \in \mathbb{R}^{n\times r}$. Initialize with $X_0 \in \mathcal{O}(r)$ and $Y_0 = P_{\geq 0}[LX_0]$, $\alpha > 0$, $k = 0$.

```
1: while not convergence do
2:     Compute a SVD factorization L^T Y_k + α^{-1} X_k = U_k Σ_k V_k^T.
3:     X_{k+1} = U_k V_k^T.
4:     Y_{k+1} = P_{≥0}[L X_{k+1}].
5:     k ← k + 1.
6: end while
```

4 Convergence analysis.

In this section, we analyze the strong global convergence of SPAM (Algorithm 2). Furthermore, we provide a convergence result for SAM (Algorithm 1) under an additional condition.

The next proposition establishes convergence to a stationary point of (3) when Algorithm 2 terminates in a finite number of iterations.

Proposition 1 *Let $\{(X_k, Y_k)\}$ be a sequence generated by the Algorithm 2. Suppose that at the k–th iteration $X_{k+1} = X_k$ holds. Then (X_k, Y_k) satisfies the KKT conditions presented in Lemma* (2).

Proof Let $\{(X_k, Y_k)\} \in \mathbb{R}^{r\times r} \times \mathbb{R}^{n\times r}$ a sequence generated by Algorithm 2. Since Algorithm 2 preserves feasibility, we have that $X_k \in \mathcal{O}(r)$ and $Y_k \geq 0$, for all $k \in \mathbb{N}$. Hence, the conditions (9b) and (9c) are verified for any point (X_k, Y_k).

From the iteration procedure, we have that $Y_k - LX_k = P_{\geq 0}[LX_k] - LX_k \geq 0$, which means that the point (X_k, Y_k) fulfills the condition (9d). Now, let $(i, j) \in \mathcal{I}$ an arbitrary pair of indices, so we have the following cases.

i) Suppose $(LX_k)_{ij} \geq 0$. It follows from the updating formula for Y_k that $(Y_k)_{ij} = (P_{\geq 0}[LX_k])_{ij} = (LX_k)_{ij}$, which yields

$$(Y_k - LX_k)_{ij}(Y_k)_{ij} = [(Y_k)_{ij} - (LX_k)_{ij}](Y_k)_{ij} = [(LX_k)_{ij} - (LX_k)_{ij}](Y_k)_{ij} = 0(Y_k)_{ij} = 0.$$

ii) Now, suppose $(LX_k)_{ij} < 0$. In this particular case we have $(Y_k)_{ij} = (P_{\geq 0}[LX_k])_{ij} = 0$, which directly implies that

$$(Y_k - LX_k)_{ij}(Y_k)_{ij} = (Y_k - LX_k)_{ij} 0 = 0.$$

Thus, in both cases we obtain $(Y_k - LX_k)_{ij}(Y_k)_{ij} = 0$, and this is valid for each $(i, j) \in \mathcal{I}$. In view of this expression, we arrive at $(Y_k - LX_k)_{ij}(Y_k)_{ij} = 0$. Therefore, the complementarity condition (9e) holds for the point $\{(X_k, Y_k)\}$.

Finally, it follows from (15) that

$$[L^\top Y_k - X_{k+1}Y_k^T L X_{k+1}] - \alpha^{-1}\left((X_{k+1}-X_k) - X_{k+1}(X_{k+1}-X_k)^\top X_{k+1}\right) = 0. \tag{18}$$

Substituting the hypothesis ($X_{k+1} = X_k$) in the equation above, we get

$$L^\top Y_k - X_k Y_k^\top L X_k = 0,$$

which means that the point (X_k, Y_k) verifies the condition (9a). This completes the proof.

From now on, we study the asymptotic behavior of Algorithm 2 for infinite sequences $\{(X_k, Y_k)\}$ such that $X_{k+1} \neq X_k$ for all $k \geq 0$. Otherwise, Proposition 1 states that Algorithm 2 returns a stationary point of problem (3).

The following lemma shows that the sequences $\{X_k\}$ and $\{Y_k\}$ generated by the proposed iterative process are bounded.

Lemma 4 *Let $\{(X_k, Y_k)\}$ be an infinite sequence generated by Algorithm 2. Then, the sequences $\{X_k\}$ and $\{Y_k\}$ are bounded.*

Proof Since, for all k, X_k is an orthogonal matrix, we have $||X_k||_F = \sqrt{r}$, hence the sequence $\{X_k\}$ is clearly bounded. Now, by the construction of Algorithm 1, we obtain that $Y_k = P_{\geq 0}[LX_k]$, which leads to

$$||Y_k||_F^2 = ||P_{\geq 0}[LX_k]||_F^2 = \sum_{ij} \max\{(LX_k)_{ij}, 0\}^2 \leq \sum_{ij} (LX_k)_{ij}^2 = ||LX_k||_F^2.$$

Thus, $||Y_k||_F^2 \leq ||LX_k||_F^2 = tr(X_k^\top L^\top L X_k) = tr(LL^\top) = tr(A)$, this last relation implies that $||Y_k||_F \leq \sqrt{tr(A)}$, for all $k \in \mathbb{N}$. Therefore, we conclude that the sequence $\{Y_k\}$ is also bounded. □

Next we present a convergence result for SPAM.

Theorem 1 *Let $\{(X_k, Y_k)\}$ be an infinite sequence generated by Algorithm 2. Then, any accumulation point of $\{(X_k, Y_k)\}$ satisfies the KKT conditions* (9). *In particular, whenever $\{(X_k, Y_k)\}$ converges, it converges to a KKT point of* (3).

Proof Let $\{(X_k, Y_k)\}_{k\in\mathcal{K}}$ be a subsequence of $\{(X_k, Y_k)\}$ converging to some (X^*, Y^*). Repeating the steps of the proof of Proposition 1, we have that (X_k, Y_k) satisfies the optimality conditions (9b)–(9c)–(9d) and (9e), for all $k \geq 0$. Applying limits in all these conditions and considering the compactness of the orthogonal group and the fact that the half–space $\mathbb{R}_+^{n\times r}$ is closed, we obtain that (X^*, Y^*) satisfies all the conditions. Hence, it only remains to show that (X^*, Y^*) verifies condition (9a).

From the minimization properties (15) and (16), we get

$$\mathcal{F}(X_{k+1}, Y_{k+1}) \leq \mathcal{F}(X_{k+1}, Y_k) \leq \mathcal{F}(X_k, Y_k) - \frac{1}{2\alpha}||X_{k+1} - X_k||_F^2, \tag{19}$$

Since we are assuming that $X_{k+1} \neq X_k$ for all $k \geq 0$, then relation (19) implies that the sequence $\{\mathcal{F}(X_k, Y_k)\}$ is monotonically decreasing, and therefore it is convergent because $\mathcal{F}(X, Y) \geq 0$, for every $(X, Y) \in \mathbb{R}^{r\times r} \times \mathbb{R}^{n\times r}$. Moreover, by rearranging inequality (19) we obtain

$$||X_{k+1} - X_k||_F^2 \leq 2\alpha\left(\mathcal{F}(X_k, Y_k) - \mathcal{F}(X_{k+1}, Y_{k+1})\right),$$

which leads to

$$\lim_{k\to\infty} ||X_{k+1} - X_k||_F^2 = 0. \tag{20}$$

On the other hand, form the optimality of X_{k+1}, we have

$$(L^\top Y_k - X_{k+1}Y_k^T L X_{k+1}) - \alpha^{-1}\left((X_{k+1}-X_k) - X_{k+1}(X_{k+1}-X_k)^\top X_{k+1}\right) = 0. \tag{21}$$

Observe that the equation (21) can be conveniently rewritten as

$$L^\top Y_k - X_k Y_k^\top L X_k = X_{k+1}Y_k^\top L(X_{k+1}-X_k) + (X_{k+1}-X_k)Y_k^\top L X_k + \alpha^{-1}\left((X_{k+1}-X_k) - X_{k+1}(X_{k+1}-X_k)^\top X_{k+1}\right),$$

this equality yields

$$
\begin{aligned}
||L^\top Y_k - X_k Y_k^\top L X_k||_F &= ||X_{k+1} Y_k^\top L(X_{k+1} - X_k) + (X_{k+1} - X_k) Y_k^\top L X_k||_F + \\
&\quad + \alpha^{-1} ||(X_{k+1} - X_k) - X_{k+1}(X_{k+1} - X_k)^\top X_{k+1}||_F \\
&\leq ||X_{k+1} Y_k^\top L(X_{k+1} - X_k)||_F + ||(X_{k+1} - X_k) Y_k^\top L X_k||_F + \alpha^{-1}(1+r)||X_{k+1} - X_k||_F^2 \\
&\leq ||X_{k+1} Y_k^\top L||_F ||X_{k+1} - X_k||_F + ||X_{k+1} - X_k||_F ||Y_k^\top L X_k||_F + \alpha^{-1}(1+r)||X_{k+1} - X_k||_F^2 \\
&= 2||Y_k^\top L||_F ||X_{k+1} - X_k||_F + \alpha^{-1}(1+r)||X_{k+1} - X_k||_F^2, \\
&\leq (2\, tr(A) + \alpha^{-1}(1+r))\, ||X_{k+1} - X_k||_F, \qquad (22)
\end{aligned}
$$

The second equality above is obtained due to X_{k+1} and X_k being orthogonal matrices, while the last inequality follows from the boundedness of $\{Y_k\}$. Taking limits in (22) and using (20), we arrive at

$$\lim_{k\to\infty} ||L^\top Y_k - X_k Y_k^\top L X_k||_F = 0,$$

which implies that

$$0 = \lim_{k\to\mathcal{K}} ||L^\top Y_k - X_k Y_k^\top L X_k||_F = ||L^\top Y^* - X^*(Y^*)^\top L X^*||_F,$$

that is, $L^\top Y^* - X^*(Y^*)^\top L X^* = 0$. Therefore, the sequence $\{(X_k, Y_k)\}$ asymptotically satisfies the reduced KKT conditions (9), from which the conclusions of the proposition follow readily. So the theorem is proved. □

Now we are going to derive a convergence result for SAM.

Theorem 2 *Let $\{(X_k, Y_k)\}$ be an infinite sequence generated by Algorithm 1. Assume that $\lim_{k\to\infty} X_{k+1} - X_k = 0$. Then, any accumulation point of $\{(X_k, Y_k)\}$ satisfies the KKT conditions* (9)*. In particular, whenever $\{(X_k, Y_k)\}$ converges, it converges to a KKT point of* (3).

Proof Let (X^*, Y^*) an arbitrary accumulation point of $\{(X_k, Y_k)\}$, i.e., let us suppose that $\lim_{k\in\mathcal{K}}(X_k, Y_k) = (X^*, Y^*)$. Again, notice that the four KKTs conditions (9b), (9c), (9d) and (9e) are easy to derive by carrying out the same steps used in the proof of of the previous theorem. Thus we will focus on proving that (X^*, Y^*) also fulfills the condition (9a). Since X_{k+1} is the global solution of the orthogonal constrained least–square problem (12), then X_{k+1} must verify the following equality

$$L^\top (L X_{k+1} - Y_k) - X_{k+1}(L^\top (L X_{k+1} - Y_k))^\top X_{k+1} = 0,$$

or equivalently

$$L^\top Y_k - X_{k+1} Y_k^\top L X_{k+1} = 0. \qquad (23)$$

This last equation can be reformulated as

$$L^\top Y_k - X_k Y_k^\top L X_k = X_{k+1} Y_k^\top L(X_{k+1} - X_k) + (X_{k+1} - X_k) Y_k^\top L X_k,$$

which implies that

$$
\begin{aligned}
||L^\top Y_k - X_k Y_k^\top L X_k||_F &= ||X_{k+1} Y_k^\top L(X_{k+1} - X_k) + (X_{k+1} - X_k) Y_k^\top L X_k||_F \\
&\leq ||X_{k+1} Y_k^\top L(X_{k+1} - X_k)||_F + ||(X_{k+1} - X_k) Y_k^\top L X_k||_F \\
&\leq ||X_{k+1} Y_k^\top L||_F ||X_{k+1} - X_k||_F + ||X_{k+1} - X_k||_F ||Y_k^\top L X_k||_F \\
&= 2||Y_k^\top L||_F ||X_{k+1} - X_k||_F, \\
&\leq 2\, tr(A)\, ||X_{k+1} - X_k||_F. \qquad (24)
\end{aligned}
$$

Applying limits in (24) and keeping in mind the assumption $\lim_{k\to\infty} X_{k+1} - X_k = 0$, we get

$$\lim_{k\to\infty} ||L^\top Y_k - X_k Y_k^\top L X_k||_F = 0,$$

which leads to

$$0 = \lim_{k\to\mathcal{K}} ||L^\top Y_k - X_k Y_k^\top L X_k||_F = ||L^\top Y^* - X^*(Y^*)^\top L X^*||_F,$$

that is, $L^\top Y^* - X^*(Y^*)^\top L X^* = 0$. This completes the demonstration. □

5 Computational results.

In this section, we conduct three groups of experiments to demonstrate the numerical efficiency of our proposals. In particular, we present comparisons with the modified version of the method of alternating projections (MoMAP) developed in [14], the difference-of-convex approach with variable step-size (SpFeas) introduced in [11] (see Algorithm 1 in [11]) and with the FISTA-type method (IPG-FISTA) described in Example 7 of [8]. Both our procedures and the methods of other authors were coded in Matlab. All the experiments were performed on an intel(R) CORE(TM) i7-8750H, CPU 2.20 GHz with 16 GB RAM.

Since for each k-th iteration, the SpFeas algorithm performs a non-monotone line-search to compute a step-size α_k, in the implementation of this method, we use the well-known backtracking heuristic [19] to obtain an appropriate step-size. In addition, for the backtracking scheme we use, as initial step-size, the value $\alpha_k^{\text{ini}} = n/l_t$, where n corresponds to the number of rows of matrix $A \in \mathbb{R}^{n\times n}$ that we want to factorize, and $l_t = \lambda_{\max}(L^\top L)$ is the maximum eigenvalue of $L^\top L$, where $L \in \mathbb{R}^{n\times r}$ satisfies that $A = LL^\top$. Additionally, in our SPAM algorithm we vary the proximal parameter $\alpha \equiv \alpha_k$ at each iteration, using the rule $\alpha_k = \max\{(\frac{\text{tol}_{\text{inq}}}{k^2}), 10^{-20}\}$, where tol_{inq} denotes the tolerance associated with equality constraints (see inequality (26)).

For the SpFeas, MoMAP, SAM and SPAM algorithms, we will use the following stopping criterium

$$\min\{(LX_k)_{i,j}\} \geq -\text{tol}_{\text{inq}}, \tag{25}$$

where $X_k \in \mathcal{O}(r)$ is the orthogonal matrix generated by each algorithm. On the other hand, the IPG-FISTA algorithm is stopped when it finds a matrix X_k satisfying

$$||A - X_k X_k^\top||_F^2 \leq \text{tol}_{\text{eq}}. \tag{26}$$

The difference in the stopping criterium between the algorithms is due to the fact that the SpFeas, MoMAP, SAM and SPAM methods preserve the equality constraint $A = XX^\top$ at each iteration and seeks to satisfy the inequality $X \geq 0$ throughout the iterative process, while IPG-FISTA works in the opposite way. These two tolerances are set to $\text{tol}_{\text{inq}} = \text{tol}_{\text{eq}} = 10^{-8}$, for all the experiments. Let X^* be the estimated solution obtained for an algorithm, then in all the tables reported in this section, *Iter*, *Time*, *Res* and *Min-val* will denotes the number of iterations, the total computational time (in seconds), the residual norm $||A - X^*X^{*\top}||_F$ and the minimum value of X^*, i.e. $\min\{(X^*)_{i,j}\}$, respectively.

5.1 Experiment 1.

In the first experiment, as in [8], we examine the effectiveness and efficiency of the algorithms in factorizing a 40×40 completely positive matrix when the parameter r varies. In particular, we consider the general matrix presented below

$$A_n = \begin{bmatrix} 0 & \mathbf{1}_{n-1}^\top \\ \mathbf{1}_{n-1} & I_{n-1} \end{bmatrix}^\top \begin{bmatrix} 0 & \mathbf{1}_{n-1}^\top \\ \mathbf{1}_{n-1} & I_{n-1} \end{bmatrix} \in \mathbb{R}^{n\times n},$$

where $\mathbf{1}_n \in \mathbb{R}^n$ denotes the all-ones-vector and I_n denotes the n-by-n identity matrix. It is well–known that $A_n \in int(\mathcal{CP}_n)$ for all $n \geq 2$, see reference [24]. In this subsection, we analyze the performance of the five methods on the factorization of A_n with $n = 40$ and $r \in \{40, 51, 61, 71, 81, 91, 101, 111, 121, 131\}$. In order to make this experiment reproducible, the initial orthogonal matrix necessary for the SpFeas, MoMAP, SAM and SPAM algorithms is generated as follows: $X_0 = Q_1 \in \mathbb{R}^{r\times r}$, where $Q_1 \in \mathcal{O}(r)$ is Q-factor of the QR factorization of the matrix $\mathbf{1}_{r,r}$, satisfying $\mathbf{1}_{r,r} = Q_1 R_1$, where R_1 is the corresponding upper triangular matrix and $\mathbf{1}_{r,r}$ denotes the r-by-r all-ones-matrix. Unfortunately, this initial point cannot be used to start iterating the IPG-FISTA method because this approach needs, as a starting point, a matrix belonging to the set $\mathcal{S} = \{X \in \mathbb{R}^{n\times r} : X \geq 0,\ ||X||_F \leq \sqrt{tr(A)}\}$. Thus, for this procedure, we select $X_0 = P_{\mathcal{S}}\,(X_{\text{ini}}(1:n,:))$, where X_{ini} is the initial point used for the rest of methods, $X_{\text{ini}}(1:n,:)$ is the truncated version of X_{ini} obtained after eliminating the rows $n+1, n+2, \ldots, r$ of X_{ini}, and $P_{\mathcal{S}}(\cdot)$ is the projection operator onto $\mathcal{S}$.

Notice that all the considered algorithms (except IPG-FISTA) need as input a matrix $L \in \mathbb{R}^{n\times r}$ such that $A = LL^\top$ whose entries are not necessarily non-negative. To generate this matrix we use the following approach: First, we compute the spectral decomposition of A such that $A = V\Lambda V^\top$ for some orthogonal matrix $V \in \mathcal{O}(n)$ and $\Lambda \in \mathbb{R}^{n\times n}$ the diagonal matrix containing the eigenvalues of A, then notice that the square root of A, i.e. $\bar{L} = V\Lambda^{1/2}V^\top$ verifies $A = \bar{L}\bar{L}^\top$. Nevertheless, $\bar{L}$ is a $n \times n$ matrix, and we need a matrix like $\bar{L}$ but with size $n \times r$. To overcome this issue, we use the column replication strategy. Precisely, we build L by

$$L = \left[\bar{l}_1, \bar{l}_2, \ldots, \bar{l}_{n-1}, \frac{1}{\sqrt{m}}\bar{l}_n, \frac{1}{\sqrt{m}}\bar{l}_n, \ldots, \frac{1}{\sqrt{m}}\bar{l}_n\right],$$

where $\bar{l}_i$ denotes the i-th column of $\bar{L}$ and the vector $\frac{1}{\sqrt{m}}\bar{l}_n$ is repeated $m = r - n + 1$ times. This strategy appears in [17], and it can be proved that $A = LL^\top$ holds.

We want to emphasize that, to design the matrix L, a more advantageous computationally alternative is to conduct the Cholesky decomposition (A_n is always positive definite) of A_n such that $A_n = \tilde{L}\tilde{L}^\top$, and then form L using the column replication strategy over $\tilde{L}$. Nonetheless, it can be verified that the matrix L obtained with this approach is directly a solution to problem (1). Therefore, with a purely academic purpose, in this experiment, we use the matrix L obtained from the spectral decomposition of A_n. However, in practical uses of our algorithms, it is always recomendable to first try to compute the matrix L using the Cholesky-based strategy, since computing a Cholesky factorization has lower computational complexity than computing a spectral decomposition.

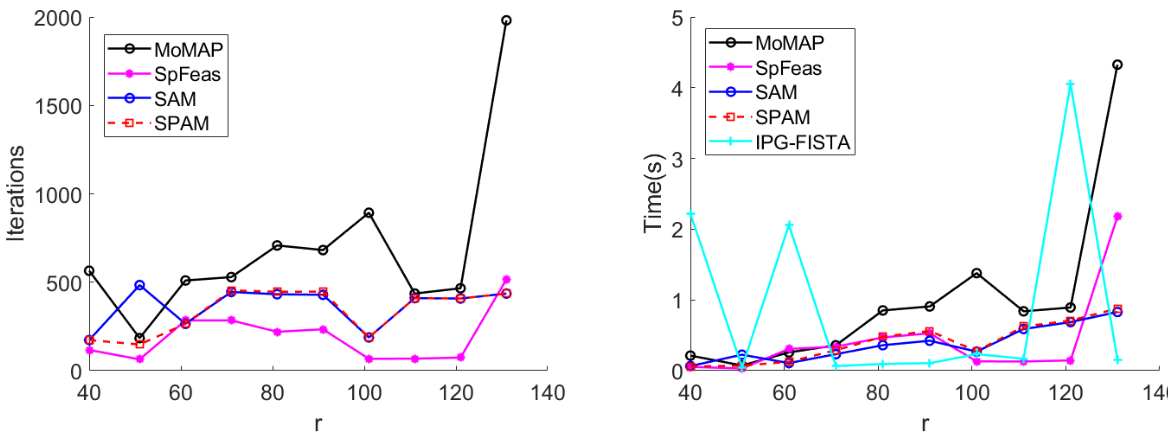
For each value of r, we try to solve problem (1) for the matrix A_{40} with the five algorithms using the stopping criteria explained at the beginning of this section and additionally, we impose a maximum number of iterations of $\text{Iter}_{\max} = 60000$. The numerical results related to this experiment are summarized in Table 1. Form this table, we notice that the IPG-FISTA method executes the maximum number of iterations for $r = 40$ without obtaining a small residual (Res). In addition, for the instances $r = 61$ and $r = 121$ IPG-FISTA reaches the desired precision after a huge number of iterations. For the rest of the problems, it obtains a competitive performance compared to the rest of the methods. In general, we observe that the method that shows a more robust performance in terms of iterations and computational time is the SpFeas method, while our methods remain competitive with SpFeas. In fact, for the instances $r = 61$, $r = 71$, $r = 81$ and $r = 91$ the SAM method achieves an approximation of the solution in less computational time than SpFeas. We also note that for almost all instances our methods converge faster than MoMAP.

In Figure 1, we plot the number of iterations and the computational time required by each method for all the values of r. In the figure associated with the number of iterations, we omit the curve obtained by the IPG-FISTA method since the number of iterations necessary for this method is on another scale and therefore limits the visualization of the curves corresponding to the remaining four methods. As shown in Figure 1, the fastest method in terms of iterations was the SpFeas closely followed by the SAM and SPAM procedures. In particular, we note that for instances $r = 51$, the SAM method obtained an approximation of the solution in 484 iterations, while its proximal version only needs 148 iterations. This example illustrates that SPAM sometimes is more efficient than SAM, which is good news because SPAM has better theoretical guarantees than SAM.

Regarding the computational time curves, we see that the curve related to IPG-FISTA shows an erratic behavior, in some situations it takes a long time to achieve convergence, while for other instances it was the more efficient method. For the rest of the algorithms, roughly speaking, we observe that the computational time increases as fast as r grows, this pattern is expected since all of these approaches must compute a SVD factorization on each iteration, and this decomposition takes longer time as the dimension of the matrix to be factored grows. Finally, we notice that our methods solve the problems in a computational time very close to that required by the SpFeas method. Especially, for the instance $r = 131$, our methods converged faster than SpFeas. Among the four methods that must carry out the SVD factorization (all methods except IPG-FISTA), the MoMAP method was the slowest in almost all instances.

Table 1 Numerical results associated with the completely positive factorization of A_{40} and varying r.

Methods	Iter	Time	Res	Min-val	Iter	Time	Res	Min-val
			$r = 40$				$r = 51$	
MoMAP	565	0.21	1.14e-13	-9.74e-9	184	0.07	2.41e-13	-9.12e-9
SpFeas	117	0.05	1.75e-13	-8.75e-9	64	0.03	1.70e-13	-8.72e-9
SAM	173	0.06	1.45e-13	-9.46e-9	484	0.23	1.09e-13	-9.97e-9
SPAM	173	0.06	1.36e-13	-9.46e-9	148	0.07	1.09e-13	-9.62e-9
IPG-FISTA	60000	2.22	1.95e+00	0	1208	0.05	6.93e-09	0
			$r = 61$				$r = 71$	
MoMAP	510	0.26	1.73e-13	-9.74e-9	529	0.36	1.51e-13	-9.95e-9
SpFeas	284	0.31	1.13e-13	1.96e-4	284	0.34	1.34e-13	-6.93e-16
SAM	265	0.11	8.67e-14	-9.82e-9	445	0.23	1.67e-13	-9.71e-9
SPAM	265	0.13	1.53e-13	-9.82e-9	454	0.29	1.31e-13	-9.82e-9
IPG-FISTA	45040	2.06	9.50e-09	0	1207	0.06	9.28e-09	0
			$r = 81$				$r = 91$	
MoMAP	708	0.85	1.31e-13	-1.00e-8	682	0.91	2.03e-13	-9.91e-9
SpFeas	220	0.47	1.32e-13	-6.49e-16	234	0.53	2.41e-13	-4.24e-16
SAM	432	0.36	1.59e-13	-9.88e-9	429	0.42	9.61e-14	-9.97e-9
SPAM	446	0.48	1.06e-13	-9.74e-9	447	0.56	2.37e-13	-9.68e-9
IPG-FISTA	1207	0.10	9.16e-09	0	1207	0.11	9.07e-09	0
			$r = 101$				$r = 111$	
MoMAP	893	1.38	3.44e-13	-9.97e-9	436	0.84	2.79e-13	-9.79e-9
SpFeas	67	0.13	2.53e-13	-7.88e-9	68	0.13	1.63e-13	-8.85e-9
SAM	188	0.27	1.50e-13	-9.60e-9	410	0.59	1.56e-13	-9.80e-9
SPAM	188	0.28	3.47e-13	-9.60e-9	410	0.62	2.09e-13	-9.80e-9
IPG-FISTA	2674	0.23	9.90e-09	0	1715	0.17	9.21e-09	0
			$r = 121$				$r = 131$	
MoMAP	466	0.89	2.75e-13	-9.81e-9	1981	4.33	3.23e-13	-9.96e-9
SpFeas	74	0.15	3.47e-13	-8.33e-9	516	2.18	1.65e-13	2.36e-15
SAM	408	0.69	2.28e-13	-9.89e-9	436	0.83	2.04e-13	-9.73e-9
SPAM	408	0.71	1.19e-13	-9.89e-9	436	0.88	1.40e-13	-9.73e-9
IPG-FISTA	44467	4.05	9.03e-09	0	1696	0.16	5.67e-09	0



(a) r vs number of iterations. (b) r vs computational time.

Fig. 1 The number of iterations and total computational time required for factorizing A_{40}, for several values of r.

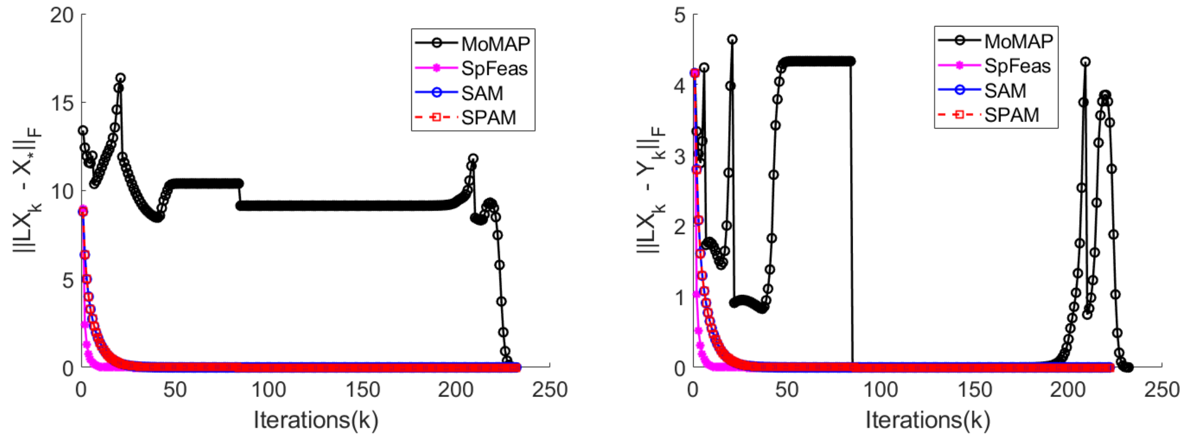
5.2 Experiment 2.

Next, we perform a second experiment addressing the completely positive factorization of the following matrix

$$A = \begin{bmatrix} 41 & 43 & 80 & 56 & 50 \\ 43 & 62 & 89 & 78 & 51 \\ 80 & 89 & 162 & 120 & 93 \\ 56 & 78 & 120 & 104 & 62 \\ 50 & 51 & 93 & 62 & 65 \end{bmatrix}. \tag{27}$$

Table 2 Numerical results corresponding to the completely positive factorization of the matrix A given by (27).

Methods	Iter	Time	Res	Min-val
		$r = 40$		
MoMAP	261	0.0045	2.56e-13	2.41e-04
SpFeas	68	0.0036	5.40e-13	-2.47e-17
SAM	221	0.0039	3.13e-13	6.80e-16
SPAM	221	0.0042	3.13e-13	6.80e-16



(a) Iterations vs $||LX_k - X_*||_F$. (b) Iterations vs $||LX_k - Y_k||_F$.

Fig. 2 Behavior of the MoMAP, SpFeas, SAM and SPAM algorithms solving the Experiment 2.

This matrix is completely positive with $\texttt{cpr}(A) = \texttt{rank}(A) = 3$, see [8]. In this subsection, we analyse the numerical performance af the SpFeas, MoMAP, SAM and SPAM algorithms, solving problem (1) for the matrix present above with $r = 3$. To generate the matrix $L \in \mathbb{R}^{n\times r}$, we compute $\bar{L} \in \mathbb{R}^{n\times n}$ via the spectral decomposition of A, as we explained in the previous subsection. Afterwards, we choose $L = \bar{L}(:, 3 : 5)$, here we are using Matlab notation, i.e. L is the matrix obtained from $\bar{L}$ by removing its first two columns (these two columns are all-zero-vectors). Additionally, we select the r-by-r identity matrix as a starting point for all the iterative procedures. For this experiment, we use the tolerance $\text{tol}_{\text{inq}} = $ 1e-16.

In Figure 2, we illustrate the convergence history of each algorithm. Notice that all the MoMAP, SpFeas, SAM and SPAM methods generate two sequences of matrices $\{X_k\}$ and $\{Y_k\}$ such that $X_k \in \mathcal{O}(r)$ and $Y_k \geq 0$, for all $k \geq 0$. In fact, in all these methods the matrix Y_k is given by $Y_k = P_{X\geq 0}(LX_k) = \max(LX_k, 0)$. Therefore, in Figure 2 (a) we present the curves of iterations against the residual $||LX_k - X^*||_F$ where $X^* \in \mathbb{R}^{n\times r}$ is the solution obtained by each method, that is $X^* = LX_K$ where K is the last iteration of the algorithm. In Figure 2 (b), we give the behavior of the residual norm $||LX_k - Y_k||_F$ along the iterations.

In Figures 2 (a)-(b), we observe a non-monotone pattern in the curves associated with the MoMAP method. However this method achieves convergence to a solution of (1) for the considered matrix. In contrast, the curves associated with the rest of the methods show a monotone decrease for both residual norms. Notice that this monotone descent behavior shown in Figure 2 (b) is expected for our proposed algorithms. In fact this is an intrinsic property of our schemes, see the inequalities (14)-(19). In addition, we also see that the SpFeas achieves convergence with the desired accuracy in less number of iterations than the rest of the methods, while our two approaches converge in almost the same number of iterations as MoMAP. Nonetheless, we have to emphasize that SpFeas is the unique method that does not obtain a matrix with all entries being non-negative. These conclusions are also evidenced in the numerical results reported in Table 2.

Table 3 Numerical results on factorizing random completely positive matrices.

Methods	Iter	Time	Res	Min-val	Iter	Time	Res	Min-val
	$n = 50$, $r = 2n$				$n = 100$, $r = 2n$			
MoMAP	397.0	0.64	6.17e-19	-9.89e-9	295.4	1.99	1.89e-19	-9.85e-9
SpFeas	134.9	0.48	7.10e-19	-9.64e-9	114.2	1.51	2.31e-19	-9.75e-9
SAM	36.1	0.04	6.29e-19	-8.54e-9	29.9	0.14	2.00e-19	-8.92e-9
SPAM	35.8	0.05	6.10e-19	-8.95e-9	29.6	0.17	1.74e-19	-8.86e-9
IPG-FISTA	1235.5	0.11	9.98e-09	0	1558.2	0.47	9.99e-09	0
	$n = 200$, $r = 2n$				$n = 300$, $r = 2n$			
MoMAP	241.8	6.99	5.19e-20	-9.91e-9	169.0	12.14	4.40e-20	-9.83e-9
SpFeas	52.2	2.86	6.79e-20	-6.55e-9	103.3	14.23	4.73e-20	7.90e-9
SAM	25.2	0.54	5.29e-20	-8.79e-9	22.2	1.18	4.56e-20	-8.58e-9
SPAM	24.0	0.63	5.28e-20	-8.49e-9	20.4	1.34	4.58e-20	-8.88e-9
IPG-FISTA	1328.7	1.64	9.99e-09	0	1077.7	5.01	9.99e-09	0
	$n = 50$, $r = 1.5n + 1$				$n = 100$, $r = 1.5n + 1$			
MoMAP	1563.8	1.54	4.34e-19	-2.36e-6	1494.2	5.33	2.33e-19	-9.91e-9
SpFeas	143.6	0.40	5.93e-19	-9.57e-9	120.3	0.89	2.42e-19	-9.56e-9
SAM	47.0	0.05	5.29e-19	-8.74e-9	37.5	0.11	2.25e-19	-8.93e-9
SPAM	46.8	0.06	4.45e-19	-8.89e-9	36.6	0.12	2.26e-19	-9.20e-9
IPG-FISTA	1459.5	0.14	9.98e-09	0	1594.3	0.38	9.99e-09	0

5.3 Experiment 3.

In the third experiment, we tested all the algorithms on 120 randomly generated problems. The matrix $A \in \mathbb{R}^{n\times n}$ is assembled as follows: in each test we randomly generate a matrix $\bar{B} \in \mathbb{R}^{n\times 2n}$ with $\bar{A} = |\bar{B}||\bar{B}|^\top$, where $|M|$ denotes the matrix obtained from M after applying absolute value to all its entries. Since computing a completely positive factorization of $\bar{A}$ is equivalent to solving problem (1) for $\bar{A}/||\bar{A}||_F^2$, then we set $A = \bar{A}/||\bar{A}||_F^2$. In our experiments, we consider the values $n \in \{50, 100, 200, 300\}$ and choose $r \in \{2n, 1.5n + 1, 3n + 1\}$. For each pair (n, r), we execute the algorithms on ten randomly generated problems. Observe that when $r = 2n$ or $r = 3n + 1$, the existence of a positive factorization of A is always guaranteed. In contrast, when $r = 1.5n + 1$ may not be a solution to problem (1). However, we achieve good results for all the considered values of r.

In addition, the initial point for the algorithms SpFeas, MoMAP, SAM and SPAM is randomly generated using the Matlab's commands:

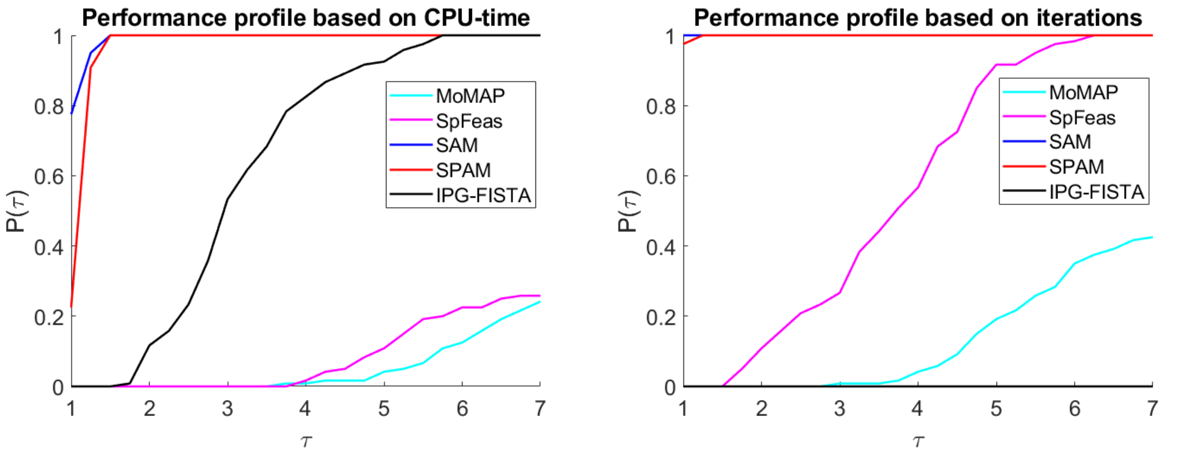
$$X = \texttt{randn}(r); \quad [X_0, \sim] = \texttt{qr}(X), \tag{28}$$

while the initial point for the IPG-FISTA method is constructed by $X_0 = P_{\mathcal{S}}\left(X_{\text{ini}}(1:n,:)\right)$, where X_{ini} is the matrix presented in (28) and $X_{\text{ini}}(1:n,:)$ denotes the truncated version of X_{ini} obtained as explained in the previous subsection. Again for the SpFeas, MoMAP, SAM and SPAM algorithms, we generate the matrix $L \in \mathbb{R}^{n\times r}$ using the spectral decomposition-based approach and the column replication strategy in Subsection 5.1. In tables 3 and 4, we report the average of *Iter*, *Time*, *Res* and *Min-val* obtained by the five algorithms. The information contained in these tables, clearly shows that the most efficient methods, both in terms of iterations and in terms of computational time, were SAM and SPAM. Additionally, we observe that the MoMAP and IPG-FISTA methods required a much higher number of iterations to achieve the desired accuracy than the rest of procedures.

To compare the efficiency of the methods over the total of 120 problems (10 random problems multiplied by the 12 possible combinations of (n, r)) generated, we adopt the performance profile [13] introduced by Dolan and More to illustrate the whole performance of the five methods for the three sets of problems based on Tables 3-4. As shown in Figure 3, the approach SAM has a great advantage over the IPG-FISTA, SpFeas and MoMAP, since with the least number of iterations the SAM method successfully solves 78% of the problems in less computational time than the rest of the methods, while the percentage obtained by our proximal variant (SPAM) was 22%. This suggests that our two proposals were clearly superior to the other procedures.

Table 4 Numerical results on factorizing random completely positive matrices.

Methods	Iter	Time	Res	Min-val	Iter	Time	Res	Min-val
	$n = 200$, $r = 1.5n + 1$				$n = 300$, $r = 1.5n + 1$			
MoMAP	913.7	13.88	6.34e-20	-9.96e-09	668.2	25.59	3.99e-20	-9.93e-09
SpFeas	54.9	1.65	6.98e-20	-6.59e-09	113.4	8.56	4.03e-20	1.21e-07
SAM	31.1	0.38	6.35e-20	-9.05e-09	27.6	0.84	4.10e-20	-8.69e-09
SPAM	29.1	0.42	6.49e-20	-8.83e-09	24.9	0.89	4.05e-20	-8.33e-09
IPG-FISTA	1398.3	1.35	9.99e-09	0	1163.5	4.42	9.99e-09	0
	$n = 50$, $r = 3n + 1$				$n = 100$, $r = 3n + 1$			
MoMAP	286.0	1.00	5.72e-19	-9.73e-09	120.7	1.67	2.10e-19	-9.76e-09
SpFeas	125.9	0.83	8.51e-19	-9.59e-09	105.6	2.87	2.18e-19	-9.75e-09
SAM	27.9	0.08	6.40e-19	-8.74e-09	22.7	0.23	1.98e-19	-8.59e-09
SPAM	27.9	0.08	6.80e-19	-8.83e-09	22.6	0.27	2.08e-19	-8.57e-09
IPG-FISTA	1333.1	0.17	9.99e-09	0	1508.8	0.54	9.99e-09	0
	$n = 200$, $r = 3n + 1$				$n = 300$, $r = 3n + 1$			
MoMAP	96.3	6.93	7.55e-20	-9.74e-09	80.6	16.36	4.63e-20	-9.68e-09
SpFeas	49.9	6.52	7.72e-20	-7.50e-09	91.3	31.32	4.90e-20	9.68e-08
SAM	18.8	1.17	7.35e-20	-8.58e-09	16.1	3.24	4.53e-20	-8.79e-09
SPAM	18.7	1.31	7.28e-20	-7.91e-09	15.9	2.58	4.71e-20	-8.10e-09
IPG-FISTA	1294.5	2.21	1.00e-08	0	1059.0	7.14	9.99e-09	0



(a) Performance profile based on the total computational time.

(b) Performance profile based on the number of iterations.

Fig. 3 Performance profile of all the methods for the experiments contained in Tables 3-4.

6 Concluding remarks

In the present paper, we study the completely positive matrix factorization problem where, given a symmetric matrix, one tries to find a Cholesky-type factorization with an entrywise nonnegative factor.

We propose a novel formulation to address this type of problem, which is inspired from Lemma 1 in the Groetzner-Dür's paper [14]. The proposed formulation consists of the minimization of a convex quadratic function that involves orthogonality and nonnegativity constraints. This model is challenging to solve directly due to the fact that the orthogonality constraints break the convexity of the problem. To deal with this model, we use the variable splitting technique to separate the constraints and then, we introduce and apply an alternating minimization method.

The advantage of this proposal is that the optimization subproblem that arises in each iteration has an analytical solution. We show that the alternating minimization algorithm enjoys strong global convergence results to KKT points under an assumption related to the distance between consecutive iterates. Additionally, we propose an alternating minimization algorithm equipped with a proximal term that has the same global convergence guarantees but without any assumptions.

Finally, we conducted several numerical experiments to compare our algorithms with other existing methods. Our numerical results strongly validate the proposals and provide evidence of their efficiency and faster performance with respect to the competitors.

Acknowledgements.

Harry Oviedo was financially supported by the Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, through the FES startup package for scientific research. Roger Behling was supported by Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ), Grant E-26/201.345/2021

Conflict of interest statement.

The authors declare that they have no conflict of interest.

Data availability statement.

Data sharing not applicable to this article as no datasets were analysed during the current study. In particular, the data studied were generated randomly and we explained how they were explicitly generated.

Ethical approval

Not Applicable.

Funding

Roger Behling has received research support from Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ), Grant E-26/201.345/2021.

Authors' contributions

All authors wrote the text of the manuscript. Harry Oviedo generated Figures 1-3 and Tables 1-4. All authors reviewed the article.

References

1. Berman A and Shaked-Monderer N. *Completely positive matrices.* World Scientific Publishing Co., Inc., River Edge, 2002.
2. P.-A Absil, R. Mahony, and R Sepulchre. *Optimization Algorithms on Matrix Manifolds.* Princeton, 2008.
3. Amir Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1):185–209, 2015. doi: `https://doi.org/10.1137/13094829X`.
4. Abraham Berman, Christopher King, and Robert Shorten. A characterisation of common diagonal stability over cones. *Linear and Multilinear Algebra*, 60(10):1117–1123, 2012.
5. Avi Berman, Mirjam Dür, and Naomi Shaked-Monderer. Open problems in the theory of completely positive and copositive matrices. *Electronic Journal of Linear Algebra*, 29:46–58, 2015.
6. Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007.
7. Immanuel M. Bomze, Peter J. C. Dickinson, and Georg Still. The structure of completely positive matrices according to their cp-rank and cp-plus-rank. *Linear Algebra and its Applications*, 482:191–206, 2015.
8. Radu Ioan Boţ and Dang-Khoa Nguyen. Factorization of completely positive matrices using iterative projected gradient steps. *Numerical Linear Algebra with Applications*, 28(6):e2391, 2021.
9. Jakub Wiktor Both. On the rate of convergence of alternating minimization for non-smooth non-strongly convex optimization in banach spaces. *Optimization Letters*, 16(2):729–743, 2022.
10. Kelly C. A test of the markovian model of dna evolution. *Biometrics*, 50(3):653–664, 1994.

11. Chen Chen, Ting Kei Pong, Lulin Tan, and Liaoyuan Zeng. A difference-of-convex approach for split feasibility with applications to matrix factorizations and outlier detection. *Journal of Global Optimization*, 78(1):107–136, 2020. doi: `https://doi.org/10.1007/s10898-020-00899-8`.
12. Cooley D. and Thibaud E. Decompositions of dependence for high-dimensional extremes. *Biometrika*, 106(3):587–604, 2019.
13. Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.
14. Patrick Groetzner and Mirjam Dür. A factorization method for completely positive matrices. *Linear Algebra and its Applications*, 591:1–24, 2020.
15. Hall M Jr. and Newman M. Copositive and completely positive quadratic forms. *Proc Camb Philos Soc*, 59:329–339, 1963.
16. Natarajan K., Teo CP., and Zheng Z. Mixed zero-one linear programs under objective uncertainty: a completely positive representation. *Operation Research*, 59(3):713–728, 2011.
17. Zhijian Lai and Akiko Yoshise. Completely positive factorization in orthogonality optimization via smoothing method. *arXiv preprint arXiv:2107.01538*, 2021.
18. Gray L.J. and Wilson D.G. Nonnegative factorization of positive semidefinite nonnegative matrices. *Linear Algebra and its Applications*, 31:119–127, 1980.
19. Jorge Nocedal and Stephen J Wright. *Numerical optimization.* Springer, 1999. doi: `https://doi.org/10.1007/978-0-387-40065-5`.
20. SIAM International Conference on Data Mining, editor. *On the equivalence of nonnegative matrix factorization and specrral clustering.* In H. Hargupta and J. Srivastava and C. Kamath and A. Goodman (Eds).
21. Harry Oviedo. Proximal point algorithm on the stiefel manifold. *Preprint in Optimization–Online. http://www. optimization-online. org/DB_ FILE/2021/05/8401. pdf*, 2021.
22. Dickinson P.J.C. and Mirjam Dür. Linear-time complete positivity detection and decomposition of sparse matrices. *SIAM Journal on Matrix Analysis and Applications*, 33(3):701–720, 2012.
23. Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. doi: `https://doi.org/10.1007/BF02289451`.
24. Bryan Shader, Naomi Shaked-Monderer, and Daniel B Szyld. Nearly positive matrices. *Linear Algebra and its Applications*, 449:520–544, 2014.