An inertial Riemannian gradient ADMM for nonsmooth manifold optimization ⋆

Xiaoquan Wang [a], Hu Shao [a], Haoning Xi [b], Pengjie Liu [a]

[a] *School of Mathematics, China University of Mining and Technology, Jiangsu, China*

[b] *Newcastle Business School, The University of Newcastle, Newcastle, Australia*

Abstract

The Alternating Direction Method of Multipliers (ADMM) is widely recognized for its efficiency in solving separable optimization problems. However, its application to optimization on Riemannian manifolds remains a significant challenge. In this paper, we propose a novel inertial Riemannian gradient ADMM (iRG-ADMM) to solve Riemannian optimization problems with nonlinear constraints. Our key contributions are as follows: (i) we introduce an inertial strategy applied to the Riemannian gradient, enabling faster convergence for smooth subproblems constrained on Riemannian manifolds; (ii) for nonsmooth subproblems in Euclidean space, we incorporate existing well-established algorithms for efficient solution; and (iii) we establish the ϵ-stationarity of iRG-ADMM under mild conditions. Finally, we demonstrate the effectiveness of iRG-ADMM through extensive numerical experiments, including applications to Sparse Principal Component Analysis (SPCA), highlighting its superior performance compared to existing methods.

Key words: Riemannian manifold; Alternating direction multiplier method; Nonsmooth optimization; Inertial strategy; ϵ-stationary solution.

1 Introduction

In recent years, optimization on the manifold has been extensively studied and applied across various fields, including but not limited to automatic control [1], aerospace engineering [2], and machine learning [3]. We begin by considering the following optimization problem,

$$\begin{aligned} &\min \mathcal{A}(x) \\ &\text{s.t. } x \in \mathbb{M}, \end{aligned} \tag{1}$$

where $\mathcal{A}$ is possibly nonsmooth, and $\mathbb{M}$ is an embedded compact Riemannian submanifold in Euclidean space. This optimization problem (1) is motivated by a wide range of applications across various scientific and technological domains as a result of natural geometry and latent data simplicity, such as sparse principal component analysis [4], low-rank matrix completion [3], parse inverse covariance estimation [5], blind deconvolution [6] and dictionary learning [7]. However, solving manifold optimization presents significant challenges compared to classical optimization in the Euclidean space. In particular, when the search space is a manifold, the vectors within it do not exhibit properties such as linear combinations, which hinders the application of traditional algorithms that rely on linear structures. As a result, there has been considerable attention on developing efficient algorithms tailored specifically to manifold optimization problems. When the objective function f in (1) is smooth, a variety of algorithms based on the Riemannian gradient has been proposed, such as the Riemannian conjugate gradient method [8,9,10,11], the Riemannian trust-region method [12,13], and the Riemannian quasi-Newton method [14]. For non-smooth optimization problems on Riemannian manifolds, however, algorithm design becomes more complex. Existing literature has explored the Riemannian subgradient method [15,16], Riemannian proximal gradient method [17,18,19], Riemannian proximal-linear algorithm [20], Riemannian proximal point algorithm [21], etc.

In this paper, we focus on the following optimization problem,

$$\begin{aligned} &\min \mathcal{A}(x) + \mathcal{B}(\mathcal{C}(x)) \\ &\text{s.t. } x \in \mathbb{M}, \end{aligned} \tag{2}$$

where $\mathcal{A}$ is smooth could be nonconvex, $\mathcal{B}$ is nonsmooth

⋆ The material in this paper was not presented at any conference. Corresponding author H. Shao.

Email addresses: wxq4869@163.com (Xiaoquan Wang), shaohu@cumt.edu.cn (Hu Shao), alice.xi@newcastle.edu.au (Haoning Xi), liupengjie2019@163.com (Pengjie Liu).

and convex, $\mathbb{M}$ is an embedded compact submanifold in Euclidean space, nnd $\mathcal{C}$ is a potentially nonlinear differential mapping. The problem (2) is a generalization of (1) and highly representative in Riemannian manifolds optimization problems. Notable examples include the Rayleigh quotient problem [12,13], the Brockett cost function problem [13,11], the sparse principal component analysis (SPCA) problem [18] and the dual principal component pursuit [22], all of which can be viewed as degenerate cases of this framework. In problem (2), the nonsmooth function $\mathcal{B}$ acts on $\mathcal{C}(x)$, which introduces various inconveniences in practical problem-solving. Notably, $\mathcal{A}$ and $\mathcal{B}$ exhibit favorable separability. Therefore, a natural approach is to introduce an auxiliary variable to reformulate the problem (2) into the following separable optimization problem:

$$\begin{aligned} &\min\ \mathcal{A}(x)+\mathcal{B}(y)\\ &\text{s.t. } \mathcal{C}(x)+y=0,\ x\in\mathbb{M}. \end{aligned} \tag{3}$$

For separable optimization problems (3), the Alternating Direction Method of Multipliers (ADMM) is a widely considered and effective approach. ADMM, initially proposed by Glowinski and Marroco [23] and Gabay and Mercier [24] in the 1970s, originates from the Douglas-Rachford operator splitting method [25]. It has since emerged as a powerful tool for solving linearly constrained separable optimization problems. ADMM is particularly well-suited for machine learning applications, where very high numerical precision is often unnecessary, but low computational overhead per iteration is crucial. ADMM leverages the separability of the objective function, decomposing inherently difficult problems into two relatively simpler subproblems. To solve each subproblem, ADMM utilizes Lagrange multipliers to temporarily transform the constrained problem into an unconstrained one. This approach not only simplifies handling constraints but also addresses certain limitations inherent in penalty function methods and gradient projection methods. Due to its versatility and strong performance, ADMM and its numerous variants have garnered significant attention. For a more detailed discussion of ADMM and its applications, we refer readers to [26,27,28,29,30,31] and the references therein.

The ADMM in Euclidean spaces has seen significant development, however, the research on ADMM for Riemannian manifolds remains relatively limited. Lai and Osher [32] focused on a special case of (3) and proposed splitting orthogonality constraints (SOC) algorithm for solving such problems. The SOC method demonstrates strong performance when addressing subproblems within its framework. However, since SOC is rooted in conventional optimization frameworks involving orthogonality constraints, extending it to arbitrary Riemannian manifolds presents substantial challenges. Kovnatsky et al. [33] introduced the Manifold ADMM (MADMM) algorithm to tackle optimization problems where the objective function consists of a smooth component constrained on a Riemannian manifold and a nonsmooth component constrained in Euclidean space. In MADMM, Riemannian gradient-based methods can be used to solve the subproblem associated with the smooth component, while the nonsmooth subproblem benefits from well-established algorithms in Euclidean space, simplifying its resolution. Consequently, MADMM exhibits both strong performance and computational efficiency. Although SOC [32] and MADMM [33] demonstrate promising numerical performance, their convergence is not guaranteed. To address this limitation, Li et al. [34] introduced intermediate variables and developed a class of efficient Riemannian ADMM algorithms under the assumption that $\mathcal{C}$ is a matrix mapping. They further established the ϵ-stability of these algorithms, marking a significant advancement in this domain.

Besides, for a class of structured multi-block Riemannian manifold optimization problems, Zhang et al. [35] proposed a series of effective proximal ADMM frameworks, incorporating stochastic and gradient-based algorithms and establishing their ϵ-stability. However, the algorithms in [35] require the final subproblem to be smooth subproblem in Euclidean space, which limits their applicability to problems such as SPCA [4].

This paper introduces a novel optimization framework, inertial Riemannian gradient ADMM (iRG-ADMM), designed to solve the optimization problem (3) more efficiently and practically. By addressing key limitations of existing methods and providing strong theoretical guarantees, iRG-ADMM opens new avenues for solving manifold optimization problems with nonlinear constraints and convex, potentially nonsmooth, objective functions. The major contributions of the paper are summarized as follows,

- We propose an inertial strategy that is integrated into the ADMM framework for solving manifold optimization problems. This strategy introduces a new update direction for the primal variable x, significantly improving the convergence rate and computational efficiency. This is particularly advantageous for problems where the solution to the first subproblem strongly influences the overall algorithm performance, as in ADMM-type methods.
- iRG-ADMM is designed to be generalizable and scalable, as it does not rely on a specific manifold. Unlike existing methods such as SOC [32] (which is tailored to the Stiefel manifold) and [36] (designed for the oblique manifold), iRG-ADMM can be applied to various types of manifolds without requiring manifold-specific adjustments. This scalability enhances its applicability to a broad range of optimization problems.
- One key advantage of iRG-ADMM is that it only requires the objective function of the final subproblem to be convex, rather than smooth. This relaxed assumption opens the door to solving a wider class of real-world problems, many of which involve nonsmooth objective functions but are still convex. This enhancement significantly broadens the algorithm's practical applicability, while retaining strong theoretical guar-

antees. iRG-ADMM is capable of solving manifold optimization problems with nonlinear constraints. To the best of our knowledge, this aspect of manifold optimization has received limited attention, and our work presents a foundational approach for tackling such problems. This positions iRG-ADMM as an important starting point for future research in the domain of constrained manifold optimization.
- We establish the ϵ-stability of iRG-ADMM under common assumptions and provide a detailed analysis of its convergence properties. Using the Moreau envelope, we derive stability guarantees that ensure the algorithm's robustness. Furthermore, we identify the permissible parameter ranges for iRG-ADMM, which will assist practitioners in applying the method effectively in different settings.
- Through extensive experiments, we compare iRG-ADMM with existing manifold optimization algorithms. The results demonstrate that iRG-ADMM not only outperforms current methods in terms of computational efficiency but also maintains robustness across a variety of benchmark problems. These findings validate the practical advantages of the proposed algorithm.

The rest of this paper is organized as follows: we give mathematical notations and recall some known results for further analysis in Section 2; we introduce the iRG-ADMM and establish their ϵ-stability for problem (3) under suitable assumptions in Section 3; we present results of numerical experiments with SPCA to demonstrate the effectiveness of the propose iRG-ADMM in Section 4; and summarize the study with future directions in Section 5.

2 Notations and Preliminaries

2.1 Notations

In this subsection, we introduce some standard notations that will be used throughout this paper. First, we consider the Euclidean space, which can be interpreted as a space of vectors, matrices, or tensors. For any vectors $u, v \in \mathbb{R}^n$, we denote the Euclidean inner product by $\langle u, v\rangle = u^\top v$, and the Euclidean norm is defined as $\|u\| = \sqrt{\langle u, u\rangle}$, where the superscript "$\top$" denotes the transpose of a vector or matrix. This definition corresponds to the l_2-norm of a vector. For any matrices $U, V \in \mathbb{R}^{m\times n}$, the Euclidean inner product is defined as $\langle U, V\rangle = \mathrm{Tr}(U^\top V)$, and the Frobenius norm of U is denoted as $\|U\|_F = \sqrt{\langle U, U\rangle}$. Additionally, we define the l_2-norm of a matrix U as $\|U\| = \sqrt{\mu_{U^\top U}}$, where $\mu_{U^\top U}$ represents the largest eigenvalue of the matrix $U^\top U$. Naturally, for vectors or matrices x and y in Euclidean space, we define their distance as $\mathrm{dist}(x, y) = \|x - y\|$. Finally, for the problem (3), we introduce the associated augmented Lagrangian function (ALF) as follows:

$$\mathcal{L}_\rho(x, y, \lambda) = \mathcal{A}(x)+\mathcal{B}(y)+\langle\lambda, \mathcal{C}(x)+y\rangle+\frac{\rho}{2}\|\mathcal{C}(x)+y\|^2, \tag{4}$$

where λ is the Lagrangian multiplier and $\rho > 0$ is a penalty parameter.

2.2 Moreau envelope

Considering a function $\mathcal{B} : \mathbb{R}^p \to \mathbb{R}$, we can define its Moreau envelope [39], [40], [41] as follows:

$$M_{\mathcal{B}}(z) = \min_y\{\mathcal{B}(y) + \frac{\gamma}{2}\|y - z\|^2\},$$

where $\gamma > 0$ is a parameter. Then, its corresponding proximity operator could be defined:

$$\mathrm{Prox}_{\gamma,\mathcal{B}}(z) = \operatorname*{argmin}_y\{\mathcal{B}(y) + \frac{\gamma}{2}\|y - z\|^2\},$$

If $\mathcal{B}$ is convex and $\gamma > 0$, then $\mathrm{Prox}_{\gamma,\mathcal{B}}(\cdot)$ is monotone, single-valued and Lipschitz, and $M_{\mathcal{B}}(z)$ satisfies [41,42,43]:

$$\nabla M_{\mathcal{B}}(z) = \gamma(z - \mathrm{Prox}_{\gamma,\mathcal{B}}(z)) \in \partial\mathcal{B}(\mathrm{Prox}_{\gamma,\mathcal{B}}(z)).$$

Definition 1 [41] Let $\mathcal{B}$ be a proper, lower semicontinuous and convex function. We call $\mathcal{B}$ satisfies the implicit Lipschitz subgradient property if for any $\gamma > 0$, there exists $L_{\mathcal{B}} > 0$ (depending on γ) such that for any $u, v \in \mathrm{ran}(\mathrm{Prox}_{\gamma,\mathcal{B}})$,

$$\|\nabla M_{\mathcal{B}}(z_1) - \nabla M_{\mathcal{B}}(z_2)\| \leqslant L_{\mathcal{B}}\|u - v\|,$$
$$\forall z_1 \in \mathrm{Prox}^{-1}_{\gamma,\mathcal{B}}(u),\ z_2 \in \mathrm{Prox}^{-1}_{\gamma,\mathcal{B}}(v).$$

2.3 Preliminaries on Riemannian submanifolds in Euclidean spaces

Definition 2 Consider a Riemannian manifold $\mathbb{M}$ embedded in a Euclidean space. For any $x \in \mathbb{M}$, the tangent space $\mathcal{T}_x\mathbb{M}$ at x is a linear subspace consisting of the derivatives of all smooth curves on $\mathbb{M}$ passing x, that is

$$\mathcal{T}_x\mathbb{M} = \{\tau(0) : \tau(0) = x, \tau([-\delta, \delta]) \in \mathbb{M}, \text{for some } \delta > 0,\ \tau \text{ is smooth}\}.$$

The Riemannian metric, i.e., the inner product between $u, v \in \mathcal{T}_x\mathbb{M}$, is defined to be $\langle u, v\rangle_x := \langle u, v\rangle$, where the $\langle\cdot,\cdot\rangle$ is the Euclidean inner product.

Definition 3 [3] For smoothing funcion f on the the Riemannian manilfold $\mathbb{M}$, the Riemannian gradient $\mathbf{grad}f(x)$ is a tangent vector in $\mathcal{T}_x\mathbb{M}$ satisfying $v[f] = \langle v, \mathbf{grad}f(x)\rangle_x$ for any $v \in \mathcal{T}_x\mathbb{M}$. If $\mathbb{M}$ is an embedded submanifold of a Euclidean space, we have

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

where $\mathbf{Proj}_x(\cdot)$ is a orthogonal projection operator onto the subspace $\mathcal{T}_x\mathbb{M}$, which is a nonexpansive linear transformation.

Definition 4 [3] A retraction on $\mathbb{M}$ is a smooth map $\mathbf{Re}$: $\mathcal{T}\mathbb{M} \to \mathbb{M}$ satisfying: (i) $\mathbf{Re}_x(0_x) = x$, where 0_x denotes

the zero element of $\mathcal{T}_x\mathbb{M}$; (ii) with thecanonical identification $\mathcal{T}_{0_x}\mathcal{T}_x\mathbb{M} \simeq \mathcal{T}_x\mathbb{M}$, $\mathbf{Re}_x$ satisfies $D\mathbf{Re}_x(0_x) = id_{\mathcal{T}_x\mathbb{M}}$, where $\mathbf{Re}_x$: $\mathcal{T}_x\mathbb{M} \to \mathbb{M}$ is the restriction of $\mathbf{Re}$ at x, $D\mathbf{Re}_x$ is the differential of $D\mathbf{Re}_x$, and $id_{\mathcal{T}_x\mathbb{M}}$ denotes the identity map on $\mathcal{T}_x\mathbb{M}$.

Definition 5 [3] The function $f : \mathbb{M} \to \mathbb{R}$ is $L_{\mathbb{M}}$-smooth. There exists a constant $L_{\mathbb{M}} > 0$, for all $x \in \mathbb{M}$, $z = \mathbf{Re}_x(u)$ with $u \in \mathcal{T}_x\mathbb{M}$ such that

$$f(z) \leqslant f(x) + \langle \mathbf{grad} f(x), u\rangle_x + \frac{L_{\mathbb{M}}}{2}\|u\|^2.$$

Lemma 1 [45] *Suppose $\mathbb{M}$ is a compact and complete Riemannian manifold embedded in Euclidean space $\mathbb{R}^n$ and f is L_f-Lipschitz smooth in $\mathbb{R}^n$, then f is also $L_{\mathbb{M}}$-geodesic smooth on $\mathbb{M}$, where $L_{\mathbb{M}}$ is determined by the manifold $\mathbb{M}$ and f.*

Definition 6 [3] A vector transport on $\mathbb{M}$ is a smooth mapping $\mathbb{T} : \mathcal{T}\mathbb{M} \oplus \mathcal{T}\mathbb{M} \to \mathcal{T}_x\mathbb{M} : (\eta, \xi) \to \mathbb{T}_\eta(\xi)$ that satisfies the following conditions for any $x \in \mathbb{M}$: (i) there exists a retraction $\mathbf{Re}$ such that $\mathbb{T}_\eta(\xi) \in \mathcal{T}_{\mathbf{Re}_x(\eta)}\mathbb{M}$, $\forall \eta,\ \xi \in \mathcal{T}_x\mathbb{M}$; (ii) (consistency) $\mathbb{T}_{0_x}(\xi) = \xi$ for all $\xi \in \mathcal{T}_x\mathbb{M}$; (iii) (linearity) $\mathbb{T}_\eta(a\xi + b\gamma) = a\mathbb{T}_\eta(\xi) + b\mathbb{T}_\eta(\gamma)$ for all $a, b \in \mathbb{R}$ and $\eta, \xi, \gamma \in \mathcal{T}_x\mathbb{M}$. Here, $\oplus$ denotes the Whitney sum. The vector transport $\mathbb{T}^y_x(v)$ or equivalently $\mathbb{T}_u(v)$ with $y = \mathbf{Re}_x(u)$ transports $v \in \mathcal{T}_x\mathbb{M}$ along the retraction curve definded by direction u to the $\mathcal{T}_y\mathbb{M}$.

Definition 7 [46] For (x^*, y^*, λ^*) where $x^* \in \mathbb{M}$, if there exits $\mathcal{R}^*$ such that

$$\mathcal{R}^* \in \begin{pmatrix} \mathbf{Proj}_{x^*}\Big\{\nabla A(x^*) + \mathcal{C}(x^*)^\top \lambda^*\Big\} \\ \partial\mathcal{B}(y^*) + \lambda^* \\ C(x^*) + y^* \end{pmatrix},$$

then (x^*, y^*, λ^*) is called ϵ stationary solution if $\|\mathcal{R}^*\| \leqslant \epsilon$.

3 The inertial Riemannian gradient ADMM

In this section, we presented iRG-ADMM in Algorithm 1, which incorporates an inertial Riemannian gradient technique with variable step sizes to ensure fast convergence of the problem (3).

First, we present some fundamental assumptions regarding Problem (3).

Assumption 1 (i) The function $\mathcal{B} : \mathbb{R}^m \to \mathbb{R}$ is proper and lower-semicontinuous with an effective domain denoted by $\mathrm{dom}\mathcal{B} := \{x \in \mathbb{R}^m \mid \mathcal{B}(x) < +\infty\}$ and satisfies the implicit Lipschitz subgradient property . Moreover, the proximal oracle of $\mathcal{B}$ is available, i.e., given $p \in \mathbb{R}^m$ and a constant $\gamma > 0$, we can solve the following problem:

$$\min\{\mathcal{B}(y) + \frac{\gamma}{2}\|y - z\|^2\},\ y \in \mathrm{dom}\mathcal{B}.$$

(ii) $\mathcal{A}(x) + \mathcal{B}(y)$ is bounded below, i.e.,

$$\underline{\mathbf{F}} := \inf_{x\in\mathbb{M},\ y\in\mathrm{dom}\mathcal{B}} \{\mathcal{A}(x) + \mathcal{B}(y)\} > -\infty.$$

(iii) The mapping $\mathcal{C} : \mathbb{M} \subset \mathbb{R}^n \to \mathbb{R}^q$ is gradient bounded with $\mathbf{B}_C$, i.e.,

$$\|\nabla\mathcal{C}(x)\| \leqslant \mathbf{B}_C, \forall x \in \mathbb{M}.$$

(iv) Fixed $(\bar{y}, \bar{\lambda})$, the function $\mathcal{L}_\rho(x, \bar{y}, \bar{\lambda})$ is $L_{\mathbb{M}}$-smooth.

Now we can propose the iRG-ADMM.

3.1 Descript of iRG-ADMM

Algorithm 1 iRG-ADMM

Given initial vector $(x^0, y^0, \hat{y}^0, \lambda^0)$ and postive constant $\psi \in (0.5, 2)$ and $\alpha \leqslant \min\{1, \frac{1}{2L_{\mathbb{M}}}\}$. Let $\mathbf{d}_{-1} = \hat{g}^0_x$.

for $k = 0, \ldots$ **do**

% x-**subproblem**:

[Step 1] Compute Riemannian gradient at x^k with as follows,

$$\hat{g}^k_x = \mathbf{Proj}_{x^k}\Big\{\nabla\mathcal{C}(x^k)^\top(\lambda^k + \rho(\mathcal{C}(x^k) + y^k)) + \nabla\mathcal{A}(x^k)\Big\}.$$

[Step 2] Compute the inertial tangent vector in the $\mathcal{T}_{x^k}\mathbb{M}$ as $\mathbf{d}_k = \hat{g}^k_x + \varphi_k(\hat{g}^k_x - \mathbb{T}^{x^k}_{x^{k-1}}(\mathbf{d}_{k-1}))$ with $\varphi_k = \min\{\frac{\|\alpha\hat{g}^k_x\|}{\|\hat{g}^k_x - \mathbb{T}^{x^k}_{x^{k-1}}(\mathbf{d}_{k-1})\|}, 1\}$, then we set $x^{k+1} = \mathbf{R}_{x^k}(-\alpha\mathbf{d}_k)$.

% y-**subproblem**:

[Step 3] Update the auxilary variable $\hat{y}^k = (1 - \psi)\hat{y}^{k-1} + \psi y^k$,

[Step 4] Solve $y^{k+1} = \underset{y\in\mathbb{R}^m}{\mathrm{argmin}} \Big\{\hat{\mathcal{B}}_k(y)\Big\}$, where

$$\hat{\mathcal{B}}_k(y) := \mathcal{B}(y) + \langle\lambda^k, y\rangle + \frac{\rho}{2}\|\mathcal{C}(x^{k+1}) + y\|^2 + \frac{\gamma}{2}\|y - \hat{y}^k\|^2.$$

% λ-**subproblem**:

[Step 5] $\lambda^{k+1} = \lambda^k + \rho(\mathcal{C}(x^{k+1}) + y^{k+1})$.

end for

$x^{k+1}, y^{k+1}, \lambda^{k+1}, \hat{y}^{k+1}$.

Remark 2 *(i) The parameters ρ in Algorithm1 could be selected as follows,*

$$\rho \geqslant \max\Big\{\frac{4(L_g + \gamma)^2}{\gamma}, \frac{4\psi^2}{4\psi^2 - 1}\Big\}.$$

The aforementioned selection of ρ is crucial for our subsequent convergence analysis. Moreover, when implementing iRG-ADMM in practice, the values of ψ and γ are predetermined. Thus, in accordance with the above conditions, it becomes relatively easy to choose an appropriate value for ρ.

(ii) In general, the efficiency of solving the first subproblem plays a crucial role in determining the overall

performance of ADMM-based algorithms. However, directly solving subproblems on manifolds often introduces certain challenges. In this paper, we leverage the smoothness properties of the x-subproblem to compute its Riemannian gradient and incorporate ideas from the inertial strategy [37,38] to construct a new update direction $\boldsymbol{d}_k$ for solving the x-subproblem. First, we compute the Riemannian gradient $\hat{g}_x^k$ and then, using an appropriate vector transport, combine the "inertia" from the previous update direction $\boldsymbol{d}_{k-1}$ with the current Riemannian gradient $\hat{g}_x^k$. This inertial strategy helps mitigate oscillations, enhances the stability of the solution to the x-subproblem, and, in turn, improves the overall efficiency of the algorithm.

(iii) To solve the y-subproblem, we introduce an auxiliary variable $\hat{y}$ along with a proximal term to ensure convergence. Since y is constrained within Euclidean space, we can apply a variety of well-established algorithms designed for Euclidean optimization to efficiently solve this subproblem.

3.2 Convergence analysis

In this section, we establish the ϵ-stationary solution of iRG-ADMM. First, we present a lemma concerning the sequence of Lagrange multipliers $\{\lambda^k\}$.

Lemma 3 *Suppose that the sequence $\{y^k, \hat{y}^k, \lambda^k\}$ is generated by Algorithm 1. Then, for $k \geqslant 1$, we have*

$$\begin{aligned}\|\lambda^{k+1} - \lambda^k\|^2 \leqslant & 2(L_{\mathcal{B}} + \gamma)^2 \|y^{k+1} - y^k\|^2 \\ & + 2\gamma^2 \|\hat{y}^k - \hat{y}^{k-1}\|^2.\end{aligned} \tag{5}$$

PROOF. From optimality condition of y-subproblem in Algorithm 1, we have

$$\begin{aligned}0 &\in \partial \mathcal{B}(y^{k+1}) + \lambda^k + \beta(\mathcal{C}(x^k) + y^k) + \gamma(y^{k+1} - \hat{y}^k) \\ &= \partial \mathcal{B}(y^{k+1}) + \lambda^{k+1} + \gamma(y^{k+1} - \hat{y}^k) \\ &= \partial \left(\mathcal{B} + \frac{\gamma}{2} \| \cdot - (\hat{y}^k + \frac{\lambda^{k+1}}{\gamma}) \|^2 \right) (y^{k+1}),\end{aligned}$$

then we have

$$y^{k+1} = \mathrm{Prox}_{\gamma, \mathcal{B}}(\hat{\mathrm{y}}^{\mathrm{k}} + \frac{\lambda^{\mathrm{k+1}}}{\gamma}),$$

thus

$$\nabla M_{\mathcal{B}}(\hat{y}^k + \frac{\lambda^{k+1}}{\gamma}) = \lambda^{k+1} + \gamma(\hat{y}^k - y^{k+1}),$$

it holds that

$$\begin{aligned}\|\lambda^{k+1} - \lambda^k\| = & \|\nabla M_{\mathcal{B}}(\hat{y}^k + \frac{\lambda^{k+1}}{\gamma}) - \nabla M_{\mathcal{B}}(\hat{y}^{k-1} + \frac{\lambda^k}{\gamma}))\| \\ & + \gamma \|\hat{y}^k - \hat{y}^{k-1}\| + \gamma \|y^{k+1} - y^k\| \\ \leqslant & (L_{\mathcal{B}} + \gamma) \|y^{k+1} - y^k\| + \gamma \|\hat{y}^k - \hat{y}^{k-1}\|.\end{aligned}$$

Finally, we have

$$\|\lambda^{k+1} - \lambda^k\|^2 \leqslant 2(L_{\mathcal{B}} + \gamma)^2 \|y^{k+1} - y^k\|^2 + 2\gamma^2 \|\hat{y}^k - \hat{y}^{k-1}\|^2.$$

■

To proceed, we denote sequence $\{\omega^k\} = \{x^k, y^k, \hat{y}^k, \lambda^k\}$ and function $\hat{\mathcal{L}}_\rho(\omega^k) := \mathcal{L}_\rho(x^k, y^k, \lambda^k) + \frac{\gamma}{2}\|y^k - \hat{y}^k\|$, which plays key role in the convergence analysis. Then, we provide a lemma to characterize the progress achieved by a single iterate of the proposed algorithm.

Lemma 4 *Suppose that the sequence $\{\omega^k\}$ is generated by Algorithm 1. Then, for $k \geqslant 1$, it holds that*

$$\begin{aligned}&\tilde{\mathcal{L}}_\rho(\omega^{k+1}) - \tilde{\mathcal{L}}_\rho(\omega^k) \\ =& -(\frac{\gamma}{2} - \frac{2(L_{\mathcal{B}} + \gamma)^2}{\rho}) \|y^{k+1} - y^k\|^2 + \frac{2\gamma^2}{\rho} \|\hat{y}^k - \hat{y}^{k-1}\|^2 \\ & - \frac{\gamma(2-\psi)}{2\psi} \|\hat{y}^{k+1} - \hat{y}^k\|^2 - \frac{\alpha - \alpha^3}{2} \|\hat{g}_x^k\|^2.\end{aligned} \tag{6}$$

PROOF. We begin the proof from the x-subproblem. If (iv) in Assumption 1 holds, then it is easy to obtain

$$\begin{aligned}&\tilde{\mathcal{L}}_\rho(x^{k+1}, y^k, \hat{y}^k, \lambda^k) - \tilde{\mathcal{L}}_\rho(x^k, y^k, \hat{y}^k, \lambda^k) \\ \leqslant & \langle \hat{g}_x^k, -\alpha \mathbf{d}_k \rangle + \frac{L_{\mathbb{M}}}{2} \|\alpha \mathbf{d}_k\|^2,\end{aligned}$$

then we have

$$\begin{aligned}&\tilde{\mathcal{L}}_\rho(x^{k+1}, y^k, \hat{y}^k, \lambda^k) - \tilde{\mathcal{L}}_\rho(x^k, y^k, \hat{y}^k, \lambda^k) \\ \leqslant & \frac{\alpha}{2} \|\hat{g}_x^k - \mathbf{d}_k\|^2 - \frac{\alpha}{2} \|\hat{g}_x^k\|^2 - \frac{\alpha}{2} \|\mathbf{d}_k\|^2 + \frac{L_{\mathbb{M}}}{2} \|\alpha \mathbf{d}_k\|^2 \\ \leqslant & \frac{\alpha \psi_k^2}{2} \|\hat{g}_x^k - \mathbb{T}_{x^{k+1}}^{x^k}(\mathbf{d}_{k-1})\|^2 - \frac{\alpha}{2} \|\hat{g}_x^k\|^2 - \frac{\alpha}{4} \|\mathbf{d}_k\|^2 \\ \leqslant & \frac{\alpha^3 - \alpha}{2} \|\hat{g}_x^k\|^2 - \frac{\alpha}{4} \|\mathbf{d}_k\|^2 \\ \leqslant & -\frac{\alpha - \alpha^3}{2} \|\hat{g}_x^k\|^2.\end{aligned} \tag{7}$$

Consider the y-subproblem with the strong convexity of $\hat{\mathcal{B}}_k(y)$, we obtain

$$\begin{aligned}&\mathcal{B}(y^{k+1}) + \langle \lambda^k, y^{k+1} \rangle + \frac{\rho}{2} \|\mathcal{C}(x^{k+1}) + y^{k+1}\|^2 \\ &\quad + \frac{\gamma}{2} \|y^{k+1} - \hat{y}^k\|^2 \\ \leqslant & \mathcal{B}(y^k) + \langle \lambda^k, y^k \rangle + \frac{\rho}{2} \|\mathcal{C}(x^{k+1}) + y^k\|^2 + \frac{\gamma}{2} \|y^k - \hat{y}^k\|^2 \\ &\quad - \frac{\gamma}{2} \|y^{k+1} - y^k\|^2,\end{aligned}$$

which can be equivalently expressed as

$$\begin{aligned}&\tilde{\mathcal{L}}_\rho(x^{k+1}, y^{k+1}, \hat{y}^k, \lambda^k) - \tilde{\mathcal{L}}_\rho(x^{k+1}, y^k, \hat{y}^k, \lambda^k) \\ \leqslant & -\frac{\gamma}{2} \|y^{k+1} - y^k\|^2.\end{aligned} \tag{8}$$

To proceed, considering the updating of $\hat{y}^k$, it holds that

$$\begin{aligned}
&\tilde{\mathcal{L}}_\rho(x^{k+1},y^{k+1},\hat{y}^{k+1},\lambda^k)-\tilde{\mathcal{L}}_\rho(x^{k+1},y^{k+1},\hat{y}^{k},\lambda^k)\\
=&\frac{\gamma}{2}\|y^{k+1}-\hat{y}^{k+1}\|^2-\frac{\gamma}{2}\|y^{k+1}-\hat{y}^{k}\|^2\\
=&\frac{\gamma}{2}(\|y^{k+1}-\hat{y}^{k}\|^2-2\langle\hat{y}^{k+1}-\hat{y}^{k},y^{k+1}-\hat{y}^{k}\rangle\\
&+\|\hat{y}^{k+1}-\hat{y}^{k}\|^2)-\frac{\gamma}{2}\|y^{k+1}-\hat{y}^{k}\|^2\\
=&\frac{\gamma}{2}\|\hat{y}^{k+1}-\hat{y}^{k}\|^2-\gamma\langle\hat{y}^{k+1}-\hat{y}^{k},y^{k+1}-\hat{y}^{k}\rangle.
\end{aligned}$$

From updating of $\hat{y}^k$ in Algorithm 1, we have

$$\begin{aligned}
&\tilde{\mathcal{L}}_\rho(x^{k+1},y^{k+1},\hat{y}^{k+1},\lambda^k)-\tilde{\mathcal{L}}_\rho(x^{k+1},y^{k+1},\hat{y}^{k},\lambda^k)\\
=&\frac{\gamma}{2}\|\hat{y}^{k+1}-\hat{y}^{k}\|^2-\frac{\gamma}{\psi}\|\hat{y}^{k+1}-\hat{y}^{k}\|^2.
\end{aligned}\tag{9}$$

By the λ-subproblem , it holds that

$$\begin{aligned}
&\tilde{\mathcal{L}}_\rho(x^{k+1},y^{k+1},\hat{y}^{k+1},\lambda^{k+1})-\tilde{\mathcal{L}}_\rho(x^{k+1},y^{k+1},\hat{y}^{k+1},\lambda^k)\\
=&\frac{1}{\rho}\|\lambda^{k+1}-\lambda^k\|^2.
\end{aligned}\tag{10}$$

Recall (5) and (10), such that

$$\begin{aligned}
&\tilde{\mathcal{L}}_\rho(x^{k+1},y^{k+1},\hat{y}^{k+1},\lambda^{k+1})-\tilde{\mathcal{L}}_\rho(x^{k+1},y^{k+1},\hat{y}^{k+1},\lambda^k)\\
\leqslant&\frac{2(L_g+\gamma)^2}{\rho}\|y^{k+1}-y^k\|^2+\frac{2\gamma^2}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|^2.
\end{aligned}\tag{11}$$

Combining (7), (8), (9) and (11), we have

$$\begin{aligned}
&\tilde{\mathcal{L}}_\rho(x^{k+1},y^{k+1},\hat{y}^{k+1},\lambda^{k+1})-\tilde{\mathcal{L}}_\rho(x^{k},y^{k},\hat{y}^{k},\lambda^k)\\
=&-(\frac{\gamma}{2}-\frac{2(L_{\mathcal{B}}+\gamma)^2}{\rho})\|y^{k+1}-y^k\|^2+\frac{2\gamma^2}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|^2\\
&-\frac{\gamma(2-\psi)}{2\psi}\|\hat{y}^{k+1}-\hat{y}^k\|^2-\frac{\alpha-\alpha^3}{2}\|\hat{g}_x^k\|^2.
\end{aligned}$$

■

Remark 5 *(i) For most commonly used machine learning models, the constraint $\mathcal{C}(x)+y=0$ in Problem (3) simplifies to the form $\hat{C}x+y-b=0$, where $\hat{C}$ is a matrix. Additionally, in these models, the smooth function $\mathcal{A}$ is often gradient Lipschitz continuous (with Lipschit constant L_A). These conditions, tailored to practical applications, render Lemma 4 independent of (iv) in Assumption 1. In fact,*

$$\nabla\tilde{\mathcal{L}}_\rho(x,y^k,\hat{y}^k,\lambda^k)=\nabla\mathcal{A}(x)+C^\top\Big(\lambda^k+\beta(Cx+y^k-b)\Big),$$

for any x_1, $x_2\in\mathbb{M}$, we can deduce that

$$\begin{aligned}
&\|\nabla\tilde{\mathcal{L}}_\rho(x_1,y^k,\hat{y}^k,\lambda^k)-\nabla\tilde{\mathcal{L}}_\rho(x_2,y^k,\hat{y}^k,\lambda^k)\|\\
\leqslant&\|\nabla\mathcal{A}(x_1)-\nabla\mathcal{A}(x_2)\|+\beta C^\top C\|x_1-x_2\|\\
\leqslant&(L_A+\beta C^\top C)\|x_1-x_2\|.
\end{aligned}$$

Recall Lemma 1, we can conclude that $\tilde{\mathcal{L}}_\rho(x,y^k,\hat{y}^k,\lambda^k)$ is geodesically smooth. The above discussion demonstrates that, for a large class of practical problems, the geodesic smoothness of $\tilde{\mathcal{L}}_\rho(x,y^k,\hat{y}^k,\lambda^k)$ naturally holds.

(ii) For $k\geq 0$, we define a Lyapunov function as follows,

$$\mathcal{F}_\rho(\tilde{\omega}^k):=\tilde{\mathcal{L}}_\rho(\omega^k)+\frac{2\gamma^2}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|^2,$$

where $\tilde{\omega}^k:=(\omega^k,\hat{y}^{k-1})$. Recall (6), it is noticed that $\mathcal{F}_\rho(\tilde{\omega}^k)$ is non-increasing monotonely and there exists a constant $\zeta^*>0$, holds that

$$\begin{aligned}
&\mathcal{F}_\rho(\tilde{\omega}^k)-\mathcal{F}_\rho(\tilde{\omega}^{k+1})\\
\geqslant&\zeta^*(\|y^{k+1}-y^k\|^2-\|\hat{y}^{k+1}-\hat{y}^k\|^2-\|\hat{g}_x^k\|^2).
\end{aligned}\tag{12}$$

Next, we present a lemma to demonstrate that $\mathcal{F}_\rho$ is bounded below.

Lemma 6 *If the parameters conditions are satisfied and Assumptions 1 holds, then the Lyapunov functions $\mathcal{F}_\rho$ are lower bounded $\overline{\boldsymbol{F}}$.*

PROOF. It is obvious that $\mathcal{F}_\rho(\tilde{\omega}^k)\geqslant\tilde{\mathcal{L}}_\rho(\omega^k)+\frac{2\gamma^2}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|^2$. Besides, recall that $\nabla M_{\mathcal{B}}(\hat{y}^k+\frac{\lambda^{k+1}}{\gamma})=\lambda^{k+1}+\gamma(\hat{y}^k-y^{k+1})$, we have

$$\begin{aligned}
&\tilde{\mathcal{L}}_\rho(\omega^k)+\frac{2\gamma^2}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|^2\\
=&\mathcal{A}(x^k)+\mathcal{B}(y^k)+\frac{\rho}{2}\|\mathcal{C}(x^k)+y^k\|^2+\frac{\gamma}{2}\|y^k-\hat{y}^k\|^2\\
&+\langle\nabla M_{\mathcal{B}}(\hat{y}^{k-1}+\frac{\lambda^k}{\gamma})-\gamma(\hat{y}^k-y^k),\mathcal{C}(x^k)+y^k\rangle\\
&+\frac{2\gamma^2}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|^2\\
=&\mathcal{A}(x^k)+\mathcal{B}(y^k)+\frac{\rho}{2}\|\mathcal{C}(x^k)+y^k\|^2+\frac{\gamma}{2}\|y^k-\hat{y}^k\|^2\\
&+\langle\nabla M_{\mathcal{B}}(\hat{y}^{k-1}+\frac{\lambda^k}{\gamma})-\gamma(\hat{y}^k-y^k),\mathcal{C}(x^k)+y^k\rangle\\
&-\gamma\langle\hat{y}^{k-1}-y^k,\mathcal{C}(x^k)+y^k\rangle+\frac{2\gamma^2}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|^2\\
\geqslant&\mathcal{A}(x^k)+\mathcal{B}(y^k)-\frac{1}{2}\|\nabla M_{\mathcal{B}}(\hat{y}^{k-1}+\frac{\lambda^k}{\gamma})\|^2\\
&+\frac{\gamma}{\psi}\langle\hat{y}^k-\hat{y}^{k-1},\mathcal{C}(x^k)+y^k\rangle+\frac{\gamma}{2}\|y^k-\hat{y}^k\|^2\\
&+\frac{\rho-1}{2}\|\mathcal{C}(x^k)+y^k\|^2+\frac{2\gamma^2}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|^2.
\end{aligned}$$

Rearrange the right side of inequality, we have

$$\begin{aligned}
&\tilde{\mathcal{L}}_\rho(\omega^k)+\frac{2\gamma^2}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|^2\\
\geqslant&\mathcal{A}(x^k)+\mathcal{B}(y^k)+\frac{\gamma}{2}\|y^k-\hat{y}^k\|^2\\
&+\frac{\rho-1}{2}\|\mathcal{C}(x^k)+y^k+\frac{\gamma}{\psi(\rho-1)}(\hat{y}^k-\hat{y}^{k-1})\|^2\\
&+(\frac{2\gamma^2}{\rho}-\frac{\gamma^2}{2\psi^2(\rho-1)})\|y^k-\hat{y}^k\|^2\\
\geqslant&\mathcal{A}(x^k)+\mathcal{B}(y^k).
\end{aligned}$$

Then, it is obviuos that

$$\mathcal{F}_\rho(\tilde{\omega}^k)\geqslant\overline{\mathbf{F}},$$

this finishes the proof. ■

Now, denote $\mathcal{F}_\rho(\tilde{\omega}^k)\leqslant\mathcal{F}_\rho(\tilde{\omega}^0):=\underline{\mathbf{F}}$, and we proceed to analyze the ϵ-stationary solution of iRG-ADMM.

Theorem 7 *Assume that the sequence $\{x^k,y^k,\hat{y}^k,\lambda^k\}$ is generated by Algorithm 1, the assumption 1 and parameters conditions hold. Then Algorithm 1 finds an ϵ-stationary solution of (3) in at most T iterations, where*

$$T:=\frac{8\boldsymbol{L}(\rho)^2(\underline{\boldsymbol{F}}-\overline{\boldsymbol{F}})}{\zeta^*\epsilon^2},$$

and

$$\boldsymbol{L}(\rho):=\frac{(\boldsymbol{B}_C+1)(L_{\mathcal{B}}+2\gamma)}{\rho}+\frac{\gamma}{\psi}+1.$$

PROOF. Let us begin with the following relation,

$$\|\mathcal{C}(x^{k+1})+y^{k+1}\|=\frac{1}{\rho}\|\lambda^{k+1}-\lambda^k\|,$$

then we have

$$\|\mathcal{C}(x^{k+1})+y^{k+1}\|\leqslant\frac{(L_{\mathcal{B}}+\gamma)}{\rho}\|y^{k+1}-y^k\|+\frac{\gamma}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|.$$

Considering the Riemannian gradient updating in the Step 1 in Algorithm 1, it holds that

$$\begin{aligned}
&\mathbf{Proj}_{x^{k+1}}\Big\{\nabla\mathcal{C}(x^{k+1})^\top(\lambda^{k+1}+\rho(\mathcal{C}(x^{k+1})+y^{k+1}))\\
&\qquad+\nabla\mathcal{A}(x^{k+1})\Big\}\\
=&\hat{g}_x^{k+1}.
\end{aligned}$$

Consequently, we obtain

$$\begin{aligned}
&\mathrm{dist}\Big(\mathbf{Proj}_{x^k}\Big\{\nabla\mathcal{C}(x^{k+1})^\top\lambda^{k+1}+\nabla\mathcal{A}(x^{k+1})\Big\},0\Big)\\
=&\mathrm{dist}\Big(\hat{g}_x^{k+1}-\nabla\mathcal{C}(x^{k+1})^\top(\mathcal{C}(x^{k+1})+y^{k+1}),0\Big)\\
\leqslant&\|\hat{g}_x^{k+1}\|+\frac{\mathbf{B}_C}{\rho}\|\lambda^{k+1}-\lambda^k\|.\\
\leqslant&\|\hat{g}_x^{k+1}\|+\frac{\mathbf{B}_C(L_{\mathcal{B}}+\gamma)}{\rho}\|y^{k+1}-y^k\|+\frac{\mathbf{B}_C\gamma}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|.
\end{aligned}$$

Considering the y-subproblem. such that

$$\begin{aligned}
&\lambda^k+\beta(\mathcal{C}(x^{k+1})+y^{k+1})+\gamma(y^{k+1}-\hat{y}^k)+\tilde{B}_{k+1}\\
=&\lambda^{k+1}+\gamma(y^{k+1}-\hat{y}^k)+\tilde{B}_{k+1}=0,
\end{aligned}$$

where $\tilde{B}_{k+1}\in\partial\mathcal{B}(y^{k+1})$. To proceed, we have

$$\mathrm{dist}\Big(\tilde{B}_{k+1}+\lambda^{k+1},0\Big)\leqslant\gamma\|y^{k+1}-\hat{y}^k\|=\frac{\gamma}{\psi}\|\hat{y}^{k+1}-\hat{y}^k\| \tag{13}$$

We take

$$\mathcal{R}_{k+1}\in\begin{pmatrix}\mathbf{Proj}_{x^{k+1}}\Big\{\nabla A(x^{k+1})+\mathcal{C}(x^{k+1})^\top\lambda^{k+1}\Big\}\\ \partial\mathcal{B}(y^{k+1})+\lambda^{k+1}\\ C(x^{k+1})+y^{k+1}\end{pmatrix},$$

which implies

$$\begin{aligned}
&\|\mathcal{R}_{k+1}\|\\
\leqslant&\mathrm{dist}\Big(\mathbf{Proj}_{x^{k+1}}\Big\{\nabla\mathcal{C}(x^{k+1})^\top\lambda^{k+1}+\nabla\mathcal{A}(x^{k+1})\Big\},\ 0\Big)\\
&+\mathrm{dist}\Big(\tilde{B}_{k+1}+\lambda^{k+1},0\Big)+\|\mathcal{C}(x^{k+1})+y^{k+1}\|,
\end{aligned}$$

then we have

$$\begin{aligned}
\|\mathcal{R}_{k+1}\|\leqslant&\|\hat{g}_x^{k+1}\|+\frac{(\mathbf{B}_C+1)(L_{\mathcal{B}}+\gamma)}{\rho}\|y^{k+1}-y^k\|\\
&+\frac{(\mathbf{B}_C+1)\gamma}{\rho}\|\hat{y}^k-\hat{y}^{k-1}\|+\frac{\gamma}{\psi}\|\hat{y}^{k+1}-\hat{y}^k\|\\
\leqslant&\mathbf{L}(\rho)(\|\hat{g}_x^{k+1}\|+\|y^{k+1}-y^k\|+\|\hat{y}^k-\hat{y}^{k-1}\|\\
&+\|\hat{y}^{k+1}-\hat{y}^k\|).
\end{aligned}$$

On the other hand, we consider (12) such that

$$\begin{aligned}
&\zeta^*\sum_{k=1}^{T}(\|\hat{g}_x^{k+1}\|^2+\|y^{k+1}-y^k\|^2+\|\hat{y}^{k+1}-\hat{y}^k\|^2\\
&\qquad+\|\hat{y}^k-\hat{y}^{k-1}\|^2)\\
\leqslant&2\underline{\mathbf{F}}-2\overline{\mathbf{F}}.
\end{aligned}$$

As a result, there exists an index $1\leqslant k^*<T$, using the

fact $\sum_{i=1}^{p}\|a_i\| \leqslant \sqrt{p\sum_{i=1}^{p}a_i^2}$, such that

$$\|\hat{g}_x^{k^*+1}\| + \|y^{k^*+1} - y^{k^*}\| + \|\hat{y}^{k^*+1} - \hat{y}^{k^*}\| + \|\hat{y}^{k^*} - \hat{y}^{k^*-1}\| \leqslant \sqrt{\frac{8(\underline{\mathbf{F}} - \overline{\mathbf{F}})}{\zeta^* T}}.$$

Then it holds that

$$\|\mathcal{R}_{k^*+1}\| \leqslant \mathbf{L}(\rho)\sqrt{\frac{8(\underline{\mathbf{F}} - \overline{\mathbf{F}})}{\zeta^* T}} \leqslant \epsilon.$$

We can finish this proof by the above discussion. ■

4 Numerical experiments

In this section, we examine the numerical performance of the proposed iRG-ADMM algorithm and report comparison results with the existing methods on Sparse Principal Component Analysis (SPCA).

Principal Component Analysis (PCA) is widely used in data processing and dimensionality reduction. However, a notable limitation of PCA is that each principal component is a linear combination of all the original variables, which can make the interpretation of the results challenging. To address this issue, incorporating sparse structures into PCA and balancing the sparsity of the solution with the quality of the principal component analysis becomes highly significant. We consider the following Sparse Principal Component Analysis (SPCA) model:

$$\begin{aligned} \min\ & -\frac{1}{2}\mathrm{Tr}(X^\top H^\top H X) + \mu\|X\|_1 \\ \text{s.t. } & X \in \mathrm{St}(n,p), \end{aligned} \tag{14}$$

which can be formulated as following seperable stucture,

$$\begin{aligned} \min\ & -\frac{1}{2}\mathrm{Tr}(X^\top H^\top H X) + \mu\|Y\|_1 \\ \text{s.t. } & X = Y,\ X \in \mathrm{St}(n,p), \end{aligned} \tag{15}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of matrix , the l_1-norm is defined as $\|X\|_1 = \sum_{ij} \mid X_{ij} \mid$. $H \in \mathbb{R}^{m\times n}$ is a data matrix, $\mu > 0$ is a regularization parameter and $\mathrm{St}(n,p)$ is a Stiefel manifold.

In this experiment, we introduce a series of popular algorithms to solve SPCA (14), Which are ManPG-Ada [18], SOC [32] and MADMM [33], to compare with Algorithm 1. First, we propose the iteration frameworks of these algorithms.

The ManPG-Ada in [18] for solving (14) has a basic iterates as follows:

$$\begin{cases} V^{k+1} = \underset{V\in\mathrm{St}(n,p)}{\mathrm{argmin}}\ \{\langle -H^\top H X^k, V\rangle + \dfrac{1}{2t}\|V\|^2 \\ \qquad\qquad + \mu\|D(X^k + V)\|_1\}, \\ X^{k+1} = \mathbf{Re}_{x^k}(\theta V^k), \end{cases} \tag{16}$$

where θ and t are stepsizes. The authors of [18] suggest to solve the V-subproblem by using a semi-smooth Newton method. For SOC [32], we adjust the (14) and makes the auxiliary variable Y constraints on the $\mathrm{St}(n,p)$, the modified SPCA model is as follows.

$$\begin{aligned} \min\ & -\frac{1}{2}\mathrm{Tr}(X^\top H^\top H X) + \mu\|Y\|_1 \\ \text{s.t. } & X = Y,\ Y \in \mathrm{St}(n,p), \end{aligned} \tag{17}$$

then the iteration framework of SOC [32] solves the (17) is given by

$$\begin{cases} X^{k+1} = \underset{X}{\mathrm{argmin}}\{-\dfrac{1}{2}\mathrm{Tr}(X^\top H^\top H X) + \mu\|X\|_1 \\ \qquad\qquad + \langle\lambda^k, X - Y^k\rangle + \dfrac{\rho}{2}\|X - Y^k\|_F^2\} \\ Y^{k+1} = \underset{Y\in\mathrm{St}(n,p)}{\mathrm{argmin}}\ \{\langle\lambda^k, X^{k+1} - Y\rangle + \dfrac{\rho}{2}\|X^{k+1} - Y\|_F^2\}, \\ \lambda^{k+1} = \lambda^k + \beta(X^{k+1} - Y^{k+1}). \end{cases} \tag{18}$$

To apply Algorithm 1 and MADMM [33], we introduce the ALF for (15) as follows,

$$\begin{aligned} \mathcal{L}_\rho(X,Y,\lambda) = & -\frac{1}{2}\mathrm{Tr}(X^\top H^\top H X) + \mu\|Y\|_1 - \langle\lambda, X - Y\rangle \\ & + \frac{\rho}{2}\|X - Y\|_F^2, \end{aligned}$$

then MADMM [33] for solving the problem (15) has following iteration framework.

$$\begin{cases} X^{k+1} = \underset{X\in\mathrm{St}(n,p)}{\mathrm{argmin}}\ \{\mathcal{L}_\rho(X, Y^k, \lambda^k),\} \\ Y^{k+1} = \underset{Y}{\mathrm{argmin}}\{\mathcal{L}_\rho(X^{k+1}, Y, \lambda^k)\}, \\ \lambda^{k+1} = \lambda^k + \rho(X^{k+1} - Y^{k+1}). \end{cases} \tag{19}$$

Based on the gradient of $\mathcal{L}_\rho(X,Y,\lambda)$ with respect to X, we can present iteration framework of Algorithm 1 as follows,

$$\begin{cases} \hat{g}_x^k = \mathbf{Proj}_{X^k}(-H^\top H X^k + \lambda^k + \rho(X^k - Y^k)), \\ \mathbf{d}_k = \hat{g}_x^k + \varphi_k \cdot (\hat{g}_x^k - \mathbb{T}_{x^{k-1}}^{x^k}(\mathbf{d}_{k-1})), \\ X^{k+1} = \mathbf{Re}_{x^k}(-\alpha\mathbf{d}_k), \\ \hat{Y}^k = (1-\psi)\hat{Y}^{k-1} + \psi Y^k, \\ Y^{k+1} = \mathrm{Prox}_{\frac{\mu}{\gamma+\beta},\|\|_1}(\dfrac{-\lambda^k + \gamma\hat{Y}^k}{\gamma+\rho}), \\ \lambda^{k+1} = \lambda^k + \rho(X^{k+1} - Y^{k+1}). \end{cases} \tag{20}$$

We now give the setups of this numerical experiment. The data matrix $H \in \mathbb{R}^{m\times n}$ is generated randomly whose entries follow the standard Gaussian distribution $\mathcal{N}(0,1)$. We choose $\mu = 1$, n from

$\{300, 500, 700, 900\}$, and p from $\{50, 100\}$. In this experiment, We terminate all the algorithms as following criterion with respect fo value of objective function F in (14), which means

$$\frac{\|F(X^{k+1}) - F(X^k)\|}{\|F(X^{k+1})\|} < 10^{-10}.$$

Additionally, we use objective values and CPU time to illustrate the behavior of the algorithms.

For the algorithms used in the experiments, we set the parameters as follows: In (16), the parameters are set to their default values, as specified in the corresponding paper [18]; in (18), we solve the Y-subproblem using the proximal gradient method with $\rho = 50$ and step size $\eta = 0.01$; in (19), the X-subproblem is solved using the Riemannian gradient method with $\rho = 100$ and step size $\eta = 0.01$; in (20), we set $\gamma = 1$, $\rho = 20$, $\alpha = 0.015$, and $\psi = 0.618$ based on empirical experience.

Now we can conduct the experiments to test the performance of the algorithms.

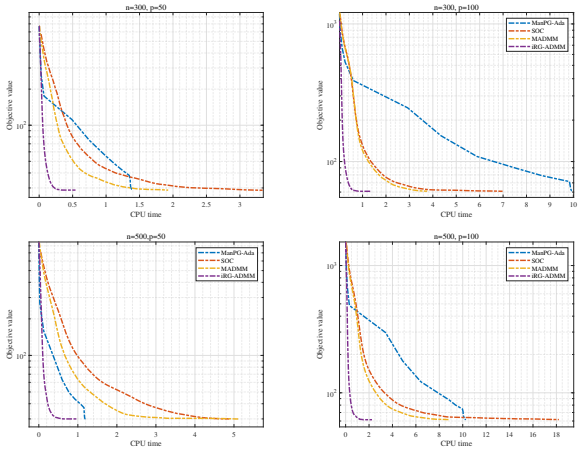

Fig. 1. Comparison of the ManPG-Ada, SOC, MADMM and iRG-ADMM algorithms.

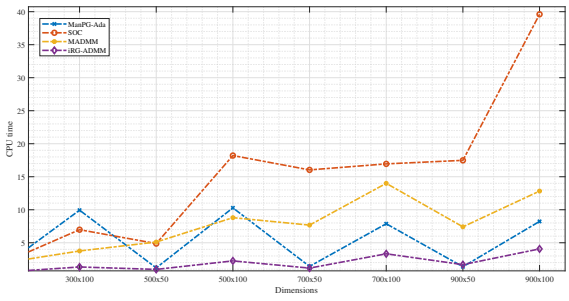

Fig. 2. Trends of CPU time under different (n, p).

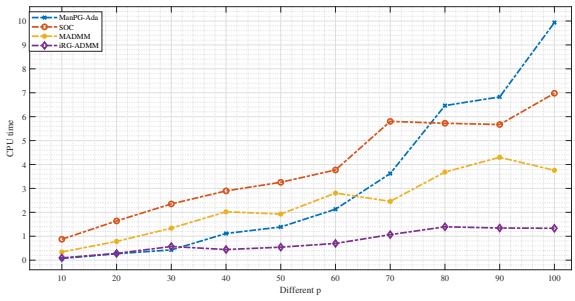

Fig. 3. Trends of CPU time under different p with $n = 300$.

Figure 1 shows that iRG-ADMM achieves smaller objective function values and significantly shorter CPU times compared to ManPG-Ada, SOC, and MADMM. As shown in Figure 2, the computational time for ManPG-Ada, SOC, and MADMM increases substantially when p is changed from 50 to 100. To further investigate this phenomenon, we conduct experiments with $n = 300$ by varying $p \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. As illustrated in Figure 3, the computational efficiency of ManPG-Ada, SOC, and MADMM is highly sensitive to the value of p under a fixed termination criterion. In contrast, the CPU time of iRG-ADMM shows minimal variation with respect to p. This indicates that the computational efficiency of iRG-ADMM is relatively insensitive to the value of p, highlighting its potential for scaling to larger problems. These results suggest that the extension of classical ADMM to Riemannian manifolds, as realized in iRG-ADMM, has practical significance, despite requiring more complex parameter settings.

5 Conclusion

In this paper, we propose an innovative approach for solving a class of Riemannian manifold optimization problems, where the objective functions are composed of both smooth and nonsmooth components. By exploiting the separable structure of these problems, we developed an ADMM-based algorithm that integrates an inertial strategy to improve the convergence of the smooth subproblems on Riemannian manifolds. For the nonsmooth components, we employed well-established algorithms in Euclidean space, ensuring that the overall method is both efficient and scalable. We provide a comprehensive theoretical analysis of the proposed algorithm, proving its ϵ-stationary solution under broad conditions, which guarantees its effectiveness in a wide range of optimization scenarios. The theoretical results validate the robustness of the algorithm, demonstrating that it can achieve convergence even in challenging optimization settings with nonsmooth objective components. Furthermore, we apply the proposed algorithm to Sparse Principal Component Analysis (SPCA), a widely used problem in data processing and dimensionality reduction. The numerical results confirmed the superior performance of our method, showing that it not only

outperforms existing algorithms in terms of objective function values but also demonstrates improved computational efficiency, particularly in large-scale problems. Our extension of classical ADMM to Riemannian manifolds provides a powerful tool for solving complex optimization problems that involve both smooth and nonsmooth components. The proposed method is practical, efficient, and theoretically sound, offering significant potential for applications in a variety of fields, including machine learning, signal processing, and computational biology. Future research could focus on several key areas to enhance the iRG-ADMM algorithm. First, developing adaptive parameter tuning methods or automatic hyperparameter optimization could improve practical deployment across different problem settings. Extending the algorithm to handle non-convex problems, which are common in machine learning, would be a valuable next step. Additionally, parallelizing the algorithm for distributed optimization could improve its efficiency for large-scale applications. Exploring its applicability to other types of manifolds, such as the Stiefel manifold or Grassmannian, would broaden its scope.

Acknowledgements

The work described in this paper was jointly supported by grants from the National Natural Science Foundation of China (Project Nos. 72071202).

References

[1] Ochoa, Daniel E., Poveda, Jorge I. (2025). Robust global optimization on smooth compact manifolds via hybrid gradient-free dynamics, *Automatica*, 171, 111916,

[2] Hauser, J. (2002). A projection operator approach to the optimization of trajectory functionals. *IFAC Proceedings Volumes*, 35(1), 377–382.

[3] Boumal., N. (2023). An Introduction to Optimization on Smooth Manifolds. *Cambridge University Press*.

[4] Jolliffe, I.T., Trendafilov, N.T., Uddin, M. (2003). A modified principal component technique based on the Lasso. *Journal of Computational and Graphical Statistics*, 12(3), 531–547.

[5] Bollhöfer, M., Eftekhari, A.,Scheidegger, S., Schenk, O.(2019). Large-scale sparse inversecovariance matrix estimation, *SIAM Journal on Scientific Computing* , 41, A380–A401,

[6] Huang, W. and Hand, P. (2017). Blind deconvolution by a steepest descent algorithm on a Quotient Manifold. *SIAM Journal on Imaging Sciences*, 11(4), 2757–2785.

[7] Cherian., A. Sra., S. (2016). Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE Transactions on Neural Networks and Learning Systems*, 28(12), 2859–2871.

[8] Smith, S.T. (1994). Optimization techniques on Riemannian manifolds. *Fields Institute Communications* 3(3), 113–135 .

[9] Ring, W., Wirth, B. (2012). Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2), 596–627

[10] Tang, C., Tan, W., Xing, S. et al. (2023). A class of spectral conjugate gradient methods for Riemannian optimization. *Numerical Algorithms*, 94, 131–147.

[11] Zhu, X.J., Sato, H. (2020). Riemannian conjugate gradient methods with inverse retraction. *Computational Optimization and Applications*, 77, 779–810.

[12] Absil, P.A., Mahony, R., Sepulchre, R. (2008). Optimization Algorithms on Matrix Manifolds. *Princeton University Press*, Princeton .

[13] Zhao, S., Yan, T., Wang, K. et al. (2023). Adaptive Trust-Region Method on Riemannian Manifold. *Journal of Scientific Computing*, 96, 67.

[14] Huang, W., Gallivan, K., Absil, P. (2015). A Broyden Class of Quasi-Newton Methods for Riemannian Optimization. *SIAM Journal on Optimization*, 25(3),1660–1685.

[15] Ferreira, O., Oliveira, P. (1998). Subgradient algorithm on Riemannian manifolds. *Journal Of Optimization Theory And Applications*, 97(1), 93–104.

[16] Li, X., Chen, S., Deng, Z., Qu, Q., So, A.M.-C. (2021). Weakly convex optimization over Stiefel manifold using Riemannian subgradient-type methods. *SIAM Journal on Optimization*, 31(3), 1605–1634.

[17] Huang, W., Wei, K. (2022). Riemannian proximal gradient methods. *Mathematical Programming*, 194, 371–413.

[18] Chen, S., Ma, S., So, A.M.-C., Zhang, T. (2020). Proximal gradient method for nonsmooth optimization over the Stiefel manifold. *SIAM Journal on Optimization*, 30(1), 210–239.

[19] Huang, W., Wei, K. (2023). An inexact Riemannian proximal gradient method. *Computational Optimization and Applications* 85, 1–32.

[20] Wang, Z., Liu, B., Chen, S., Ma, S., Xue, L., Zhao, H. (2022).A Manifold Proximal Linear Method for Sparse Spectral Clustering with Application to Single-Cell RNA Sequencing Data Analysis. *INFORMS Journal on Optimization*, 4(2), 200-214.

[21] Chen, S., Deng, Z., Ma, S., So, A.M.-C. (2021). Manifold proximal point algorithms for dual principal component pursuit and orthogonal dictionary learning. *IEEE transactions on signal processing*, 69, 4759–4773.

[22] Tsakiris, M., Vidal, R. (2018). Dual Principal Component Pursuit. *Journal of Machine Learning Research*, 19, 1-49,

[23] Glowinski, R., Marrocco, A. (1975). Sur l'approximation par éléments finis d'ordre un, et la résolution, par pénalisation dualité, d'une classe de problèmes de Dirichlet non linéaires. *ESAIM : Mathematical Modelling and Numerical Analysis*, 41-76.

[24] Gabay, D., Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximatio. *Computers & Mathematics with Applications*, 2, 17-40.

[25] Douglas, J., Rachford, H. (1956). On the numerical solution of heat conduction problems in two and three space variables. *Transactions Of The American Mathematical Society*, 82(2), 421–439.

[26] Eckstein, J., Bertsekas, D.P.(1992). On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(3), 293–318.

[27] He, B., Liu, H., Wang, Z., Yuan, X. (2014). A strictly contractive Peaceman-CRachford splitting method for convex programming. *SIAM Journal on Optimization*, 24, 1011-1040

[28] Lin, Z., Liu. R., Sun, Z. (2011). Linearized alternating direction method with adaptive penalty for low-rank Representation .*Advances in Neural Information Processing systems*, 612-620.

[29] Bai, J., Hager, W., Zhang, H. (2022) .An inexact accelerated stochastic ADMM for separable convex optimization. *Computational Optimization and Applications*, 81, 479-518

[30] Zeng, Y., Wang, Z.,Bai, J., Sheng, X. (2024). An accelerated stochastic ADMM for nonconvex and nonsmooth finite-sum optimization. *Automatica*, 163, 1115554.

[31] Liu, P., Jian, J., Shao, H. et al. (2024). A Bregman-Style Improved ADMM and its Linearized Version in the Nonconvex Setting: Convergence and Rate Analyses. *Journal of the Operations Research Society of China*, 12, 298–340.

[32] Lai, R., Osher, S. (2014). A Splitting Method for Orthogonality Constrained Problems.*Journal of Scientific Computing*, 58, 431–449

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

[34] Li,J., Ma, S., Srivastava, T. (2023). A Riemannian ADMM. arXiv:2211.02163.

[35] Zhang, J., Ma, S., Zhang, S. (2020). Primal-dual optimization algorithms over Riemannian manifolds: an iteration complexity analysis. *Mathematical Programming*, 184, 445–490.

[36] Kleinsteuber, M., Shen, H. (2012). Blind source separation with compressively sensed linear mixtures. *IEEE Signal Processing Letters* , 19(2), 107–110.

[37] Ochs, P., Chen, J., Brox, T., Pock,T. (2014). iPiano: inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7 1388-1419.

[38] Ochs, P., Brox, T., Pock, T, (2015) .iPiasco: inertial proximal algorithm for strongly convex optimization, *Journal of Mathematical Imaging and Vision*, 53, 171-181.

[39] Moreau, J. (1965). Proximité et dualité dans un espace hilbertien. Bulletin des Sciences Mathématiques 9, 273–299

[40] Rockafellar, R.T., Wets, R.J.B. (1997). Variational Analysis. Springer, New York

[41] Zeng, J., Yin, W., Zhou, DX. (2022). Moreau Envelope Augmented Lagrangian Method for Nonconvex Optimization with Linear Constraints. *Journal of Scientific Computing*, **91**, 61.

[42] Drusvyatskiy, D. (2018). The proximal point method revisited. *SIAG/OPT Views and News*, 26, 1–8.

[43] Drusvyatskiy, D., Paquette, C. (2019). Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, 178, 503–558.

[44] Davis, D., Drusvyatskiy, D. (2019). Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1), 207–239.

[45] Boumal, N., Absil, P-A., Cartis, C. (2019). Global rates of convergence for nonconvex optimization on manifolds. IMA Journal of Numerical Analysis, 39(1):1–33, 2019.

[46] Yang, W., Zhang, L., Song. R. (2014). Optimality conditions for the nonlinear programming problems on Riemannian manifolds. Pacific Journal of Optimization, 10(2):415–434.