Completely Positive Factorization by Riemannian Smoothing Method*

ZHIJIAN LAI† AKIKO YOSHISE‡

July 2021
Revised December 2021

Abstract

Copositive optimization is a special case of convex conic programming, and it optimizes a linear function over the cone of all completely positive matrices under linear constraints. Copositive optimization provides powerful relaxations of NP-hard quadratic problems or combinatorial problems, but there are still many open problems regarding copositive or completely positive matrices. In this paper, we focus on one of such open problems; finding a completely positive (CP) factorization for a given completely positive matrix. We treat it as a nonsmooth Riemannian optimization, i.e., a minimization of a nonsmooth function over the Riemannian manifolds. To solve this problem, we present a general smoothing framework for nonsmooth Riemannian optimization and guarantee convergence to a stationary point of the original problem. An advantage is that we can implement it quickly with minimal effort by directly using the existing standard smooth Riemannian solvers, such as Manopt. Numerical experiments show the efficiency of our method especially for large-scale CP factorizations.

Keywords: completely positive factorization, smoothing method, nonsmooth Riemannian optimization

AMS: 15A23, 15B48, 90C48, 90C59

1 Introduction

The space of $n \times n$ real symmetric matrices $\mathcal{S}_n$ is endowed with the trace inner product $\langle A, B\rangle := \mathrm{trace}(AB)$. A matrix $A \in \mathcal{S}_n$ is called *completely positive* if for some $r \in \mathbb{N}$ there exists an entrywise nonnegative matrix $B \in \mathbb{R}^{n\times r}$ such that $A = BB^\top$, and we call B a *CP factorization* of A. We define $\mathcal{CP}_n$ as the set of all $n \times n$ completely positive matrices, equivalently characterized as

$$\mathcal{CP}_n := \{BB^\top \in \mathcal{S}_n \mid B \text{ is a nonnegative matrix } \} = \mathrm{conv}\{xx^\top \mid x \in \mathbb{R}^n_+\},$$

where $\mathrm{conv}(S)$ denotes the convex hull of a given set S. We denote the set of all $n \times n$ copositive matrices by $\mathcal{COP}_n := \{A \in \mathcal{S}_n \mid x^\top Ax \geq 0 \text{ for all } x \in \mathbb{R}^n_+\}$. It is known that $\mathcal{COP}_n$ and $\mathcal{CP}_n$ are duals of each other under the trace inner product [30, Theorem 2.6]. Both $\mathcal{CP}_n$ and $\mathcal{COP}_n$ are proper convex cones [24, Chapter 5]. For any positive integer n, we have the following inclusion relationship among other important cones in conic optimization:

$$\mathcal{CP}_n \subseteq \mathcal{S}^+_n \cap \mathcal{N}_n \subseteq \mathcal{S}^+_n \subseteq \mathcal{S}^+_n + \mathcal{N}_n \subseteq \mathcal{COP}_n,$$

where $\mathcal{S}^+_n$ is the cone of $n \times n$ symmetric positive semidefinite matrices and $\mathcal{N}_n$ is the cone of $n \times n$ symmetric nonnegative matrices. See the monograph [1] for a comprehensive description of $\mathcal{CP}_n$ and $\mathcal{COP}_n$.

*This paper was previously titled "Completely Positive Factorization in Orthogonality Optimization via Smoothing Method."

†Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan. E-mail:s2130117@s.tsukuba.ac.jp

‡Corresponding author. Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan. E-mail:yoshise@sk.tsukuba.ac.jp

Conic optimization is a subfield of convex optimization that studies minimization of linear functions over proper cones. Here, if the proper cone is $\mathcal{CP}_n$ or its dual cone $\mathcal{COP}_n$, we call the conic optimization problem a *copositive programming* problem. Copositive programming is closely related to many nonconvex, NP-hard quadratic and combinatorial optimizations. For example, consider the so-called standard quadratic optimization,

$$\min\{x^\top Mx \mid \mathbf{e}^\top x = 1, x \in \mathbb{R}^n_+\}, \tag{1}$$

where $M \in \mathcal{S}_n$ is possibly not positive semidefinite and $\mathbf{e}$ is the all-ones vector. Bomze et al. [9] showed that the following completely positive reformulation,

$$\min\{\langle M, X\rangle \mid \langle E, X\rangle = 1, X \in \mathcal{CP}_n\},$$

where E is the all-ones matrix, is equivalent to (1). Burer [15] reported a more general result, where any quadratic problem with binary and continuous variables can be rewritten as a linear program over $\mathcal{CP}_n$. As an application to combinatorial problems, consider the problem of computing the independence number $\alpha(G)$ of a graph G with n nodes. De Klerk and Pasechnik [22] showed that

$$\alpha(G) = \max\{\langle E, X\rangle \mid \langle A + I, X\rangle = 1, X \in \mathcal{CP}_n\},$$

where A is the adjacency matrix of G. For surveys on applications of copositive programming, see [6, 10, 16, 27].

The difficulty of the above problems lies entirely in the completely positive conic constraint. Note that because neither $\mathcal{COP}_n$ nor $\mathcal{CP}_n$ is self-dual, the primal-dual interior point method for conic optimization does not work as is. Besides this, there are many open problems related to completely positive cones. One is checking membership in $\mathcal{CP}_n$, which was shown to be NP-hard by [26]. Computing or estimating the cp-rank, as defined later in (3), is also an open problem. We refer the reader to [4, 27] for a detailed discussion of those unresolved issues.

In this paper we focus on finding a CP factorization for a given $A \in \mathcal{CP}_n$, i.e., the *CP factorization problem*:

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

which seems to be closely related to the membership problem $A \in \mathcal{CP}_n$. Sometimes, a matrix is shown to be completely positive through duality, or rather, $\langle A, X\rangle \geqslant 0$ for all $X \in \mathcal{COP}_n$, but in this case, a CP factorization will not necessarily be obtained.

1.1 Related work on CP factorization

Various methods of solving CP factorization problems have been studied. Jarre and Schmallowsky [32] stated a criterion for complete positivity, based on the augmented primal dual method to solve a particular second-order cone problem. Dickinson and Dür [25] dealt with complete positivity of matrices that possess a specific sparsity pattern and proposed a method for finding CP factorizations of these special matrices that can be performed in linear time. Nie [35] formulated the CP factorization problem as an $\mathcal{A}$-truncated K-moment problem, for which the author developed an algorithm that solves a series of semidefinite optimization problems. Sponsel and Dür [42] considered the problem of projecting a matrix onto $\mathcal{CP}_n$ and $\mathcal{COP}_n$ by using polyhedral approximations of these cones. With the help of these projections, they devised a method to compute a CP factorization for any matrix in the interior of $\mathcal{CP}_n$. Bomze [7] extended it to a CP factorization of an $n \times n$ matrix containing A as a principal submatrix when a known CP factorization of an $(n-1) \times (n-1)$ matrix A is given. Sikirić, Schürmann and Vallentin [40] developed a simplex-like method for a rational CP factorization that works if the input matrix allows a rational CP factorization.

In 2020, Groetzner and Dür [29] applied the alternating projection method to the CP factorization problem by posing it as an equivalent feasibility problem (see (FeasCP)). Shortly afterwards, Chen and Pong et al. [18] reformulated the split feasibility problem as a difference-of-convex optimization problem and solved (FeasCP) as a specific application. In fact, we will solve this equivalent feasibility problem (FeasCP) by other means in this paper. In 2021, Boţ and Nguyen

[12] proposed a projected gradient method with relaxation and inertia parameters for the CP factorization problem, aimed at solving

$$\min_X \{\|A - XX^\top\|^2 \mid X \in \mathbb{R}^{n\times r}_+ \cap \mathcal{B}(0, \sqrt{\mathrm{trace}(A)})\}, \tag{2}$$

where $\mathcal{B}(0,\varepsilon) := \{X \in \mathbb{R}^{n\times r} \mid \|X\| \le \varepsilon\}$ is the closed ball centered at 0. The authors argued that its optimal value is zero if and only if $A \in \mathcal{CP}_n$.

1.2 Our contributions and organization of the paper

Inspired by the idea of Groetzner and Dür [29], wherein (CPfact) is equivalent to a feasibility problem containing an orthogonality constraint, we treat the latter as a nonsmooth Riemannian optimization and solve it through a general Riemannian smoothing method. Our contributions are summarized as follows:

1. To the best of our knowledge, this study is the first to handle CP factorization via Riemannian optimization, for which various techniques have been developed.

2. In particular, we present a general framework of Riemannian smoothing for the nonsmooth Riemannian optimization problem and guarantee convergence to a stationary point of the original problem.

3. We apply the general framework of the Riemannian smoothing to CP factorization. Numerical experiments clarify that our method is competitive with other efficient CP factorization methods, especially for large-scale matrices.

In section 2, we review the process to reconstruct (CPfact) into another feasibility problem; in particular, we take a different approach to this problem from those in other studies. In section 3, we describe the general frame of smoothing method on Riemannian optimization. To apply it to the CP factorization problem, we employ a smoothing function named LogSumExp. Section 4 is a collection of numerical experiments.

2 Preliminaries

2.1 cp-rank and cp-plus-rank

First, let us recall some basic properties of completely positive matrices. Generally, one can have many CP factorizations for a given A even if the numbers of columns are distinct, which gives rise to the following definition. The cp-rank of $A \in \mathcal{S}_n$, denoted by $\mathrm{cp}(A)$, is defined as

$$\mathrm{cp}(A) := \min_B \{r \in \mathbb{N} \mid A = BB^\top, B \in \mathbb{R}^{n\times r}, B \ge 0\}, \tag{3}$$

where $\mathrm{cp}(A) = \infty$ if $A \notin \mathcal{CP}_n$. Similarly, we can define the cp-plus-rank as

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

Immediately, for all $A \in \mathcal{S}_n$, we have

$$\mathrm{rank}(A) \le \mathrm{cp}(A) \le \mathrm{cp}^+(A). \tag{4}$$

Every CP factorization B of A is of the same rank as A since $\mathrm{rank}(XX^\top) = \mathrm{rank}(X)$ holds for any matrix X. The first inequality of (4) comes from the fact that for any CP factorization B,

$$\mathrm{rank}(A) = \mathrm{rank}(B) \le \text{ the number of columns of } B.$$

The second is trivial by definition.

Note that computing or estimating the cp-rank of any given $A \in \mathcal{CP}_n$ is still an open problem. The following result gives a tight upper bound of the cp-rank for $A \in \mathcal{CP}_n$ in terms of order n.

Theorem 2.1 (Bomze, Dickinson, and Still [8, Theorem 4.1]). *For all $A \in \mathcal{CP}_n$, we have*

$$\mathrm{cp}(A) \le \mathrm{cp}_n := \begin{cases} n & \text{for } n \in \{2,3,4\} \\ \frac{1}{2}n(n+1) - 4 & \text{for } n \ge 5. \end{cases}$$

The following result is useful for distinguishing completely positive matrices in either the interior or on the boundary of $\mathcal{CP}_n$.

Theorem 2.2 (Dickinson [23, Theorem 3.8]). *We have*

$$\begin{aligned}\operatorname{int}(\mathcal{CP}_n) &= \{A \in \mathcal{S}_n \mid \operatorname{rank}(A) = n, \operatorname{cp}^+(A) < \infty\} \\ &= \{A \in \mathcal{S}_n \mid \operatorname{rank}(A) = n, A = BB^\top, B \in \mathbb{R}^{n\times r}, B \geq 0, \\ &\qquad b_j > 0 \text{ for at least one column } b_j \text{ of } B\}.\end{aligned}$$

2.2 CP factorization as a feasibility problem

Groetzner and Dür [29] reformulated the CP factorization problem as an equivalent feasibility problem containing an orthogonality constraint.

Given $A \in \mathcal{CP}_n$, we can easily get another CP factorization $\widehat{B}$ with r' columns for every integer $r' \geq r$, if we also have a CP factorization B with r columns. The simplest way to construct such an $n \times r'$ matrix $\widehat{B}$ is to append $k := r' - r$ zero columns to B, i.e., $\widehat{B} := [B, 0_{n\times k}] \geq 0$. Another way is called *column replication*, i.e.,

$$\widehat{B} := [b_1, \ldots, b_{n-1}, \underbrace{\frac{1}{\sqrt{m}} b_n, \ldots, \frac{1}{\sqrt{m}} b_n}_{m:=r'-n+1 \text{ columns}}], \tag{5}$$

where b_i denotes the i-th column of B. It is easy to see that $\widehat{B}\widehat{B}^\top = BB^\top = A$. The next lemma is easily derived from the previous discussion, and it implies that there always exists an $n \times \operatorname{cp}_n$ CP factorization for any $A \in \mathcal{CP}_n$.

Lemma 2.3. *Suppose that $A \in \mathcal{S}_n$, $r \in \mathbb{N}$. Then, $r \geq \operatorname{cp}(A)$ if and only if A has a CP factorization B with r columns.*

Let $\mathcal{O}(r)$ denote the orthogonal group of order r, i.e., the set of $r \times r$ orthogonal matrices. The following lemma is essential to our study. (Note that many authors have proved the existence of such an orthogonal matrix X (see, e.g., [43, Lemma 1] and [29, Lemma 2.6]).)

Lemma 2.4. *Let $B, C \in \mathbb{R}^{n\times r}$. $BB^\top = CC^\top$ if and only if there exists $X \in \mathcal{O}(r)$ with $BX = C$.*

The next proposition puts the previous two lemmas together.

Proposition 2.5. *Let $A \in \mathcal{CP}_n, r \geq \operatorname{cp}(A), A = \bar{B}\bar{B}^\top$, where $B \in \mathbb{R}^{n\times r}$ may possibly be not nonnegative. Then, there exists an orthogonal matrix $X \in \mathcal{O}(r)$ such that $\bar{B}X \geq 0$ and $A = (\bar{B}X)(\bar{B}X)^\top$.*

This proposition tells us that one can find an orthogonal matrix X which can turn a "bad" factorization $\bar{B}$ into a "good" factorization $\bar{B}X$. Thus, the task of finding a CP factorization of A can be formulated as the following feasibility problem:

$$\text{Find } X \text{ s.t. } \bar{B}X \geq 0 \text{ and } X \in \mathcal{O}(r), \tag{FeasCP}$$

where $r \geq \operatorname{cp}(A)$, $\bar{B} \in \mathbb{R}^{n\times r}$ is an arbitrary initial factorization $A = \bar{B}\bar{B}^\top$ and may possibly be not nonnegative.

We should notice that the condition $r \geq \operatorname{cp}(A)$ is necessary; otherwise, (FeasCP) has no solution even if $A \in \mathcal{CP}_n$. Regardless of the exact $\operatorname{cp}(A)$ which is often unknown, one can use cp_n. Note that finding an initial matrix $\bar{B}$ is not difficult. Since a completely positive matrix is necessarily positive semidefinite, one can use Cholesky decomposition or spectral decomposition and then extend it to r columns by using (5). The following corollary shows that the feasibility of (FeasCP) is precisely a criterion for complete positivity.

Corollary 2.6. *Set $r \geq \operatorname{cp}(A)$, $\bar{B} \in \mathbb{R}^{n\times r}$ an arbitrary initial factorization of A. Then $A \in \mathcal{CP}_n$ if and only if (FeasCP) is feasible. In this case, for any feasible solution X, $\bar{B}X$ is a CP factorization of A.*

In this study, solving (FeasCP) is the key to finding a CP factorization, but it is still a hard problem because $\mathcal{O}(r)$ is nonconvex.

2.3 Approaches to solving (FeasCP)

Groetzner and Dür [29] applied the so-called alternating projections method to (FeasCP). They defined the polyhedral cone, $\mathcal{P} := \{X \in \mathbb{R}^{r\times r} : \bar{B}X \geq 0\}$, and rewrote (FeasCP) as

$$\text{Find } X \text{ s.t. } X \in \mathcal{P} \cap \mathcal{O}(r).$$

The alternating projections method is as follows: choose a starting point $X_0 \in \mathcal{O}(r)$; then compute $P_0 = \text{proj}_{\mathcal{P}}(X_0)$ and $X_1 = \text{proj}_{\mathcal{O}(r)}(P_0)$, and iterate this process. Computing the projection onto $\mathcal{P}$ amounts to solving a second-order cone problem (SOCP), while computing the projection onto $\mathcal{O}(r)$ amounts to a singular value decomposition. Note that we need to solve an SOCP alternately at every iteration, which is still expensive in practice. A modified version without convergence involves calculating an approximation of $\text{proj}_{\mathcal{P}}(X_k)$ by using the Moore-Penrose inverse of $\bar{B}$; for details, see [29, Algorithm 2].

Our way is to use the optimization form. Here, we denote by $\max(\cdot)$ (resp. $\min(\cdot)$) the *max-function* (resp. *min-function*)) that selects the largest (resp. smallest) entry of a vector or matrix. Notice that $-\min(\cdot) = \max(-(\cdot))$. We associate (FeasCP) with the following optimization problem:

$$\max_{X\in\mathcal{O}(r)} \{\min(\bar{B}X)\}.$$

For consistency of notation, we turn the maximization into a minimization:

$$\min_{X\in\mathcal{O}(r)} \{\max(-\bar{B}X)\}. \tag{OptCP}$$

The feasible set $\mathcal{O}(r)$ is known to be compact [31, Observation 2.1.7]. In accordance with the extreme value theorem [37, Theorem 4.16], (OptCP) can obtain the global minimum, say t. Summarizing these observations together with Corollary 2.6 yields the following proposition.

Proposition 2.7. *Set $r \geq \text{cp}(A)$, and let $\bar{B} \in \mathbb{R}^{n\times r}$ be an arbitrary initial factorization of A. Then, the following statements are equivalent:*

1. *$A \in \mathcal{CP}_n$.*
2. *(FeasCP) is feasible.*
3. *In (OptCP), there exists a feasible solution X such that $\max(-\bar{B}X) \leq 0$; alternatively, $\min(\bar{B}X) \geq 0$.*
4. *In (OptCP), the global minimum $t \leq 0$.*

3 Riemannian smoothing method

The problem of minimizing a real-valued function over a Riemannian manifold $\mathcal{M}$, which is called Riemannian optimization, has been actively studied during the last few decades. In particular, the Stiefel manifold,

$$\text{St}(n,p) = \{X \in \mathbb{R}^{n\times p} \mid X^\top X = I\},$$

(when $n = p$, it reduces to the orthogonal group) is an important case and is our main interest here. We treat the CP factorization problem, i.e., (OptCP) as a minimization of a nonsmooth function over a Riemannian manifold, for which variants of subgradient methods [11], proximal gradient methods [21], and the alternating direction method of multipliers (ADMM) [33] have been studied.

Smoothing methods [19], which use a parameterized smoothing function to approximate the objective function, are effective on a class of nonsmooth optimizations in Euclidean space. Recently, Zhang et al. [45] extended a smoothing steepest descent method to the case of Riemannian submanifolds in $\mathbb{R}^n$. This is not the first time that smoothing methods have been studied on manifolds. Liu and Boumal [34] extended the augmented Lagrangian method and exact penalty method to the Riemannian case. The latter leads to a nonsmooth Riemannian optimization to

which they applied smoothing techniques. Cambier and Absil [17] dealt with the problem of robust low-rank matrix completion by Riemannian optimization, wherein they applied a smoothing conjugate gradient method.

In this section, we propose a general Riemannian smoothing method and apply it to the CP factorization problem.

3.1 Notation and terminology of Riemannian optimization

Let us briefly review some concepts in Riemannian optimization, following the notation of [13]. Throughout this paper, $\mathcal{M}$ will refer to a complete Riemannian submanifold of Euclidean space $\mathbb{R}^n$. Thus, $\mathcal{M}$ is endowed with a Riemannian metric induced from the Euclidean inner product, i.e., $\langle \xi, \eta \rangle_x := \xi^T \eta$ for any $\xi, \eta \in \mathrm{T}_x\mathcal{M}$, where $\mathrm{T}_x\mathcal{M} \subseteq \mathbb{R}^n$ is the tangent space to $\mathcal{M}$ at x. The Riemannian metric induces the usual Euclidean norm $\|\xi\|_x := \|\xi\| = \sqrt{\langle \xi, \xi \rangle_x}$ for $\xi \in \mathrm{T}_x\mathcal{M}$. The tangent bundle $\mathrm{T}\mathcal{M} := \bigsqcup_{x \in \mathcal{M}} \mathrm{T}_x\mathcal{M}$ is a disjoint union of the tangent spaces of $\mathcal{M}$. Let $f : \mathcal{M} \to \mathbb{R}$ be a smooth function on $\mathcal{M}$. The Riemannian gradient of f is a vector field $\operatorname{grad} f$ on $\mathcal{M}$ that is uniquely defined by the identities: for all $(x, v) \in \mathrm{T}\mathcal{M}$,

$$\mathrm{D}f(x)[v] = \langle v, \operatorname{grad} f(x) \rangle_x$$

where $\mathrm{D}f(x) : \mathrm{T}_x\mathcal{M} \to \mathrm{T}_{f(x)}\mathbb{R} \cong \mathbb{R}$ is the differential of f at $x \in \mathcal{M}$. Since $\mathcal{M}$ is an embedded submanifold of $\mathbb{R}^n$, we have a simpler statement for f that is also well defined on the whole $\mathbb{R}^n$:

$$\operatorname{grad} f(x) = \operatorname{Proj}_x(\nabla f(x)),$$

where $\nabla f(x)$ is the usual gradient in $\mathbb{R}^n$ and Proj_x denotes the orthogonal projector from $\mathbb{R}^n$ to $\mathrm{T}_x\mathcal{M}$. For a subset $D \subseteq \mathbb{R}^n$, the function $h \in C^1(D)$ is smooth, i.e., continuously differentiable on D. Given a point $x \in \mathbb{R}^n$ and $\delta > 0$, $\mathcal{B}(x, \delta)$ denotes a closed ball of radius δ centered at x. $\mathbb{R}_{++}$ denotes the set of positive real numbers. We use subscript notation x_i to select the ith entry of a vector and superscript notation x^k to designate an element in a sequence $\{x^k\}$.

3.2 Ingredients

Now let us consider nonsmooth Riemannian optimization (NRO):

$$\min_{x \in \mathcal{M}} f(x), \tag{NRO}$$

where $\mathcal{M} \subseteq \mathbb{R}^n$ and $f(x) : \mathbb{R}^n \to \mathbb{R}$ is a *proper lower semi-continuous* function (maybe nonsmooth or even non-Lipschitzian) on $\mathbb{R}^n$. For convenience, smooth Riemannian optimization (SRO) refers to (NRO) when $f(\cdot)$ is continuously differentiable on $\mathbb{R}^n$. To avoid confusion in this case, we use g instead of f,

$$\min_{x \in \mathcal{M}} g(x). \tag{SRO}$$

Throughout this subsection, we will refer to many of the concepts in [45].

First, let us review the usual concepts and properties related to generalized subdifferentials in $\mathbb{R}^n$. For a proper lower semi-continuous function $f : \mathbb{R}^n \to \mathbb{R}$, the *Fréchet subdifferential* and the *limiting subdifferential of* f *at* $x \in \mathbb{R}^n$ are defined as

$$\begin{aligned} \hat{\partial} f(x) := \{ \nabla h(x) \mid \exists \delta > 0 \text{ such that } h \in C^1(\mathcal{B}(x, \delta)) \text{ and} \\ f - h \text{ attains a local minimum at } x \text{ on } \mathbb{R}^n \}, \end{aligned}$$

$$\partial f(x) := \{ \lim_{\ell \to \infty} v^\ell \mid v^\ell \in \hat{\partial} f\left(x^\ell\right), \left(x^\ell, f\left(x^\ell\right)\right) \to (x, f(x)) \}.$$

The definition of $\hat{\partial} f(x)$ above is not the standard one: the standard definition follows [36, 8.3 Definition]. But these definitions are equivalent by [36, 8.5 Proposition]. For locally Lipschitz functions, the *Clarke subdifferential at* $x \in \mathbb{R}^n$, $\partial^\circ f(x)$, is the convex hull of the limiting subdifferential. Their relationship is as follows:

$$\hat{\partial} f(x) \subseteq \partial f(x) \subseteq \partial^\circ f(x).$$

Notice that if f is convex, $\partial f(x)$ and $\partial^\circ f(x)$ coincide with the classical subdifferential in convex analysis [36, 8.12 Proposition].

Example 1 (Bagirov, Bagirov and Mäkelä [3, Theorem 3.23]). From a result on the pointwise max-function in convex analysis, we have

$$\partial \max(x) = \operatorname{conv}\{e_i \mid i \in \mathcal{I}(x)\},$$

where e_i's are the standard bases of $\mathbb{R}^n$ and $\mathcal{I}(x) = \{i \mid x_i = \max(x)\}$.

Next, we extend our discussion to include generalized subdifferentials of a nonsmooth function on submanifolds $\mathcal{M}$. The *Riemannian Fréchet subdifferential* and the *Riemannian limiting subdifferential of f at $x \in \mathcal{M}$* (see, e.g., [45, Definition 3.1]) are defined as

$$\begin{aligned}\hat{\partial}_{\mathcal{R}} f(x) := \{&\operatorname{grad} h(x) \mid \exists \delta > 0 \text{ such that } h \in C^1(\mathcal{B}(x,\delta)) \text{ and} \\ &f - h \text{ attains a local minimum at } x \text{ on } \mathcal{M}\},\end{aligned}$$

$$\partial_{\mathcal{R}} f(x) := \{\lim_{\ell \to \infty} v^\ell \mid v^\ell \in \hat{\partial}_{\mathcal{R}} f\left(x^\ell\right), \left(x^\ell, f\left(x^\ell\right)\right) \to (x, f(x))\}.$$

If $\mathcal{M} = \mathbb{R}^n$, the above definitions coincide with the usual Fréchet and limiting subdifferentials in $\mathbb{R}^n$. Moreover, it follows directly that, for all $x \in \mathcal{M}$, one has $\hat{\partial}_{\mathcal{R}} f(x) \subseteq \partial_{\mathcal{R}} f(x)$. According to [45, Proposition 3.2], if x is a local minimizer of f on $\mathcal{M}$, then $0 \in \hat{\partial}_{\mathcal{R}} f(x)$. Thus, we call a point $x \in \mathcal{M}$ a *limiting stationary point of (NRO)* if $0 \in \partial_{\mathcal{R}} f(x)$. In this paper, we will treat it as a necessary condition for a local solution of (NRO) to exist.

The smoothing function is the most important tool of the smoothing method.

Definition 3.1 (Zhang and Chen [44, Definition 3.1]). *A function $\tilde{f}(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}_{++} \to \mathbb{R}$ is called a* smoothing function *of $f : \mathbb{R}^n \to \mathbb{R}$, if $\tilde{f}(\cdot, \mu)$ is continuously differentiable in $\mathbb{R}^n$ for any $\mu > 0$,*

$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) = f(x)$$

and there exist a constant $\kappa > 0$ and a function $\omega : \mathbb{R}_{++} \to \mathbb{R}_{++}$ such that

$$|\tilde{f}(x,\mu) - f(x)| \le \kappa \omega(\mu) \quad \text{with} \quad \lim_{\mu \downarrow 0} \omega(\mu) = 0.$$

Example 2 (Chen, Wets and Zhang [20, Lemma 4.4]). The LogSumExp function, $\operatorname{lse}(x,\mu) : \mathbb{R}^n \times \mathbb{R}_{++} \to \mathbb{R}$, given by

$$\operatorname{lse}(x,\mu) = \mu \log(\textstyle\sum_{i=1}^n \exp(x_i/\mu)),$$

is the smoothing function of $\max(x)$ because we can see that

(i) $\operatorname{lse}(\cdot,\mu)$ is smooth on $\mathbb{R}^n$ for any $\mu > 0$. Its gradient $\nabla_x \operatorname{lse}(x,\mu)$ is given by $\sigma(\cdot,\mu) : \mathbb{R}^n \to \Delta^{n-1}$,

$$\nabla_x \operatorname{lse}(x,\mu) = \sigma(x,\mu) := \frac{1}{\sum_{\ell=1}^n \exp(x_\ell/\mu)} [\ \exp(x_1/\mu), \cdots, \exp(x_n/\mu)\ ]^\top, \tag{6}$$

where $\Delta^{n-1} := \{x \in \mathbb{R}^n \mid \sum_{i=1} x_i = 1, x_i \ge 0\}$ is the unit simplex.

(ii) For all $x \in \mathbb{R}^n$ and $\mu > 0$, we have $\max(x) < \operatorname{lse}(x,\mu) \le \max(x) + \mu \log(n)$. Then, the constant $\kappa = \log(n)$ and $\omega(\mu) = \mu$. The above inequalities imply that $\lim_{z \to x, \mu \downarrow 0} \operatorname{lse}(z,\mu) = \max(x)$.

Gradient sub-consistency or consistency are crucial to showing that any limit point of the Riemannian smoothing method is also a limiting stationary point of (NRO).

Definition 3.2 (Zhang and Chen [45, Definition 3.4 & 3.9]). *A smoothing function $\tilde{f}$ of f is said to satisfy* gradient sub-consistency on $\mathbb{R}^n$ *if, for any $x \in \mathbb{R}^n$,*

$$G_{\tilde{f}}(x) \subseteq \partial f(x), \tag{7}$$

where the subdifferential of f associated with $\tilde{f}$ at $x \in \mathbb{R}^n$ *is given by*

$$G_{\tilde{f}}(x) := \{u \in \mathbb{R}^n \mid \nabla_x \tilde{f}\,(z_k, \mu_k) \to u \text{ for some } z_k \to x, \mu_k \downarrow 0\}.$$

Similarly, $\tilde{f}$ is said to satisfy Riemannian gradient sub-consistency on $\mathcal{M}$ *if, for any $x \in \mathcal{M}$,*

$$G_{\tilde{f},\mathcal{R}}(x) \subseteq \partial_{\mathcal{R}} f(x), \tag{8}$$

where the Riemannian subdifferential of f associated with $\tilde{f}$ at $x \in \mathcal{M}$ *is given by*

$$G_{\tilde{f},\mathcal{R}}(x) = \{v \in \mathbb{R}^n \mid \operatorname{grad} \tilde{f}\,(z_k, \mu_k) \to v \text{ for some } z_k \in \mathcal{M}, z_k \to x, \mu_k \downarrow 0\}.$$

If one substitutes the inclusion with equality in (7), then $\tilde{f}$ satisfies *gradient consistency* on $\mathbb{R}^n$, and similarly in (8) for $\mathcal{M}$. Thanks to the following useful proposition from [45], we can induce gradient sub-consistency on $\mathcal{M}$ from that on $\mathbb{R}^n$ if f is locally Lipschitz.

Proposition 3.3 (Zhang and Chen [45, Proposition 3.10])**.** *Let f be a locally Lipschitz function and $\tilde{f}$ a smoothing function of f. For $\tilde{f}$, if gradient sub-consistency holds on $\mathbb{R}^n$, then Riemannian gradient sub-consistency holds on $\mathcal{M}$ as well.*

The next example illustrates Riemannian gradient sub-consistency on $\mathcal{M}$ for $\mathrm{lse}(x,\mu)$ in Example 2, since any convex function is locally Lipschitz continuous.

Example 3 (Chen, Wets and Zhang [20, Lemma 4.4])**.** The smoothing function $\mathrm{lse}(x,\mu)$ of $\max(x)$ satisfies gradient consistency on $\mathbb{R}^n$. That is, for any $x \in \mathbb{R}^n$,

$$\partial \max(x) = G_{\mathrm{lse}}(x) := \{\lim_{x^k \to x, \mu_k \downarrow 0} \sigma(x^k, \mu_k)\}.$$

Note that the original assertion of [20, Lemma 4.4] is gradient consistency in the Clarke sense, i.e., $\partial^\circ \max(x) = G_{\mathrm{lse}}(x)$.

3.3 Riemannian smoothing method

Motivated by the previous papers [17, 34, 45] on smoothing methods and Riemannian manifolds, we propose a general Riemannian smoothing method. Algorithm 1 is the basic framework of this general method.

Algorithm 1: Basic Riemannian smoothing method for (NRO)

Initialization: Given $\theta_\mu \in (0,1), \mu_0 > 0$, and $x_{-1} \in \mathcal{M}$, select a smoothing function $\tilde{f}$ and a Riemannian algorithm (saying sub-algorithm) for (SRO).
for $k = 0, 1, 2, \ldots$
  Solve

$$x_k = \arg\min_{x \in \mathcal{M}} \tilde{f}(x, \mu_k), \tag{9}$$

  approximately by using the chosen sub-algorithm, starting at x^{k-1};
  if final convergence test satisfied
    stop with approximate solution x_k;
  end if
  Set $\mu_{k+1} = \theta_\mu \mu_k$;
end for

Now let us describe the convergence properties of the basic method. First, assume that the function $\tilde{f}(x, \mu_k)$ has a minimizer on $\mathcal{M}$ for each value of μ_k.

Theorem 3.4. *Suppose that each x^k is an exact global minimizer of (9) in Algorithm 1. Then every limit point x^* of the sequence $\{x^k\}$ is a global minimizer of the problem (NRO).*

Proof. See Appendix A for the proof. □

This strong result requires us to find a global minimizer of each subproblem, which, however, cannot always be done. The next result concerns the convergence properties of the sequence $\tilde{f}(x^k, \mu_k)$ under the condition that $\tilde{f}$ has the following property:

$$0 < \mu_2 < \mu_1 \Longrightarrow \tilde{f}(x, \mu_2) < \tilde{f}(x, \mu_1) \text{ for all } x \in \mathbb{R}^n. \tag{10}$$

Example 4. The above property holds for $\mathrm{lse}(x,\mu)$ in Example 2; i.e., we have $\mathrm{lse}(x,\mu_2) < \mathrm{lse}(x,\mu_1)$ on $\mathbb{R}^n$, provided that $0 < \mu_2 < \mu_1$. See Appendix B for the proof.

In [17], the authors considered a special case of Theorem 3.5, wherein the smoothing function $\tilde{f}(x,\mu) = \sqrt{\mu^2 + x^2}$ of $|x|$ also satisfies (10) and a Riemannian conjugate gradient method is used for (9).

Theorem 3.5. *Suppose that $f^* := \inf_{x\in\mathcal{M}} f(x)$ exists and the smoothing function $\tilde{f}$ has property (10). Let $f^k := \tilde{f}(x^k, \mu_k)$. Then, the sequence $\{f^k\}$ generated by Algorithm 1 is strictly decreasing and bounded below by f^*; hence,*

$$\lim_{k\to\infty} |f^k - f^{k-1}| = 0.$$

Proof. See Appendix C for the proof. □

Note that the above weak result does not ensure that $\{f^k\} \to f^*$. Next, for better convergence (compared with Theorem 3.5) and an effortless implementation (compared with Theorem 3.4), we propose an enhanced Riemannian smoothing method: Algorithm 2. This is closer to the version in [45], where the authors use the Riemannian steepest descent method for solving the smoothed problem (11).

Algorithm 2: Enhanced Riemannian smoothing method for (NRO)

Initialization: Given $\theta_\mu \in (0,1), \mu_0 > 0$ and a nonnegative sequence $\{\delta_k\}$ with $\delta_k \to 0$, and $x_{-1} \in \mathcal{M}$, select a smoothing function $\tilde{f}$ and a Riemannian algorithm (saying sub-algorithm) for (SRO).

for $k = 0, 1, 2, \ldots$

Solve

$$x_k = \arg\min_{x\in\mathcal{M}} \tilde{f}(x, \mu_k), \tag{11}$$

approximately by using the chosen sub-algorithm, starting at x^{k-1}, such that

$$\|\operatorname{grad} \tilde{f}(x^k, \mu_k)\| < \delta_k; \tag{12}$$

if final convergence test satisfied

stop with approximate solution x_k;

end if

Set $\mu_{k+1} = \theta_\mu \mu_k$;

end for

Our enhancement is simple but meaningful. Roughly speaking, it allows us to use any standard method of (SRO), not just steepest descent, to solve the smoothed problem (11). Various standard Riemannian algorithms for (SRO), such as the Riemannian conjugate gradient method [38] (which often performs better than Riemannian steepest descent), the Riemannian Newton method [2, Chapter 6], and the Riemannian trust region method [2, Chapter 7], have extended the concepts and techniques used in Euclidean space to Riemannian manifolds. As shown by Theorem 3.6, no matter what kind of sub-algorithm is implemented for (11), it does not affect the final convergence as long as the chosen sub-algorithm has the basic convergence property.

Another advantage is that we can directly use the existing solver, e.g., Manopt [14], which includes the standard Riemannian algorithms mentioned above. Hence, we can choose the most suitable sub-algorithm for the problem and quickly implement it with minimal effort, even without knowledge about "retraction" (a necessary, but not easy to understand concept in Riemannian optimization).

The following result is adapted from [45, Proposition 4.2 & Theorem 4.3]. Readers are encouraged to refer to [45] for a discussion on the stationary point of associated with $\tilde{f}$ on $\mathcal{M}$.

Theorem 3.6. *Suppose that the chosen sub-algorithm is convergent,*

$$\liminf_{\ell\to\infty} \|\operatorname{grad} g(x^\ell)\| = 0, \tag{13}$$

for (SRO), and for all μ_k, the function $\tilde{f}(\cdot, \mu_k)$ satisfies the convergence assumptions of the sub-algorithm needed for g above and $\tilde{f}$ satisfies the Riemannian gradient sub-consistency on $\mathcal{M}$. Then,

1. *For each k, there exists an x^k satisfying (13); hence, Algorithm 2 is well defined.*
2. *Every limit point x^* of the sequence $\{x^k\}$ generated by Algorithm 2 is a Riemannian limiting stationary point of (NRO).*

Proof. Fix any μ_k. By (13), we have $\liminf_{\ell\to\infty} \|\operatorname{grad} \tilde{f}(x^\ell, \mu_k)\| = 0$. Hence, there is a convergent subsequence of $\|\operatorname{grad} \tilde{f}(x^\ell, \mu_k)\|$ whose limit is 0. This means that, for any $\epsilon > 0$, there exists an integer ℓ_ϵ such that $\|\operatorname{grad} \tilde{f}(x^{\ell_\epsilon}, \mu_k)\| < \epsilon$. If $\epsilon = \delta_k$, we get $x^k = x^{\ell_\epsilon}$. Thus, statement (1) holds.

Next, suppose that x^* is a limit point of $\{x^k\}$ generated by Algorithm 2, so that there is an infinite subsequence $\mathcal{K}$ such that $\lim_{k\in\mathcal{K}} x^k = x^*$. From (1), we have

$$\lim_{k\in\mathcal{K}} \|\operatorname{grad} \tilde{f}(x^k, \mu_k)\| \le \lim_{k\in\mathcal{K}} \delta_k = 0,$$

and we find that $\operatorname{grad} \tilde{f}(x^k, \mu_k) \to 0$ for $k \in \mathcal{K}, x^k \in \mathcal{M}, x^k \to x^*, \mu_k \downarrow 0$. Hence,

$$0 \in G_{\tilde{f},\mathcal{R}}(x^*) \subseteq \partial_{\mathcal{R}} f(x^*).$$

□

Now let us consider the selection strategy of the nonnegative sequence $\{\delta_k\}$ with $\delta_k \to 0$. In [45], when $\mu_{k+1} = \theta_\mu \mu_k$ shrinks, the authors set $\delta_{k+1} := \theta_\delta \delta_k$ with an initial value of δ_0 and constant $\theta_\delta \in (0, 1)$. In the spirit of the usual smoothing methods as described in [19], one can set $\delta_k := \gamma\mu_k$ with a constant $\gamma > 0$. The latter is an adaptive rule, because μ_k determines subproblem (11) and its stopping criterion at the same time. The merits and drawbacks of the two rules require more discussion, but the latter seems to be more reasonable.

4 Numerical experiments

We conducted numerical experiments in which we solved (OptCP) in the framework of Algorithm 2, where different Riemannian algorithms are employed as sub-algorithms and $\operatorname{lse}(-\bar{B}X, \mu)$ is used as the smoothing function.

We have shown that $\operatorname{lse}(x, \mu)$ is a smoothing function of $\max(x)$ with gradient consistency. $\operatorname{lse}(\cdot, \mu)$ of the matrix argument can be simply derived from entrywise operations. Then, from the properties of compositions of smoothing functions [5, Proposition 1 (3)], we have that $\operatorname{lse}(-\bar{B}X, \mu)$ is a smoothing function of $\max(-\bar{B}X)$ with gradient consistency.

In practice, it is important to avoid numerical overflow and underflow when evaluating $\operatorname{lse}(x, \mu)$. Overflow occurs when any x_i is large and underflow occurs when all x_i are small. To avoid these problems, we can shift each component x_i by $\max(x)$. by using the following formula:

$$\operatorname{lse}(x, \mu) = \mu \log(1 + \textstyle\sum_{i\neq j}^{n} \exp((x_i - \max(x))/\mu)) + \max(x),$$

whose validity is easy to show.

In our experiments, we used three different built-in Riemannian algorithms of Manopt 7.0 [14] — steepest descent (SD), conjugate gradient (CG), and trust regions (RTR), denoted by SM_SD, SM_CG and SM_RTR, respectively. We compared our algorithms with the following numerical algorithms for CP factorization that were mentioned in subsection 1.1:

- SpFeasDC_ls [18]: A difference-of-convex functions approach for solving the split feasibility problem, it can be applied to (FeasCP). All the implementation details regarding the parameters we used are the same as in the numerical experiments reported in [18, Section 6.1].
- RIPG_mod [12]: This is a projected gradient method with relaxation and inertia parameters for solving (2). As shown in [12, Section 4.2], RIPG_mod is the best among many strategies of choosing parameters.
- APM_mod [29]: A modified alternating projection method for CP factorization; it is mentioned in Section 2.3.

We followed the settings used by the authors in their papers. The numerical experiments were performed on a computer equipped with an Intel Core i7-10700 @ 2.90GHz 2.90GHz and 16GB RAM. The algorithms were implemented in MatlabR2021a. The details of the experiments are as follows.

If $A \in \mathcal{CP}_n$ was of full rank, for accuracy reasons, we obtained an initial $\bar{B}$ by using Cholesky decomposition. Otherwise, $\bar{B}$ was obtained by using spectral decomposition. Then, we extended $\bar{B}$ to r columns by column replication (5). Let $r = \mathrm{cp}(A)$ if $\mathrm{cp}(A)$ was known or r was sufficiently large. We used `RandOrthMat.m` [39] to generate a random starting point X^0 on the basis of the Gram-Schmidt process.

For our three algorithms, we set $\mu_0 = 100, \theta_\mu = 0.8$ and used an adaptive rule of $\delta_k := \gamma\mu_k$ with $\gamma = 0.5$. Except for RIPG_mod, all the algorithms terminated successfully at iteration k, where $\min(\bar{B}X^k) \geq -10^{-15}$ was attained before the maximum number of iterations (5,000) was reached. In addition, SpFeasDC_ls failed when $\bar{L}_k > 10^{10}$. Regarding RIPG_mod, it terminated successfully when $\|A - X_k X_k^\top\|^2/\|A\|^2 < 10^{-15}$ was attained before at most 10,000 iterations for $n < 100$, and before at most 50,000 iterations in all other cases.

4.1 Randomly generated instances

We examined the case of randomly generated matrices to see how the methods were affected by order n or r. The instances were generated in the same way as in [29, Section 7.7]. We computed C by setting $C_{ij} := |B_{ij}|$ for all $i, j,$, where B is a random $n \times 2n$ matrix based on the Matlab command `randn`, and we took $A = CC^\top$ to be factorized. In Table 1, we set $r = 1.5n$ and $r = 3n$ for the values $n \in \{20, 30, 40, 100, 200, 400, 600, 800\}$. For each pair of n and r, we generated 50 instances if $n \leq 100$ and 10 instances otherwise. For each instance, we initialized all the algorithms at the same random starting point X^0 and initial decomposition $\bar{B}$, except for RIPG_mod. Note that each instance A was assigned only one starting point.

Table 1 lists the average time in seconds ($\mathrm{Time_s}$) and the average number of iterations ($\mathrm{Iter_s}$) among the successful instances. For our three algorithms, $\mathrm{Iter_s}$ contains the number of iterations of the subroutine. It also lists the rounded success rate (Rate) relative to the total number of instances. Boldface highlights the two best results for each pair of n and r.

As shown in Table 1, except for APM_mod, each method had a success rate of 1 for all pairs of n and r. Our three algorithms outperformed the other methods on the large-scale matrices with $n \geq 100$. In particular, SM_CG with the conjugate-gradient sub-algorithm gave the best results.

4.2 A specifically structured instance

Let $\mathbf{e}_n$ denote the all-ones vector in $\mathbb{R}^n$ and consider the matrix [29, Example 7.1],

$$A_n = \begin{pmatrix} 0 & \mathbf{e}_{n-1}^\top \\ \mathbf{e}_{n-1} & I_{n-1} \end{pmatrix}^\top \begin{pmatrix} 0 & \mathbf{e}_{n-1}^\top \\ \mathbf{e}_{n-1} & I_{n-1} \end{pmatrix} \in \mathcal{CP}_n.$$

Theorem 2.2 shows that $A_n \in \mathrm{int}(\mathcal{CP}_n)$ for every $n \geq 2$. By construction, it is obvious that $\mathrm{cp}(A_n) = n$. We tried to factorize A_n for the values $n \in \{10, 20, 50, 75, 100, 150\}$ in Table 2. For each A_n, using $r := \mathrm{cp}(A_n) = n$ and the same initial decomposition $\bar{B}$, we tested all the algorithms on the same 50 randomly generated starting points, except for RIPG_mod. Here, each instance was assigned 50 starting points.

Table 2 lists the average time in seconds ($\mathrm{Time_s}$) and the average number of iterations ($\mathrm{Iter_s}$) for the successful starting points. It also lists the rounded successful rate (Rate) relative to the total number of starting points. Boldface highlights the two best results for each n.

We can see from Table 2 that the success rates of our three algorithms were always 1, but whereas the success rates of the other methods decreased as n increased. Likewise, SM_CG with the conjugate-gradient sub-algorithm gave the best results.

Table 1. CP factorization of random completely positive matrices.

Methods	SM_SD			SM_CG			SM_RTR			SpFeasDC_ls			RIPG_mod			APM_mod		
$n(r=1.5n)$	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s
20	1	0.0409	49	1	0.0394	41	1	0.0514	35	1	**0.0027**	24	1	**0.0081**	1229	0.32	0.3502	2318
30	1	0.0549	58	1	0.0477	44	1	0.0690	36	1	**0.0075**	24	1	**0.0231**	1481	0.04	1.0075	2467
40	1	0.0735	65	1	0.0606	46	1	0.0859	37	1	**0.0216**	46	1	**0.0574**	1990	0	-	-
100	1	0.2312	104	1	**0.1520**	56	1	0.4061	45	1	**0.2831**	109	1	0.8169	4912	0	-	-
200	1	**1.0723**	167	1	**0.5485**	69	1	1.9855	53	1	2.2504	212	1	5.2908	9616	0	-	-
400	1	**14.6**	314	1	**4.1453**	86	1	22.1	69	1	36.9	636	1	90.6	17987	0	-	-
600	1	50.6	474	1	**14.7**	105	1	**46.4**	80	1	140.1	882	1	344.7	26146	0	-	-
800	1	133.3	643	1	**30.0**	109	1	**93.5**	89	1	413.3	1225	1	891.1	34022	0	-	-
Methods	SM_SD			SM_CG			SM_RTR			SpFeasDC_ls			RIPG_mod			APM_mod		
$n(r=3n)$	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s
20	1	0.0597	52	1	0.0551	42	1	0.0842	37	1	**0.0057**	15	1	**0.0105**	1062	0.30	0.7267	2198
30	1	0.0793	59	1	0.0673	45	1	0.1161	39	1	**0.0128**	17	1	**0.0336**	1127	0	-	-
40	1	0.1035	67	1	0.0882	48	1	0.1961	41	1	**0.0256**	19	1	**0.0822**	1460	0	-	-
100	1	**0.5632**	103	1	**0.3395**	57	1	1.4128	50	1	0.8115	86	1	1.1909	4753	0	-	-
200	1	**4.5548**	163	1	**2.4116**	68	1	14.3	65	1	8.1517	184	1	9.2248	9402	0	-	-
400	1	**46.5**	296	1	**19.2**	89	1	77.1	80	1	124.3	453	1	156.6	17563	0	-	-
600	1	**209.0**	446	1	**65.7**	99	1	294.5	76	1	981.8	795	1	616.7	25336	0	-	-
800	1	**609.7**	628	1	**160.4**	114	1	650.7	83	1	4027.4	1070	1	1289.4	26820	0	-	-

Table 2. CP factorization of a family of specifically structured instances.

Methods	SM_SD			SM_CG			SM_RTR			SpFeasDC_ls			RIPG_mod			APM_mod		
$n(r=n)$	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s
10	1	0.0399	71	1	0.0313	49	1	0.0424	45	1	**0.0043**	149	1	**0.0074**	2085	0.80	0.0174	616
20	1	0.0486	85	1	0.0408	63	1	0.0637	55	0.98	**0.0139**	201	0.74	**0.0212**	3478	0.90	0.0591	864
50	1	0.2599	295	1	**0.1073**	101	1	**0.2104**	76	0.98	0.3389	770	0	-	-	0.76	0.6948	1416
75	1	**0.3843**	329	1	**0.1923**	135	1	0.4293	93	0.98	1.0706	1186	0	-	-	0.64	1.4809	1510
100	1	**0.7459**	458	1	**0.3289**	168	1	0.9074	108	0.80	1.6653	1083	0	-	-	0.60	2.8150	1690
150	1	**1.8076**	647	1	**0.7837**	241	1	2.6030	145	0.70	3.7652	1170	0	-	-	0.35	9.9930	2959

4.3 An easy instance on the boundary of $\mathcal{CP}_n$

Consider the following matrix from [41, Example 2.7]:

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

The sufficient condition from [41, Theorem 2.5] ensures that this matrix is completely positive and $\mathrm{cp}(A) = \mathrm{rank}(A) = 3$. Theorem 2.2 tells us that $A \in \mathrm{bd}(\mathcal{CP}_5)$, since $\mathrm{rank}(A) \neq 5$.

We found that all the algorithms could easily factorize this matrix. However, our three algorithms could return a CP factorization B whose smallest entry was as large as possible. In fact, our methods also maximized the smallest entry in the $n \times r$ symmetric factorization of A, since (OptCP) is equivalent to

$$\max_{A=XX^\top, X\in\mathbb{R}^{n\times r}} \{\min(X)\}.$$

When we did not terminate as soon as $\min(\bar{B}X^k) \geq -10^{-15}$, for example, after 1000 iterations, our algorithms gave the following CP factorization whose the smallest entry is around $2.8573 \gg -10^{-15}$:

$$A = BB^\top \text{ , where } B \approx \begin{pmatrix} 3.5771 & 4.4766 & \mathbf{2.8573} \\ 2.8574 & 3.0682 & 6.6650 \\ 8.3822 & 7.0001 & 6.5374 \\ 5.7515 & 2.8574 & 7.9219 \\ 2.8574 & 6.7741 & 3.3085 \end{pmatrix}.$$

4.4 A hard instance on the boundary of $\mathcal{CP}_n$

Next, we examined how well these methods worked on a hard matrix on the boundary of $\mathcal{CP}_n$. Consider the following matrix on the boundary taken from [28]:

$$A = \begin{pmatrix} 8 & 5 & 1 & 1 & 5 \\ 5 & 8 & 5 & 1 & 1 \\ 1 & 5 & 8 & 5 & 1 \\ 1 & 1 & 5 & 8 & 5 \\ 5 & 1 & 1 & 5 & 8 \end{pmatrix} \in \mathrm{bd}(\mathcal{CP}_5).$$

Since A is of full rank, then by Theorem 2.2 $\mathrm{cp}^+(A) = \infty$; i.e., there is no strictly positive CP factorization for A. Hence, the global minimum of (OptCP), $t = 0$ is clear. None of the algorithms could decompose this matrix under our tolerance, 10^{-15}, in the stopping criteria. As was done in [29, Example 7.3], we investigated slight perturbations of this matrix. Given

$$MM^\top =: C \in \mathrm{int}(\mathcal{CP}_5) \text{ with } M = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

we factorized $A_\lambda := \lambda A + (1-\lambda)C$ for different values of $\lambda \in [0, 1)$ using $r = 12 > \mathrm{cp}_5 = 11$. Note that $A_\lambda \in \mathrm{int}(C\mathcal{P}_5)$ provided $0 \leq \lambda < 1$ and A_λ approached the boundary as $\lambda \to 1$. We chose the largest $\lambda = 0.99$. For each A_λ, we tested all the algorithms on 50 randomly generated starting points and computed the success rate relative to the total number of starting points.

Table 3 shows how the success rate of each algorithm changes as A_λ approaches the boundary. The table sorts the results from left to right according to overall performance. Except for SM_RTR whose success rate was always 1, the success rates of all the other algorithms significantly decreased as λ increased to 0.9999. Surprisingly, the method of SM_CG, which performed well in the previous example, seemed unable to handle instances close to the boundary.

Table 3. Success rate of CP factorization of A_λ for values of λ from 0.6 to 0.9999.

λ	SM_RTR	SM_SD	RIPG_mod	SM_CG	SpFeasDC_ls	APM_mod
0.6	1	1	1	1	1	0.42
0.65	1	1	1	1	1	0.44
0.7	1	1	1	1	1	0.48
0.75	1	1	1	1	1	0.52
0.8	1	1	1	1	0.96	0.46
0.82	1	1	1	1	0.98	0.4
0.84	1	1	1	1	0.86	0.24
0.86	1	1	1	1	0.82	0.1
0.88	1	1	1	1	0.58	0.18
0.9	1	1	1	1	0.48	0.18
0.91	1	1	1	1	0.4	0.14
0.92	1	1	1	1	0.2	0.18
0.93	1	1	0.98	1	0.22	0.22
0.94	1	1	0.98	1	0.1	0.2
0.95	1	1	1	1	0.12	0.32
0.96	1	1	0.96	0.98	0.06	0.34
0.97	1	1	0.86	0.82	0.06	0.14
0.98	1	1	0.76	0.28	0.02	0
0.99	1	0.68	0.42	0	0	0
0.999	1	0	0.14	0	0	0
0.9999	1	0	0	0	0	0

5 Concluding remarks

We examined the problem of finding a CP factorization of a given completely positive matrix and treated it as a nonsmooth Riemannian optimization. To the best of our knowledge, ours is the first study on a general smoothing framework for Riemannian optimization. The numerical experiments clarify that our method can compete with other efficient CP factorization methods, in particular on large-scale matrices.

As a future work, we plan to extend Algorithm 2 to the case of general manifolds, particularly, to quotient manifolds. This application is believed to be possible, although effort should be put into the analogous convergence results. In fact, it has been verified in practice; in a built-in example in Manopt 7.0 [14], `robust_pca.m` computes a robust version of PCA on data and optimizes a nonsmooth function over a Grassmann manifold. The nonsmooth term consists of the l_2 norm which is not squared, for robustness. In `robust_pca.m`, Riemannian smoothing with a pseudo-Huber loss function has been applied to replace the l_2 norm.

Just like other numerical methods, there is no guarantee that Algorithm 2 will successfully find a CP factorization for *every* $A \in \mathcal{CP}^n$. It follows from Proposition 2.7 that $A \in \mathcal{CP}_n$ if and only if the global minimum of (OptCP), say t, is such that $t \leqslant 0$. Since our methods only converge to a stationary point, Algorithm 2 provides us with a local minimizer at best. We are looking forward to finding a global minimizer of (OptCP) in our future work.

Acknowledgments

This work was supported by the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center, at Kyoto University, JSPS KAKENHI Grant Number (B)19H02373, and by JST SPRING Grant Number JPMJSP2124.

A Proof of Theorem 3.4

Proof. Let $\bar{x}$ be a global solution of (NRO), that is,

$$f(\bar{x}) \leq f(x) \quad \text{for all } x \in \mathcal{M}.$$

From the definition 3.1 of the smoothing function, there exist a constant $\kappa > 0$ and a function $\omega : \mathbb{R}_{++} \to \mathbb{R}_{++}$ such that for all $x \in \mathcal{M}$

$$-\kappa\omega(\mu) \leq \tilde{f}(x,\mu) - f(x) \leq \kappa\omega(\mu) \tag{14}$$

with $\lim_{\mu\downarrow 0} \omega(\mu) = 0$. Substituting x^k, and combining with the global solution $\bar{x}$, we have that

$$\tilde{f}(x^k,\mu_k) \geq f(x^k) - \kappa\omega(\mu_k) \geq f(\bar{x}) - \kappa\omega(\mu_k).$$

By rearranging this expression, we obtain

$$-\kappa\omega(\mu_k) \leq \tilde{f}(x^k,\mu_k) - f(\bar{x}). \tag{15}$$

Since x^k minimizes $\tilde{f}(x,\mu_k)$ on $\mathcal{M}$ for each μ_k, we have that $\tilde{f}(x^k,\mu_k) \leq \tilde{f}(\bar{x},\mu_k)$, which leads to

$$\tilde{f}(x^k,\mu_k) - f(\bar{x}) \leq \tilde{f}(\bar{x},\mu_k) - f(\bar{x}) \leq \kappa\omega(\mu_k). \tag{16}$$

The second inequality above follows (14). Combining (15) and (16), we obtain

$$|\tilde{f}(x^k,\mu_k) - f(\bar{x})| \leq \kappa\omega(\mu_k). \tag{17}$$

Suppose that x^* is a limit point of $\{x^k\}$, so that there is an infinite subsequence $\mathcal{K}$ such that $\lim_{k\in\mathcal{K}} x^k = x^*$. Note that $x^* \in \mathcal{M}$ because $\mathcal{M}$ is complete. By taking the limit as $k \to \infty, k \in \mathcal{K}$, on both sides of (17), again by definition of the smoothing function, we obtain

$$|f(x^*) - f(\bar{x})| = \lim_{k\in\mathcal{K}} |\tilde{f}(x^k,\mu_k) - f(\bar{x})| \leq \lim_{k\in\mathcal{K}} \kappa\omega(\mu_k) = 0.$$

Thus, it follows that $f(x^*) = f(\bar{x})$.

Since $x^* \in \mathcal{M}$ is a point whose objective value is equal to that of the global solution $\bar{x}$, we conclude that x^*, too, is a global solution. □

B Proof of Example 4

Proof. Note that under the equality,

$$\sum_{l=1}^{n} \exp(x_l/\mu) = \exp\{\mathrm{lse}(x,\mu)/\mu\},$$

the i-th component of $\sigma(x,\mu)$ can be rewritten as

$$\sigma_i(x,\mu) = \exp\{(x_i - \mathrm{lse}(x,\mu))/\mu\}.$$

For any fixed $x \in \mathbb{R}^n$, consider the derivative of a real function $\mu \to \mathrm{lse}(x,\cdot) : \mathbb{R}_{++} \to \mathbb{R}$. Then, we have

$$\begin{aligned}\nabla_\mu \mathrm{lse}(x,\mu) = \mathrm{lse}\,/\mu - \frac{\sum_{i=1}^n x_i \exp(x_i/\mu)}{\mu \exp\left(\mathrm{lse}\,/\mu\right)} &= (\mathrm{lse} - \textstyle\sum_{i=1}^n x_i \exp\{(x_i - \mathrm{lse})/\mu\})/\mu \\ &= (\mathrm{lse} - \textstyle\sum_{i=1}^n x_i\sigma_i)/\mu \leq 0,\end{aligned}$$

where "lse, σ" are shorthand for $\mathrm{lse}(x,\mu)$ and $\sigma(x,\mu)$. For the last inequality above, we observe that $\sigma \in \Delta^{n-1}$; hence, the term $\sum_{i=1}^n x_i\sigma_i$ is a convex combination of all entries of x, which implies that $\sum_{i=1}^n x_i\sigma_i \leq \max(x) < \mathrm{lse}\,$. This completes our proof. □

C Proof of Theorem 3.5

Proof. For each $k \geq 1$, x_k is obtained by approximately solving

$$\min_{x \in \mathcal{M}} \tilde{f}(x, \mu_k),$$

starting at x^{k-1}. Then, at least, we have

$$\tilde{f}(x^{k-1}, \mu_k) \geq \tilde{f}(x^k, \mu_k) = f^k.$$

Since $\mu_k = \theta_\mu \mu_{k-1} < \mu_{k-1}$, property (10) ensures

$$f^{k-1} = \tilde{f}(x^{k-1}, \mu_{k-1}) > \tilde{f}(x^{k-1}, \mu_k).$$

The claim that sequence $\{f^k\}$ is strictly decreasing follows from these two inequalities.

Suppose that, for all $\mu > 0$ and for all $x \in \mathbb{R}^n$,

$$\tilde{f}(x, \mu) \geq f(x). \tag{18}$$

Then, for each k,

$$f^k = \tilde{f}(x^k, \mu_k) \geq f(x^k) \geq \inf_{x \in \mathcal{M}} f(x) = f^*,$$

which proves our claims.

Now, we show (18) is true if the smoothing function has property (10). Fix any $x \in \mathbb{R}^n$; (10) implies that $\tilde{f}(x, \cdot)$ is strictly decreasing as $\mu \to 0$. Actually, $\tilde{f}(x, \cdot)$ is monotonically increasing on $\mu > 0$. On the other hand, from the definition of the smoothing function we have that

$$\lim_{\mu \downarrow 0} \tilde{f}(x, \mu) = f(x).$$

Hence, we have

$$\inf_{\mu > 0} \tilde{f}(x, \mu) = f(x),$$

as claimed. □

References

[1] B. Abraham and S. M. Naomi, *Completely Positive Matrices*, World Scientific, Singapore, 2003.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, 2009.

[3] A. Bagirov, N. Karmitsa, and M. M. Mäkelä, *Introduction to Nonsmooth Optimization: theory, practice and software*, Springer International Publishing, Berlin, 2014.

[4] A. Berman, M. Dür, and N. Shaked-Monderer, *Open problems in the theory of completely positive and copositive matrices*, Electronic Journal of Linear Algebra, 29 (2015), pp. 46–58.

[5] W. Bian and X. Chen, *Neural network for nonsmooth, nonconvex constrained minimization via smooth approximation*, IEEE Transactions on Neural Networks and Learning Systems, 25 (2013), pp. 545–556.

[6] I. M. Bomze, *Copositive optimization–recent developments and applications*, European Journal of Operational Research, 216 (2012), pp. 509–520.

[7] ——, *Building a completely positive factorization*, Central European Journal of Operations research, 26 (2018), pp. 287–305.

[8] I. M. Bomze, P. J. Dickinson, and G. Still, *The structure of completely positive matrices according to their cp-rank and cp-plus-rank*, Linear Algebra and Its Applications, 482 (2015), pp. 191–206.

[9] I. M. Bomze, M. Dür, E. De Klerk, C. Roos, A. J. Quist, and T. Terlaky, *On copositive programming and standard quadratic optimization problems*, Journal of Global Optimization, 18 (2000), pp. 301–320.

[10] I. M. Bomze, W. Schachinger, and G. Uchida, *Think co (mpletely) positive! matrix properties, examples and a clustered bibliography on copositive optimization*, Journal of Global Optimization, 52 (2012), pp. 423–445.

[11] P. B. Borckmans, S. E. Selvan, N. Boumal, and P.-A. Absil, *A riemannian subgradient algorithm for economic dispatch with valve-point effect*, Journal of Computational and Applied Mathematics, 255 (2014), pp. 848–866.

[12] R. I. Boţ and D.-K. Nguyen, *Factorization of completely positive matrices using iterative projected gradient steps*, Numerical Linear Algebra with Applications, (2021), p. e2391.

[13] N. Boumal, *An introduction to optimization on smooth manifolds*, Available online, Aug, (2020).

[14] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, *Manopt, a Matlab toolbox for optimization on manifolds*, Journal of Machine Learning Research, 15 (2014), pp. 1455–1459.

[15] S. Burer, *On the copositive representation of binary and continuous nonconvex quadratic programs*, Mathematical Programming, 120 (2009), pp. 479–495.

[16] ——, *A gentle, geometric introduction to copositive optimization*, Mathematical Programming, 151 (2015), pp. 89–116.

[17] L. Cambier and P.-A. Absil, *Robust low-rank matrix completion by riemannian optimization*, SIAM Journal on Scientific Computing, 38 (2016), pp. S440–S460.

[18] C. Chen, T. K. Pong, L. Tan, and L. Zeng, *A difference-of-convex approach for split feasibility with applications to matrix factorizations and outlier detection*, Journal of Global Optimization, (2020), pp. 1–30.

[19] X. Chen, *Smoothing methods for nonsmooth, nonconvex minimization*, Mathematical Programming, 134 (2012), pp. 71–99.

[20] X. Chen, R. J.-B. Wets, and Y. Zhang, *Stochastic variational inequalities: residual minimization smoothing sample average approximations*, SIAM Journal on Optimization, 22 (2012), pp. 649–673.

[21] G. de Carvalho Bento, J. X. da Cruz Neto, and P. R. Oliveira, *A new approach to the proximal point method: convergence on general riemannian manifolds*, Journal of Optimization Theory and Applications, 168 (2016), pp. 743–755.

[22] E. De Klerk and D. V. Pasechnik, *Approximation of the stability number of a graph via copositive programming*, SIAM Journal on Optimization, 12 (2002), pp. 875–892.

[23] P. J. Dickinson, *An improved characterisation of the interior of the completely positive cone*, Electronic Journal of Linear Algebra, 20 (2010), pp. 723–729.

[24] ——, *The Copositive Cone, the Completely Positive Cone and Their Generalisations*, PhD thesis, University of Groningen, 2013.

[25] P. J. Dickinson and M. Dür, *Linear-time complete positivity detection and decomposition of sparse matrices*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 701–720.

[26] P. J. Dickinson and L. Gijben, *On the computational complexity of membership problems for the completely positive cone and its dual*, Computational Optimization and Applications, 57 (2014), pp. 403–415.

[27] M. Dür, *Copositive programming–a survey*, in Recent advances in optimization and its applications in engineering, Springer, Berlin, 2010, pp. 3–20.

[28] M. Dür and G. Still, *Interior points of the completely positive cone*, The Electronic Journal of Linear Algebra, 17 (2008).

[29] P. Groetzner and M. Dür, *A factorization method for completely positive matrices*, Linear Algebra and Its Applications, 591 (2020), pp. 1–24.

[30] P. H. Groetzner, *A Method for Completely Positive and Nonnegative Matrix Factorization*, PhD thesis, University of Trier, 2018.

[31] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge University Press, Cambridge, 2012.

[32] F. Jarre and K. Schmallowsky, *On the computation of C^* certificates*, Journal of Global Optimization, 45 (2009), p. 281.

[33] A. Kovnatsky, K. Glashoff, and M. M. Bronstein, *MADMM: a generic algorithm for non-smooth optimization on manifolds*, in European Conference on Computer Vision, Springer, 2016, pp. 680–696.

[34] C. Liu and N. Boumal, *Simple algorithms for optimization on riemannian manifolds with constraints*, Applied Mathematics & Optimization, 82 (2020), pp. 949–981.

[35] J. Nie, *The $\mathcal{A}$-truncated K-moment problem*, Foundations of Computational Mathematics, 14 (2014), pp. 1243–1276.

[36] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, vol. 317, Springer Science & Business Media, Berlin, 2009.

[37] W. Rudin et al., *Principles of mathematical analysis*, McGraw-hill, New York, 1964.

[38] H. Sato and T. Iwai, *A new, globally convergent riemannian conjugate gradient method*, Optimization, 64 (2015), pp. 1011–1031.

[39] O. Shilon, *Randorthmat*, 2020.

[40] M. D. Sikirić, A. Schürmann, and F. Vallentin, *A simplex algorithm for rational cp-factorization*, Mathematical Programming, (2020), pp. 1–21.

[41] W. So and C. Xu, *A simple sufficient condition for complete positivity*, Operators and Matrices, 9 (2015), pp. 233–239.

[42] J. Sponsel and M. Dür, *Factorization and cutting planes for completely positive matrices by copositive projection*, Mathematical Programming, 143 (2014), pp. 211–229.

[43] C. Xu, *Completely positive matrices*, Linear Algebra and Its Applications, 379 (2004), pp. 319–327.

[44] C. Zhang and X. Chen, *A smoothing active set method for linearly constrained non-lipschitz nonconvex optimization*, SIAM Journal on Optimization, 30 (2020), pp. 1–30.

[45] C. Zhang, X. Chen, and S. Ma, *A riemannian smoothing steepest descent method for non-lipschitz optimization on submanifolds*, arXiv preprint arXiv:2104.04199, (2021).